Texila International Journal of Academic Research
Volume 3, Issue 2, Dec 2016

Enhanced Simplified Symmetric Key Encryption Algorithm

Article by Mahendra Kumar Shrivas[1], Antwi Baffour Boasiako[2], Sangeetha Krishanan[3], Thomas Yeboah[4]

[1]*Information Technology, Texila American University, Guyana, South America,* [2]*Part Time Lecturer, Knustford University College, Kumasi, Ghana,* [3]*Associate Dean, Faculty of Technology, Academic City College, Accra, Ghana,* [4]*Head of Department, Department of ICT, Christen Service University College, Kumasi, Ghana*

[1]*mshrivas@texilaconnect.com,* [2]*thomyebs24@gmail.com,*
[3]*antwibaffourboasiako@rocketmail.com,* [4]*sangeetha.krishnan@accghana.com*

Abstract

Data has become very important not only for individuals but for organizations as well. Data security is the biggest challenge that we care facing currently. Recent successful hacks and data breaches have certainly played an important role in the development of data security related technologies.

Cryptography is a well adopted method to ensure that data is secure and confidentiality of user data is maintained.

The content owner encrypts the actual data using an encryption key which converts the data into cipher text. The cipher text is an intermediate data which is unreadable form which can be shared amount other users and can be stored in the various storage media.

The cipher text can be converted into the actual data using the same encryption key in case of symmetric key encryption or using different key in case of asymmetric key encryption.

Currently, the Encryption/Decryption algorithms that exist depend on complex mathematical manipulations. The length of the encryption keys are growing and growing to get more secure and more stronger encryption thus processing throughput and memory consumption requirement is also growing.

In this work researchers are focusing on various symmetric key encryption throughput and memory consumption with proposed high speed new algorithm which can be useful for the devices with low memory and processing capabilities. The work sought the possibility to trim down the complicated throughput of symmetric cryptography and ensuring maximum security at the same time.

Keywords*: Cryptography, Algorithms, Authentication, Cipher text, Complex Mathematical Manipulations, Encryption, Decryption, Symmetric Key.*

Introduction

We are living in cyber age where data and information is the biggest wealth. Our personal, professional and organizations data is available in the devices which are connected with the Internet. Data Hacks and threats in computer networks are growing day by day which demands more security (Shrivas, Dr. Amoako, Boateng, & Dr. Yeboah, 2015) and reduction in both the time for data transmission and the space requirement for data storage. This can be achieved by encryption and compression, such kind of system is called compression-crypto system.

Cryptography is powerful tool which provides authenticity, privacy, integrity, and limited access to data. For the reason that networks often involve even greater risks, data is often secured with encryption, plausibly in combination with other controls. The content owner encrypts the actual data using an encryption key which converts the data into cipher text. The cipher text is an intermediate

data which is unreadable form which can be shared amount other users and can be stored in the various storage media (Shrivas & Singh, ICMCT, 2015).

The cipher text can be converted into the actual data using the same encryption key in case of symmetric key encryption or using different key in case of asymmetric key encryption.

The most significant type of cryptography is the symmetric key encryption. In the symmetric key encryption, encryption and decryption process both uses the same key thus key should be private to avoid any data breaches. Symmetric key algorithms are high speed and do not consume too much of computing resources. Although there are scope of improvement, thus in this work researchers are focusing on various symmetric key encryption algorithms like Data Encryption Standard (DES), Triple Data Encryption Standard (DES), Advanced Encryption Standard (AES), Blowfish(Shrivas & Singh, IJICTM, 2014), etc. and their throughputs and memory consumptions and proposing high speed new Enhanced Simplified Symmetric Key Encryption Algorithm (in short authors are going to call it proposed algorithm subsequently) which can be useful for the devices with low memory and processing capabilities. The work sought the possibility to trim down the complicated throughput of symmetric cryptography and ensuring maximum security at the same time.

Objectives of the study

- To make data encryption algorithms simple but difficult to cryptanalysis.
- To analysis the three main symmetric-key cryptography schemes: DES, AES and Blowfish, and come out with new cryptographic algorithms.
- To strengthen confidentiality, authenticity, integrity, availability and identification of user data and information in networks.
- To minimize complex mathematical manipulations of these algorithms.

Research questions

- Is it possible to reduce the complexity of algorithms mathematical manipulations and still ensure maximum security?
- Can the new encryption algorithms design and implementation to enhance performance?
- How effective is an algorithm using variable key length vary from 65 bytes to 72 bytes of symmetric key technique for encryption and decryption of data?
- Can the proposed Algorithm be comparable to DES, Triple-DES, Blowfish and AES?

Significance of the study

Data security in these contemporary times is a must. For your secrets to be secure, it may be essential to attach fortifications not afford by your computer operating systems. The incorporated fortifications may be sufficient in some cases. If no individual ever attempts to break into or pilfer data from a particular computer, its data will be secured. Otherwise if the impostor has not learned how to get around the simple default mechanisms, they are sufficient. Nevertheless many invaders do have the skills and resources to break various security systems. If you make your mind up to do nothing and hope that no skilled cracker targets your information, you may be fortunate, and nothing horrific will happen.

Data Encryption is one of the imperative tools for protecting data from an unauthorized access, any of various methods that are used to turn readable files into gobbledygook. Even if an attacker obtains the contents of the file, it is twaddle. It does not matter whether or not the operating system protections functioned.

Scope of the study

There are two type of Cryptography, Symmetric and Asymmetric which include encryption and decryption process. The scope of this study is limited to the symmetric key cryptographic scheme the algorithms considered in the study include: DES, Triple-DES, AES and Blowfish.

Literature review

The history and background of encryption algorithms should bring into being a clearer understanding for the need of complex mathematical computation for these algorithms for security.

The next part of this review will look at other previous work to serve as empirical evidence in other to have a platform to build upon this work.

Finally, this review will look at the more specific encryption techniques known as the AES, DES, Triple DES and Blowfish. Following this, the investigation has been expanded to search for Enhanced Simplified Symmetric Key Encryption Algorithm that can serve the same purpose those in existence.

Categorization of the cryptographic algorithms

The encryption algorithms are basically classified into two types based on the keys used for the encryption; these are the Symmetric and Asymmetric key encryption (Shrivas & Singh, IJICTM, 2014).

Asymmetric key encryption is the technique in which the keys are different for the encryption and the decryption process. They are also known as the public key encryption. One of these keys is published or public and the other is kept private. Diffie-Hellman key agreement algorithm, Rivest Shamir Adleman (RSA), Elliptic Curve Cryptography (ECC), El Gamal and Digital Signature Algorithm (DSA) are most popular asymmetric algorithms (Ahmad, Alam, Rahman, & Tamura, 2015).

Symmetric-key algorithms are a class of algorithms for cryptography that use pettily related cryptographic keys for both decryption and encryption. Typical symmetric encryption algorithms include DES, Triple DES, RC2, RC5, Twofish, Blowfish, IDEA and AES (Shrivas & Singh, IJICTM, 2014). Most symmetric algorithms can operate in two modes, namely Cipher Block Chaining Mode (CBC) or Electronic Codebook Mode (ECB) (Klinc, Hazay, Jagmohan, Krawczyk, & Rabin, 2012).

The Symmetric-key ciphers are split into categories, such as Permutation Ciphers, Transposition Ciphers and Substitution Ciphers. There exists other, but also combinations of the above. The main symmetric-key cryptography schemes include DES, AES and Blowfish[1]. (Anonov A., Gounelas F., and Kauppila J., 2006) put forth the algorithms of DES, Triple DES, AES and Blowfish as stated below:

Data encryption standard

DES was the result of a research project set up by International Business Machines (IBM) Corporation in the late 1960's which resulted in a cipher known as LUCIFER. In the early 1970's it was decided to commercialize LUCIFER and a number of significant changes were introduced.

As stated in (Shah K. R., March, 2012), the DES was published by the United States' National Bureau of Standards in January 1977 as an algorithm to be used for unclassified data (information not concerned with national security).

(Mandar M. K., March 2013), affirm that from 2001 the AES will replace DES. After 25 years of analysis, the only security problem with DES found is that its key length is too short. DES uses a 56 bit key which can be broken using brute force methods, & is now considered to be insecure for many application. It was acknowledged that DES was not secure as a result of advancement in processing power computer. From (Sharma, 2010), the purpose of NIST was to define a replacement for DES that can be used in non-military information security applications by US government agencies.

(Yogesh K., Oct 2011), stated that the DES algorithm is vulnerable to Linear Cryptanalysis attacks. By such an attack, the algorithm in its sixteen rounds can be broken using 2^{43} plaintexts. This vulnerability raises a notable risk when encrypting bulk data that may be predictable with keys that are constant.

[1] Cryptography by A. Antonov, F. Gounelas, J. Kauppila, June 13, 2006

The DES algorithm

The Data Encryption Standard (DES), as specified in FIPS Publication 46-3, is a block cipher operating on 64-bit data blocks with key length 56 bits the straightforward "work factor" of the algorithm is 2^{56} (i.e., the number of keys that would have to be tried is 2^{56} or approximately 7.6 x 10^{16}). From (Vikendra S., (2013))

1. Input plain text A = {}
2. Divide A into n blocks of 64 bits.
3. For each blocks I = 0 to n-1
4. Calculate initial permutation IP and
5. Divided into two parts
6. $L_0 \leftarrow$ left sub part
7. $R_0 \leftarrow$ right sub part
8. Round i have inputs L_{i-1}, R_{i-1}
9. Output will be

$Li = R_{i-1}$,

$R_i = L_{i-1}$ XOR f (R_{i-1}, K_i)

10. Ki is the sub key for the ith round, where $1 \leq i \leq 16$.
11. After round 16, Swap L0 and R0 (so that the decryption algorithm has the same structure as the encryption algorithm.)
12. Finally, compute IP-1
13. Output = cipher text

Triple DES

From (Ajay K., January, 2012), Triple DES was developed in 1998 and derived from DES. It applies the DES cipher algorithm three times to each of the data blocks. It has a key size of 168 bits but provides at most 112 bits of security remaining 56 bits are utilized in the keying options.

The standards define three keying options;

K_1, K_2&K_3 and are given as:

First keying option: All the three keys are independent.

Second keying operation: K_1&K_2 are independent, and $K_3 = K_1$.

Third Keying option is all the three keys are identical $K_1 = K_2 = K_3$.

The block size used in the algorithm is 64 bits and 48 DES equivalent rounds have been used to encrypt the data. The security of TDES is effective but the main limitation of the standard is that 56 bits are not actually used for the encryption.

Triple DES is slower than other block cipher methods. The following expression is used for encryption purpose. C(t) = Ek1 (Dk2 (Ek3 (t))) (Ankita P. B., April - 2013)

Triple DES algorithm

Encryption is done by-

C= (Encryption) k3 ((Decryption) k2 ((Encryption) K1 (I))).

with Decryption is done by:

I= ((Decryption) K1 ((Encryption) K2 ((Decryption) K3 (C)))

I… information

C ... cipher text

ki ... key and i is iteration

Where Encryption and Decryption are DES encryption and DES decrypt (Vikendra S., (2013))

Advanced encryption standard

(Paar C. and Pelzl J., 2010), stated that Rijndael was proposed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen. The proposed encryption's key size varies between 128, 192 and 256 bits; but only the key size of 128 bits was approved as the AES standard.

According to (Al-Hazaimeh, March 2013), the National Institute of Standards and Technology (NIST) in 1997 announced officially that Rijndael algorithm would become the Advanced Encryption Standard (AES) to replace the aging Data Encryption Standard (DES). AES algorithm is a block cipher text the block size can be 128, 192 or 256 bits. 128(AES-128), 192(AES-192) and 256 (AES-256) bits key lengths. The use of AES becomes effective in May, 2002.

The AES algorithm

1. Input Block is split up into bytes depending on its size L (see below), ex for L = 128 into 16 bytes, m_0, m_1, ..., m_n and the same is done to Input Key which is of the same size, k_0, k_1, ..., k_n.
2. According to the key size, a specific number of rounds of the following function are performed (for example, when L = 128 we have 10 rounds).
3. Round (S, Round Key) Where S is initially Input Block and Round Key is derived from Input Key via key scheduling.
4. Round (S, Round Key) = Sub Bytes(S);

Shift Rows(S);

Mix Columns(S);

Add Round Key(S, Round Key);

They added a brief description of the operations each of these internal functions performs.

Sub bytes(S): Performs $y_i = Ax^{-1} + b$ where x is every byte of S, A is an S-box and b is known dependant on the size of A (and subsequently on the size of m_i).

Shift Rows(S): This is the simplest operation of the four which simply shifts the elements of the matrix whose elements are the bytes of S by a given number of positions. For example,

with L = 128, Shift Rows(S) would look like this

$$\begin{pmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{pmatrix} \rightarrow \begin{pmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,1} & s_{1,2} & s_{1,3} & s_{1,0} \\ s_{2,2} & s_{2,3} & s_{2,0} & s_{2,1} \\ s_{3,3} & s_{3,0} & s_{3,1} & s_{3,2} \end{pmatrix}$$

Mix Columns(S): Works by manipulating the columns of the state S (remember they look like the above). It forms a polynomial with coefficients the byte values of the columns. This polynomial is then multiplied with a fixed polynomial c(*x*) modulo x^4-1. Modulo arithmetic ensures the result to be a polynomial of degree 3. The resulting polynomials' coefficients will be the column of the new state S^I. We work through all the columns of S in a similar fashion to form S^I.

Add Round Key(S, Round Key): This function performs the XOR operation between the elements of S and the elements of Round Key (Shrivas & Singh, IJICTM, 2014).

Blowfish

Blowfish is a symmetric block cipher that can be used as a drop-in replacement for DES (Data Encryption Standard) or IDEA (International Data Encryption Algorithm). It takes a variable-length key, from 32 bits to 448 bits, making it ideal for both domestic and exportable use (Alabaichi, Ahmad, & Mahmod, 2013). It is a freely available symmetric block cipher designed in 1993 by Bruce Schneier.

Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. Since then, it has been analyzed considerably, and is slowly gaining acceptance as a strong encryption algorithm. Blowfish is not patented, is license-free, and is available free for all uses.

Blowfish process

- Initialize P array and S boxes with Hexadecimal digits of Pi.
- XOR P-array with the key bits (i.e., P1 XOR (first 32 bits of key), P2 XOR (second 32 bits of key)…
- Use the above method to encrypt the all-zero string.
- This new output is P1 and P2.
- Encrypt the new P1 and P2 with the modified sub keys.
- This new output is now P3 and P4.
- Repeat the above steps until we get all the elements of P array i.e P1, P2….

Blowfish algorithm

1. Take X = input (0-64) bits.
2. Divide X it into two equal halves such that
3. XL = input(0-31)
4. XR = input (32-63).
5. For I = 1 to 15
6. Li = Li-1
7. Compute F(XL)=((S1,a + S2,b mod232) XOR S3,c) + S4,d
8. XR = XL XOR XR
9. Swap XL and XR
10. XL = XL XOR P18
11. XR = XR XOR P17
12. output = combined result of L and R
13. return output

f-Function- f-function uses s-boxes and it can be implemented into following steps

1. Get input
2. L = input[0-15]
3. R = input[16-31]
4. Centre Lc = input[8-15]
5. Centre Rc = input[16-23]
6. L = S-Box(L) (note that the result of the S-Box is a 32 bit data stream)
7. centreLc = S-Box(centreLc)
8. centreRc = S-Box(centreRc)
9. R = S-Box(R)
10. L = L + centreLc (note: this is mod 232 addition)
11. L = L XOR centreRc
12. L = L + R
13. return L

The proposed enhanced simplified symmetric key encryption algorithm

Technology is bound to catch up to all cryptosystems and surpass their computational limits. For this reason, any new encryption method should be welcomed as future input to viable alternatives, especially suggestions that comply to the “low computational cost”-“high resilience to cryptanalysis” paradigm.[2]

[2]Stergiopoulos G., Miltiadis K., and Dimitris G.Information Security and Critical Infrastructure Protection Research Laboratory, 76 Patission Ave., Athens GR-10434, Greece.

Reasons for adopting symmetric key

Authors adopted Symmetric key cryptographic scheme because only one key is needed for communication. The selected cryptographic scheme involves five requirements and these are: plain text, cipher text, encryption algorithm, decryption algorithm, and a secret key (Agrawal & Mishra, 2012).

Symmetric key length

This new proposed algorithm is a block cipher that divides data into blocks of equal length and then encrypts each block using a special mathematical set of functions known as Key. This algorithm uses variable key length which will vary from 65 bytes to 72 bytes of symmetric key technique for encryption and decryption of data i.e. it uses the same key at both ends. Selection of the key purely random based. Thus, the key distribution predicament can be handled easily. Another positive point of the algorithm is that it protects the cipher text from Brute-force attacks as the key is length in the encryption process because of 2^{288} required to break the key.

Cryptographic models

The study is designed on the basis of two models:

- Simplified Model
- Conventional Model

Diagrammatically, these models that formed the bases of the design of the work is shown below.

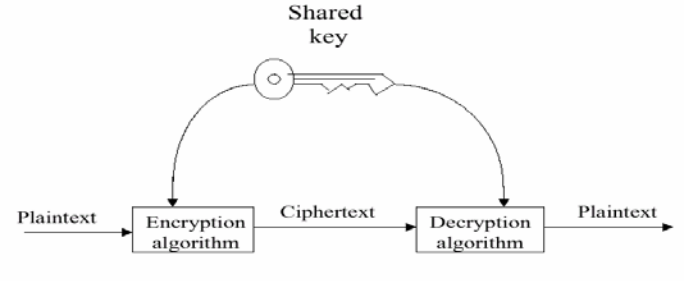

Figure 1. Simplified model for symmetric encryption and decryption techniques

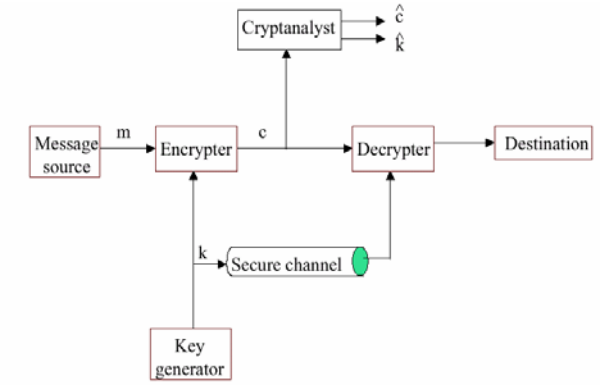

Figure 2. Conventional model

Proposed algorithm

In this algorithm, the encryption process does a variety of matrix Operation like transpose, column mix, row mix, permutation on the message for protecting it against unauthorized attacks. The propose algorithm is adopted AES, therefore has it features thereof.

Pseudo code of proposed encryption algorithm

1. Select 64 bytes key value from the variable key size 65 to 72 Key [64];
2. Select first 64 bytes plain text form the file
   Double Text [64];
3. Arrange both values in matrix form.
   Key [i][j] and Text [i][j];
4. Execute Column shifting
   Col_Shift_Key [i][j] = Col_shift (Key [i][j]);
   Col_Shift_Text [i][j] = Col_shift (Text [i][j]);
5. Apply Combine function between the two
   Com [i][j] = ComB (Col_Shift_Key [i][j],
   Col_Shift_Text [i][j]);
6. Execute Permutation function.
   Permu [i][j] = PerMut (Com [i][j]);
7. Execute Column Swapping.
   Com_Col_Swap [i][j] = Com_Col_Swap (Permu [i][j]);
8. Perform transposition.
   Trans [i][j] = Trans(Com_Col_Swap [i][j]);
9. Perform row mixing.
   Row_Mix [i][j] = Row_Mix (Trans [i][j]);
10. Repeat step 1 to 9 till Avg (Enc_Number, Ran_Number).
11. Cipher [i][j] = Row_Mix [i][j];
12. Exit.

PSEUDO code of proposed decryption algorithm

1. Select cipher text value from the encrypted file
   Cipher [i][j];
2. Apply row mixing. in the two matrixes
   Rev_Row_Mix [i][j] = Rev_Row_Mix (Cipher [i][j]);
3. Apply transpositions among respective Colum
   Trans [i][j] = Trans (Rev_Row_Mix [i][j]);
4. Execute Column Shifting.
   Rev_Com_Col_Shift [i][j] = Rev_Com_Col_Shift (Trans [i][j]]);
5. Apply Permutation function.
   Rev_Permu [i][j] = Rev_PerMut (Rev_Com_Col_Shift [i][j]);
6. Execute Division function to again break the matrix
   Div_Key [i][j] = Divi (Rev_Permu [i][j]);
   Div_Text [i][j] = Divi (Rev_Permu [i][j]);
7. Execute Column Swapping
   Rev_Col_Swap_ Div_Key [i][j] = Rev_Col_Swap (Div_ Key [i][j]);
   Rev_Col_Swap_ Div_Text [i][j] = Rev_Col_Swap (Div_ Text [i][j]);
8. Reiterate step 1 to 8 till Avg (Enc_Number, Ran_Number).
9. Text [i][j] = Rev_Col_Swap_ Div_Text [i][j]
10. End

Matrix form for encryption algorithm

For Instance: Select 16 byte key value and arrange in matrix form in the following way

1. Select 64 byte Key and arrange in the following way

Key Matrix (KM) =
$$\begin{pmatrix} K01 & K02 & K03 & K04 & K05 & K06 & K07 & K08 \\ K09 & K10 & K11 & K12 & K13 & K14 & K15 & K16 \\ K17 & K18 & K19 & K20 & K21 & K22 & K23 & K24 \\ K25 & K26 & K27 & K28 & K29 & K30 & K31 & K32 \\ K33 & K34 & K35 & K36 & K37 & K38 & K39 & K40 \\ K41 & K42 & K43 & K44 & K45 & K46 & K47 & K48 \\ K49 & K50 & K51 & K52 & K53 & K54 & K55 & K56 \\ K57 & K58 & K59 & K60 & K61 & K62 & K63 & K64 \end{pmatrix}$$

2. Select 64 byte plain text and arrange in the following way

Plain Text Matrix (PT)=
$$\begin{pmatrix} P01 & P02 & P03 & P04 & P05 & P06 & P07 & P08 \\ P09 & P10 & P11 & P12 & P13 & P14 & P15 & P16 \\ P17 & P18 & P19 & P20 & P21 & P22 & P23 & P24 \\ P25 & P26 & P27 & P28 & P29 & P30 & P31 & P32 \\ P33 & P34 & P35 & P36 & P37 & P38 & P39 & P40 \\ P41 & P42 & P43 & P44 & P45 & P46 & P47 & P48 \\ P49 & P50 & P51 & P52 & P53 & P54 & P55 & P56 \\ P57 & P58 & P59 & P60 & P61 & P62 & P63 & P64 \end{pmatrix}$$

3. Arrange key value and Plain Text in following way

$$\begin{pmatrix} K01 & K02 & K03 & K04 & K05 & K06 & K07 & K08 \\ K09 & K10 & K11 & K12 & K13 & K14 & K15 & K16 \\ K17 & K18 & K19 & K20 & K21 & K22 & K23 & K24 \\ K25 & K26 & K27 & K28 & K29 & K30 & K31 & K32 \\ K33 & K34 & K35 & K36 & K37 & K38 & K39 & K40 \\ K41 & K42 & K43 & K44 & K45 & K46 & K47 & K48 \\ K49 & K50 & K51 & K52 & K53 & K54 & K55 & K56 \\ K57 & K58 & K59 & K60 & K61 & K62 & K63 & K64 \end{pmatrix} \begin{pmatrix} P01 & P02 & P03 & P04 & P05 & P06 & P07 & P08 \\ P09 & P10 & P11 & P12 & P13 & P14 & P15 & P16 \\ P17 & P18 & P19 & P20 & P21 & P22 & P23 & P24 \\ P25 & P26 & P27 & P28 & P29 & P30 & P31 & P32 \\ P33 & P34 & P35 & P36 & P37 & P38 & P39 & P40 \\ P41 & P42 & P43 & P44 & P45 & P46 & P47 & P48 \\ P49 & P50 & P51 & P52 & P53 & P54 & P55 & P56 \\ P57 & P58 & P59 & P60 & P61 & P62 & P63 & P64 \end{pmatrix}$$

4. (a) Execute Column Shifting Function in the following way:

i. Replace 1st column of KM with 2nd column of PT
ii. Replace 3rd column of KM with 4th column of PT
iii. Replace 5th column of KM with 6th column of PT
iv. Replace 7th column of KM with 8th column of PT

$$\begin{pmatrix} K01 & K02 & K03 & K04 & K05 & K06 & K07 & K08 \\ K09 & K10 & K11 & K12 & K13 & K14 & K15 & K16 \\ K17 & K18 & K19 & K20 & K21 & K22 & K23 & K24 \\ K25 & K26 & K27 & K28 & K29 & K30 & K31 & K32 \\ K33 & K34 & K35 & K36 & K37 & K38 & K39 & K40 \\ K41 & K42 & K43 & K44 & K45 & K46 & K47 & K48 \\ K49 & K50 & K51 & K52 & K53 & K54 & K55 & K56 \\ K57 & K58 & K59 & K60 & K61 & K62 & K63 & K64 \end{pmatrix} \Rightarrow \begin{pmatrix} P02 & K02 & P04 & K04 & P06 & K06 & P08 & K08 \\ P10 & K10 & P12 & K12 & P14 & K14 & P16 & K16 \\ P18 & K18 & P20 & K20 & P22 & K22 & P24 & K24 \\ P26 & K26 & P28 & K28 & P30 & K30 & P32 & K32 \\ P34 & K34 & P36 & K36 & P38 & K38 & P40 & K40 \\ P42 & K42 & P44 & K44 & P46 & K46 & P48 & K48 \\ P50 & K50 & P52 & K52 & P54 & K54 & P56 & K56 \\ P58 & K58 & P60 & K60 & P62 & K62 & P64 & K64 \end{pmatrix}$$

4. (b) Execute Column Shifting Function in the following way:

i. Replace 2nd column of PT with 1st column of KM
ii. Replace 4rd column of PT with3rd column of KM
iii. Replace 6th column of PT with 5th column of KM
iv. Replace 8th column of PT with 7th column of KM

$$\begin{pmatrix} P01 & P02 & P03 & P04 & P05 & P06 & P07 & P08 \\ P09 & P10 & P11 & P12 & P13 & P14 & P15 & P16 \\ P17 & P18 & P19 & P20 & P21 & P22 & P23 & P24 \\ P25 & P26 & P27 & P28 & P29 & P30 & P31 & P32 \\ P33 & PT3 & P35 & P36 & P37 & P38 & P39 & P40 \\ P41 & PT2 & P43 & P44 & P45 & P46 & P47 & P48 \\ P49 & P50 & P51 & P52 & P53 & P54 & P55 & P56 \\ P57 & T58 & P59 & P60 & P61 & P62 & P63 & P64 \end{pmatrix} \Rightarrow \begin{pmatrix} P01 & K01 & P03 & K03 & P05 & K05 & P07 & K07 \\ P09 & K09 & P11 & K11 & P13 & K13 & P15 & K15 \\ P17 & K17 & P19 & K19 & P21 & K21 & P23 & K23 \\ P25 & K25 & P27 & K27 & P29 & K29 & P31 & K31 \\ P33 & K33 & P35 & K35 & P37 & K37 & P39 & K39 \\ P41 & K41 & P43 & K43 & P45 & K45 & P47 & K47 \\ P49 & K49 & P51 & K51 & P53 & K53 & P55 & K55 \\ P57 & K57 & P59 & K59 & P61 & K61 & P63 & K63 \end{pmatrix}$$

5. Apply Combine function between the two matrixes in the following way:

Joint corresponding columns of 4(b) to 4(a), that is columns 1 and 1, 2 and 2, 3 and 3, … 8 and 8

$$\begin{pmatrix} P02 & K02 & P04 & K04 & P06 & K06 & P08 & K08 \\ P10 & K10 & P12 & K12 & P14 & K14 & P16 & K16 \\ P18 & K18 & P20 & K20 & P22 & K22 & P24 & K24 \\ P26 & K26 & P28 & K28 & P30 & K30 & P32 & K32 \\ P34 & K34 & P36 & K36 & P38 & K38 & P40 & K40 \\ P42 & K42 & P44 & K44 & P46 & K46 & P48 & K48 \\ P50 & K50 & P52 & K52 & P54 & K54 & P56 & K56 \\ P58 & K58 & P60 & K60 & P62 & K62 & P64 & K64 \\ P01 & K01 & P03 & K03 & P05 & K05 & P07 & K07 \\ P09 & K09 & P11 & K11 & P13 & K13 & P15 & K15 \\ P17 & K17 & P19 & K19 & P21 & K21 & P23 & K23 \\ P25 & K25 & P27 & K27 & P29 & K29 & P31 & K31 \\ P33 & K33 & P35 & K35 & P37 & K37 & P39 & K39 \\ P41 & K41 & P43 & K43 & P45 & K45 & P47 & K47 \\ P49 & K49 & P51 & K51 & P53 & K53 & P55 & K55 \\ P57 & K57 & P59 & K59 & P61 & K61 & P63 & K63 \end{pmatrix}$$

6. Execute Permutation function in the following way: Here the first column is obtained by writing the elements of the sixteenth row and then the fifteenth row of in reverse order; similarly the other seven columns are obtained by using the fourteenth and thirteenth rows, twelfth and eleventh rows, and in that order.

$$\begin{pmatrix} K63 & K47 & K31 & K15 & K64 & K48 & K32 & K16 \\ P63 & P47 & P31 & P15 & P64 & P48 & P32 & P16 \\ K61 & K45 & K29 & K13 & K62 & K46 & K30 & K14 \\ P61 & P45 & P29 & P13 & P62 & P46 & P30 & P14 \\ K59 & K43 & K27 & K11 & K60 & K44 & K28 & K12 \\ P59 & P43 & P27 & P11 & P60 & P44 & P28 & P12 \\ K57 & K41 & K25 & K09 & K58 & K42 & K26 & K10 \\ P57 & P41 & P25 & P09 & P58 & P42 & P26 & P10 \\ K55 & K39 & K23 & K07 & K56 & K40 & K24 & K08 \\ P55 & P39 & P23 & P07 & P56 & P40 & P24 & P08 \\ K53 & K37 & K21 & K05 & K54 & K38 & K22 & K06 \\ P53 & P37 & P21 & P05 & P54 & P38 & P22 & P06 \\ K51 & K35 & K19 & K03 & K52 & K36 & K20 & K04 \\ P51 & P35 & P19 & P03 & P52 & P36 & P20 & P04 \\ K49 & K33 & K17 & K01 & K50 & K34 & K18 & K02 \\ P49 & P33 & P17 & P01 & P50 & P34 & P18 & P02 \end{pmatrix}$$

7. Execute Column Swapping Function. Swap the columns 1and 2, 3 and 4, 5 and 6, and 7 and 8.

$$\begin{pmatrix} K47 & K63 & K15 & K31 & K48 & K64 & K16 & K32 \\ P47 & P63 & P15 & P31 & P48 & P64 & P16 & P32 \\ K45 & K61 & K13 & K29 & K46 & K62 & K14 & K30 \\ P45 & P61 & P13 & P29 & P46 & P62 & P14 & P30 \\ K43 & K59 & K11 & K27 & K44 & K60 & K12 & K28 \\ P43 & P59 & P11 & P27 & P44 & P60 & P12 & P28 \\ K41 & K57 & K09 & K25 & K42 & K58 & K10 & K26 \\ P41 & P57 & P09 & P25 & P42 & P58 & P10 & P26 \\ K39 & K55 & K07 & K23 & K40 & K56 & K08 & K24 \\ P39 & P55 & P07 & P23 & P40 & P56 & P08 & P24 \\ K37 & K53 & K05 & K21 & K38 & K54 & K06 & K22 \\ P37 & P53 & P05 & P21 & P38 & P54 & P06 & P22 \\ K35 & K51 & K03 & K19 & K36 & K52 & K04 & K20 \\ P35 & P51 & P03 & P19 & P36 & P52 & P04 & P20 \\ K33 & K49 & K01 & K17 & K34 & K50 & K02 & K18 \\ P33 & P49 & P01 & P17 & P34 & P50 & P02 & P18 \end{pmatrix}$$

8. Perform Transposition Function.

$$\begin{pmatrix} K47 & P47 & K45 & P45 & K43 & P43 & K41 & P41 & K39 & P39 & K37 & P37 & K35 & P35 & K33 & P33 \\ K63 & P63 & K61 & P61 & K59 & P59 & K57 & P57 & K55 & P55 & K53 & P53 & K51 & P51 & K49 & P59 \\ K15 & P15 & K13 & P13 & K11 & P11 & K09 & P09 & K07 & P07 & K05 & P05 & K03 & P03 & K01 & P01 \\ K31 & P31 & K29 & P29 & K27 & P27 & K25 & P25 & K23 & P23 & K21 & P21 & K19 & P19 & K17 & P17 \\ K48 & P48 & K46 & P46 & K44 & P44 & K42 & P42 & K40 & P40 & K38 & P38 & K36 & P36 & K34 & P34 \\ K64 & P64 & K62 & P62 & K60 & P60 & K58 & P58 & K56 & P56 & K54 & P54 & K52 & P52 & K50 & P50 \\ K16 & P16 & K14 & P14 & K12 & P12 & K10 & P10 & K08 & P08 & K06 & P06 & K04 & P04 & K02 & P02 \\ K32 & P32 & K30 & P30 & K28 & P28 & K26 & P26 & K24 & P24 & K22 & P22 & K20 & P20 & K18 & P18 \end{pmatrix}$$

9. Perform Row Mixing Function. The rows 1 and 8, 2and 7, 3 and 6, and 4 and 5 swap

$$\begin{pmatrix} K32 & P32 & K30 & P30 & K28 & P28 & K26 & P26 & K24 & P24 & K22 & P22 & K20 & P20 & K18 & P18 \\ K16 & P16 & K14 & P14 & K12 & P12 & K10 & P10 & K08 & P08 & K06 & P06 & K04 & P04 & K02 & P02 \\ K64 & P64 & K62 & P62 & K60 & P60 & K58 & P58 & K56 & P56 & K54 & P54 & K52 & P52 & K50 & P50 \\ K48 & P48 & K46 & P46 & K44 & P44 & K42 & P42 & K40 & P40 & K38 & P38 & K36 & P36 & K34 & P34 \\ K31 & P31 & K29 & P29 & K27 & P27 & K25 & P25 & K23 & P23 & K21 & P21 & K19 & P19 & K17 & P17 \\ K15 & P15 & K13 & P13 & K11 & P11 & K09 & P09 & K07 & P07 & K05 & P05 & K03 & P03 & K01 & P01 \\ K63 & P63 & K61 & P61 & K59 & P59 & K57 & P57 & K55 & P55 & K53 & P53 & K51 & P51 & K49 & P49 \\ K47 & P47 & K45 & P45 & K43 & P43 & K24 & P24 & K39 & P39 & K37 & P37 & K35 & P35 & K33 & P33 \end{pmatrix}$$

10. Reiterate Processes 3 to 9 until Average (Encryption, Random Number).
11. Cipher

$$\begin{pmatrix} K32 & P32 & K30 & P30 & K28 & P28 & K26 & P26 & K24 & P24 & K22 & P22 & K20 & P20 & K18 & P18 \\ K16 & P16 & K14 & P14 & K12 & P12 & K10 & P10 & K08 & P08 & K06 & P06 & K04 & P04 & K02 & P02 \\ K64 & P64 & K62 & P62 & K60 & P60 & K58 & P58 & K56 & P56 & K54 & P54 & K52 & P52 & K50 & P50 \\ K48 & P48 & K46 & P46 & K44 & P44 & K42 & P42 & K40 & P40 & K38 & P38 & K36 & P36 & K34 & P34 \\ K31 & P31 & K29 & P29 & K27 & P27 & K25 & P25 & K23 & P23 & K21 & P21 & K19 & P19 & K17 & P17 \\ K15 & P15 & K13 & P13 & K11 & P11 & K09 & P09 & K07 & P07 & K05 & P05 & K03 & P03 & K01 & P01 \\ K63 & P63 & K61 & P61 & K59 & P59 & K57 & P57 & K55 & P55 & K53 & P53 & K51 & P51 & K49 & P49 \\ K47 & P47 & K45 & P45 & K43 & P43 & K24 & P24 & K39 & P39 & K37 & P37 & K35 & P35 & K33 & P33 \end{pmatrix}$$

12. End

Matrix form for decryption algorithm

1. Select cipher text value from the encrypted file

$$\begin{pmatrix} K32 & P32 & K30 & P30 & K28 & P28 & K26 & P26 & K24 & P24 & K22 & P22 & K20 & P20 & K18 & P18 \\ K16 & P16 & K14 & P14 & K12 & P12 & K10 & P10 & K08 & P08 & K06 & P06 & K04 & P04 & K02 & P02 \\ K64 & P64 & K62 & P62 & K60 & P60 & K58 & P58 & K56 & P56 & K54 & P54 & K52 & P52 & K50 & P50 \\ K48 & P48 & K46 & P46 & K44 & P44 & K42 & P42 & K40 & P40 & K38 & P38 & K36 & P36 & K34 & P34 \\ K31 & P31 & K29 & P29 & K27 & P27 & K25 & P25 & K23 & P23 & K21 & P21 & K19 & P19 & K17 & P17 \\ K15 & P15 & K13 & P13 & K11 & P11 & K09 & P09 & K07 & P07 & K05 & P05 & K03 & P03 & K01 & P01 \\ K63 & P63 & K61 & P61 & K59 & P59 & K57 & P57 & K55 & P55 & K53 & P53 & K51 & P51 & K49 & P49 \\ K47 & P47 & K45 & P45 & K43 & P43 & K24 & P24 & K39 & P39 & K37 & P37 & K35 & P35 & K33 & P33 \end{pmatrix}$$

2. Apply row mixing in the matrix by swapping rows 1 and 8, 2 and 7, 3 and 4, and 5 and 6 to obtain this result.

$$\begin{pmatrix} K47 & P47 & K45 & P45 & K43 & P43 & K41 & P41 & K39 & P39 & K37 & P37 & K35 & P35 & K33 & P33 \\ K63 & P63 & K61 & P61 & K59 & P59 & K57 & P57 & K55 & P55 & K53 & P53 & K51 & P51 & K49 & P59 \\ K15 & P15 & K13 & P13 & K11 & P11 & K09 & P09 & K07 & P07 & K05 & P05 & K03 & P03 & K01 & P01 \\ K31 & P31 & K29 & P29 & K27 & P27 & K25 & P25 & K23 & P23 & K21 & P21 & K19 & P19 & K17 & P17 \\ K48 & P48 & K46 & P46 & K44 & P44 & K42 & P42 & K40 & P40 & K38 & P38 & K36 & P36 & K34 & P34 \\ K64 & P64 & K62 & P62 & K60 & P60 & K58 & P58 & K56 & P56 & K54 & P54 & K52 & P52 & K50 & P50 \\ K16 & P16 & K14 & P14 & K12 & P12 & K10 & P10 & K08 & P08 & K06 & P06 & K04 & P04 & K02 & P02 \\ K32 & P32 & K30 & P30 & K28 & P28 & K26 & P26 & K24 & P24 & K22 & P22 & K20 & P20 & K18 & P18 \end{pmatrix}$$

3. Apply transpositions among respective Colum

$$\begin{pmatrix} K47 & K63 & K15 & K31 & K48 & K64 & K16 & K32 \\ P47 & P63 & P15 & P31 & P48 & P64 & P16 & P32 \\ K45 & K61 & K13 & K29 & K46 & K62 & K14 & K30 \\ P45 & P61 & P13 & P29 & P46 & P62 & P14 & P30 \\ K43 & K59 & K11 & K27 & K44 & K60 & K12 & K28 \\ P43 & P59 & P11 & P27 & P44 & P60 & P12 & P28 \\ K41 & K57 & K09 & K25 & K42 & K58 & K10 & K26 \\ P41 & P57 & P09 & P25 & P42 & P58 & P10 & P26 \\ K39 & K55 & K07 & K23 & K40 & K56 & K08 & K24 \\ P39 & P55 & P07 & P23 & P40 & P56 & P08 & P24 \\ K37 & K53 & K05 & K21 & K38 & K54 & K06 & K22 \\ P37 & P53 & P05 & P21 & P38 & P54 & P06 & P22 \\ K35 & K51 & K03 & K19 & K36 & K52 & K04 & K20 \\ P35 & P51 & P03 & P19 & P36 & P52 & P04 & P20 \\ K33 & K49 & K01 & K17 & K34 & K50 & K02 & K18 \\ P33 & P49 & P01 & P17 & P34 & P50 & P02 & P18 \end{pmatrix}$$

4. Execute Column Swapping. That is Swap the columns 1and 2, 3 and 4, 5 and 6, and 7 and 8.

$$\begin{pmatrix} K63 & K47 & K31 & K15 & K64 & K48 & K32 & K16 \\ P63 & P47 & P31 & P15 & P64 & P48 & P32 & P16 \\ K61 & K45 & K29 & K13 & K62 & K46 & K30 & K14 \\ P61 & P45 & P29 & P13 & P62 & P46 & P30 & P14 \\ K59 & K43 & K27 & K11 & K60 & K44 & K28 & K12 \\ P59 & P43 & P27 & P11 & P60 & P44 & P28 & P12 \\ K57 & K41 & K25 & K09 & K58 & K42 & K26 & K10 \\ P57 & P41 & P25 & P09 & P58 & P42 & P26 & P10 \\ K55 & K39 & K23 & K07 & K56 & K40 & K24 & K08 \\ P55 & P39 & P23 & P07 & P56 & P40 & P24 & P08 \\ K53 & K37 & K21 & K05 & K54 & K38 & K22 & K06 \\ P53 & P37 & P21 & P05 & P54 & P38 & P22 & P06 \\ K51 & K35 & K19 & K03 & K52 & K36 & K20 & K04 \\ P51 & P35 & P19 & P03 & P52 & P36 & P20 & P04 \\ K49 & K33 & K17 & K01 & K50 & K34 & K18 & K02 \\ P49 & P33 & P17 & P01 & P50 & P34 & P18 & P02 \end{pmatrix}$$

5. Apply Permutation function. That is column 8 forms rows 1 and 2, column 7 forms rows 3 and 4, column 6 forms rows 5 and 6 in that order.

$$\begin{pmatrix} P02 & K02 & P04 & K04 & P06 & K06 & P08 & K08 \\ P10 & K10 & P12 & K12 & P14 & K14 & P16 & K16 \\ P18 & K18 & P20 & K20 & P22 & K22 & P24 & K24 \\ P26 & K26 & P28 & K28 & P30 & K30 & P32 & K32 \\ P34 & K34 & P36 & K36 & P38 & K38 & P40 & K40 \\ P42 & K42 & P44 & K44 & P46 & K46 & P48 & K48 \\ P50 & K50 & P52 & K52 & P54 & K54 & P56 & K56 \\ P58 & K58 & P60 & K60 & P62 & K62 & P64 & K64 \\ P01 & K01 & P03 & K03 & P05 & K05 & P07 & K07 \\ P09 & K09 & P11 & K11 & P13 & K13 & P15 & K15 \\ P17 & K17 & P19 & K19 & P21 & K21 & P23 & K23 \\ P25 & K25 & P27 & K27 & P29 & K29 & P31 & K31 \\ P33 & K33 & P35 & K35 & P37 & K37 & P39 & K39 \\ P41 & K41 & P43 & K43 & P45 & K45 & P47 & K47 \\ P49 & K49 & P51 & K51 & P53 & K53 & P55 & K55 \\ P57 & K57 & P59 & K59 & P61 & K61 & P63 & K63 \end{pmatrix}$$

6. Execute Division function to again break the matrix

$$\begin{pmatrix} P02 & K02 & P04 & K04 & P06 & K06 & P08 & K08 \\ P10 & K10 & P12 & K12 & P14 & K14 & P16 & K16 \\ P18 & K18 & P20 & K20 & P22 & K22 & P24 & K24 \\ P26 & K26 & P28 & K28 & P30 & K30 & P32 & K32 \\ P34 & K34 & P36 & K36 & P38 & K38 & P40 & K40 \\ P42 & K42 & P44 & K44 & P46 & K46 & P48 & K48 \\ P50 & K50 & P52 & K52 & P54 & K54 & P56 & K56 \\ P58 & K58 & P60 & K60 & P62 & K62 & P64 & K64 \\ P01 & K01 & P03 & K03 & P05 & K05 & P07 & K07 \\ P09 & K09 & P11 & K11 & P13 & K13 & P15 & K15 \\ P17 & K17 & P19 & K19 & P21 & K21 & P23 & K23 \\ P25 & K25 & P27 & K27 & P29 & K29 & P31 & K31 \\ P33 & K33 & P35 & K35 & P37 & K37 & P39 & K39 \\ P41 & K41 & P43 & K43 & P45 & K45 & P47 & K47 \\ P49 & K49 & P51 & K51 & P53 & K53 & P55 & K55 \\ P57 & K57 & P59 & K59 & P61 & K61 & P63 & K63 \end{pmatrix} \Rightarrow$$

$$(a) \begin{pmatrix} P02 & K02 & P04 & K04 & P06 & K06 & P08 & K08 \\ P10 & K10 & P12 & K12 & P14 & K14 & P16 & K16 \\ P18 & K18 & P20 & K20 & P22 & K22 & P24 & K24 \\ P26 & K26 & P28 & K28 & P30 & K30 & P32 & K32 \\ P34 & K34 & P36 & K36 & P38 & K38 & P40 & K40 \\ P42 & K42 & P44 & K44 & P46 & K46 & P48 & K48 \\ P50 & K50 & P52 & K52 & P54 & K54 & P56 & K56 \\ P58 & K58 & P60 & K60 & P62 & K62 & P64 & K64 \end{pmatrix}$$

$$(b) \begin{pmatrix} P01 & K01 & P03 & K03 & P05 & K05 & P07 & K07 \\ P09 & K09 & P11 & K11 & P13 & K13 & P15 & K15 \\ P17 & K17 & P19 & K19 & P21 & K21 & P23 & K23 \\ P25 & K25 & P27 & K27 & P29 & K29 & P31 & K31 \\ P33 & K33 & P35 & K35 & P37 & K37 & P39 & K39 \\ P41 & K41 & P43 & K43 & P45 & K45 & P47 & K47 \\ P49 & K49 & P51 & K51 & P53 & K53 & P55 & K55 \\ P57 & K57 & P59 & K59 & P61 & K61 & P63 & K63 \end{pmatrix}$$

7. Execute Column Swapping that is swap columns 1 of (a) and 2 of (b), 3 of (a) and 4 of (b), 5 of (a) and 6 of (b), and 7 of (a) and 8 of (b).

$$\begin{pmatrix} K01 & K02 & K03 & K04 & K05 & K06 & K07 & K08 \\ K09 & K10 & K11 & K12 & K13 & K14 & K15 & K16 \\ K17 & K18 & K19 & K20 & K21 & K22 & K23 & K24 \\ K25 & K26 & K27 & K28 & K29 & K30 & K31 & K32 \\ K33 & K34 & K35 & K36 & K37 & K38 & K39 & K40 \\ K41 & K42 & K43 & K44 & K45 & K46 & K47 & K48 \\ K49 & K50 & K51 & K52 & K53 & K54 & K55 & K56 \\ K57 & K58 & K59 & K60 & K61 & K62 & K63 & K64 \end{pmatrix} \begin{pmatrix} P01 & P02 & P03 & P04 & P05 & P06 & P07 & P08 \\ P09 & P10 & P11 & P12 & P13 & P14 & P15 & P16 \\ P17 & P18 & P19 & P20 & P21 & P22 & P23 & P24 \\ P25 & P26 & P27 & P28 & P29 & P30 & P31 & P32 \\ P33 & P34 & P35 & P36 & P37 & P38 & P39 & P40 \\ P41 & P42 & P43 & P44 & P45 & P46 & P47 & P48 \\ P49 & P50 & P51 & P52 & P53 & P54 & P55 & P56 \\ P57 & P58 & P59 & P60 & P61 & P62 & P63 & P64 \end{pmatrix}$$

8. Reiterate step 1 to 8 till Avg (Enc_Number, Ran_Number).
9. Text

$$\text{Key Matrix (KM)} = \begin{pmatrix} K01 & K02 & K03 & K04 & K05 & K06 & K07 & K08 \\ K09 & K10 & K11 & K12 & K13 & K14 & K15 & K16 \\ K17 & K18 & K19 & K20 & K21 & K22 & K23 & K24 \\ K25 & K26 & K27 & K28 & K29 & K30 & K31 & K32 \\ K33 & K34 & K35 & K36 & K37 & K38 & K39 & K40 \\ K41 & K42 & K43 & K44 & K45 & K46 & K47 & K48 \\ K49 & K50 & K51 & K52 & K53 & K54 & K55 & K56 \\ K57 & K58 & K59 & K60 & K61 & K62 & K63 & K64 \end{pmatrix}$$

$$\text{Plain Text Matrix (PT)} = \begin{pmatrix} P01 & P02 & P03 & P04 & P05 & P06 & P07 & P08 \\ P09 & P10 & P11 & P12 & P13 & P14 & P15 & P16 \\ P17 & P18 & P19 & P20 & P21 & P22 & P23 & P24 \\ P25 & P26 & P27 & P28 & P29 & P30 & P31 & P32 \\ P33 & P34 & P35 & P36 & P37 & P38 & P39 & P40 \\ P41 & P42 & P43 & P44 & P45 & P46 & P47 & P48 \\ P49 & P50 & P51 & P52 & P53 & P54 & P55 & P56 \\ P57 & P58 & P59 & P60 & P61 & P62 & P63 & P64 \end{pmatrix}$$

10. End

Flow chart of proposed encryption algorithm

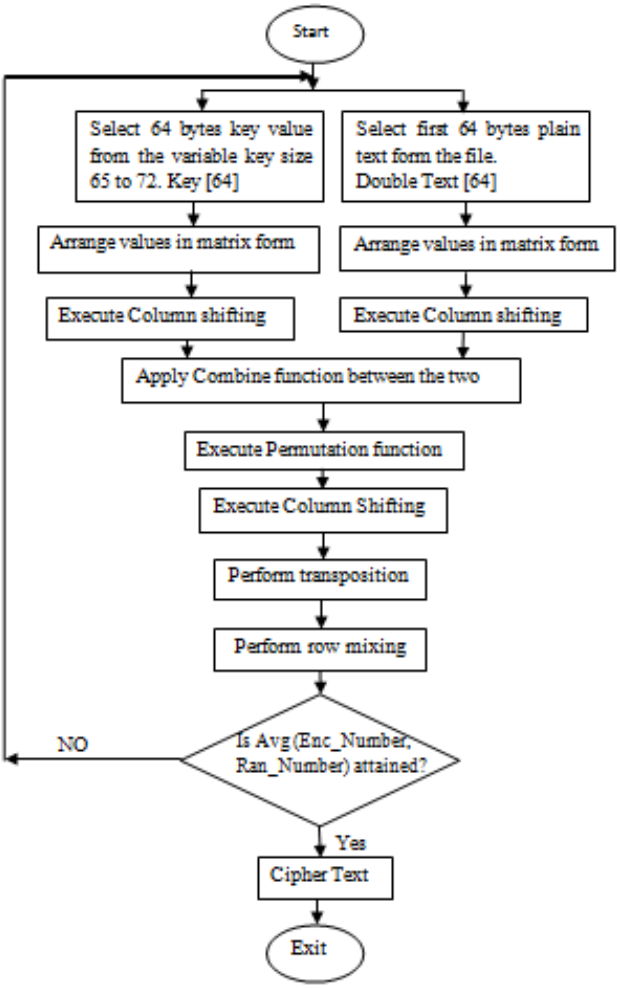

Figure-3. Flow Chart of Proposed Encryption algorithm

Flow Chart of Proposed Decryption algorithm

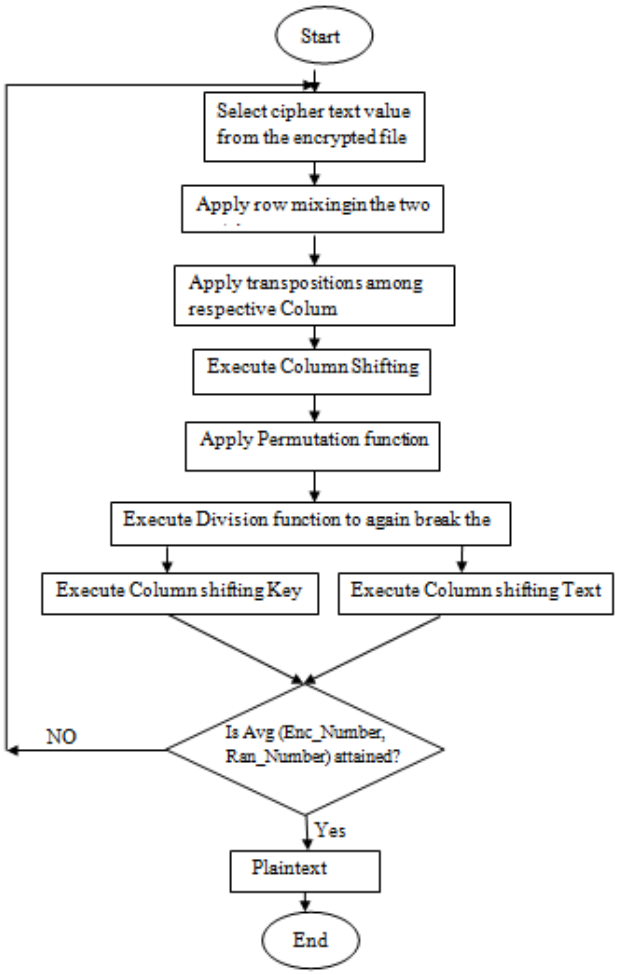

Figure 4. Flow Chart of Proposed Decryption algorithm

Cryptanalysis on the algorithm

In the Cryptography literature, the general types of attacks on a cipher are:

1. Brute force attack
2. Known plaintext attack
3. Chosen plaintext attack
4. Chosen cipher text attack

In the cipher under consideration, the keys K and plain text P both put together are containing $2n^2$ numbers, wherein each number can be represented in terms of some binary bits. Thus the key space is of size

$2^{64n^2} = 2^{10\times 6.4n^2}$

$\approx 10^{19.3^n}$

For instance, the execution of the cipher takes 10-7 seconds with a specified pair of values of the keys, for the brute force attack the time required is attained by

$$\frac{10^{19.3^n} \times 10^{-7}}{365 \times 24 \times 60 \times 60}$$

$= 57 \times 10^{19.3^n} \times 10^{-15}$

$= 57 \times 10^{(19.3^n)-15}$

From the above formula it presents that the required time for execution is several years when n>=2. With the help of cipher text only attack it cannot be broken. In the case of known plaintext attack, it has as many plaintext and cipher text pairs as require for attack. In this analysis, as the plaintext pass through several transformations on account of multiplication and addition or division by the permutation and key matrices, in every round of the iteration, before the plaintext becomes the cipher text, the on linearity include in the process does not allow anybody to break the cipher. It is impossible to choosing a plaintext or a cipher text from one as the process involved in the cipher is a complex one. With the help of chosen plaintext/cipher text attack, cipher cannot be broken. In the light of the above discussion, conclude that the cipher is a strong one.

Methodology and simulation setting

The widespread availability of inexpensive computing power is having a major impact on data analysis.

Computers allow us to carry out calculations and displays of data that were literally unthinkable only a decade ago. This will have a profound impact on the design of computer systems: an integral part of the design will be data gathering and analysis tools to determine system performance. As more and more data is gathered on each design, iterations will be carried out based on inferences from data.

Digital simulation provides a useful and effective adjunct to direct analytical evaluation of communication system performance. Indeed, there are many situations where explicit performance evaluation defies analysis and meaningful results can be obtained only through either actual prototype hardware and software evaluation or digital computer simulations.

Simulation moreover, frees the analyst from a great deal of repetitive work involved in substituting numbers into formulae and tables and enables the analyst to concentrate on results. Another advantage is the insight into system performance provided, both by the modeling process itself and by the experience gained from simulation experiments. Again, computers can assist us: it may be quite difficult or expensive to gather data or measurements, so computer assisted analysis can quantify the importance of the data gathering and analysis procedures.

Research design

Experimental research design was used in this work to manipulate some of the independent variables in order to observe the final results of the experiment.

The experiments were performed couple of times to assure that the results were consistent and were valid to ensure effective evaluation of the proposed algorithm and the other algorithms (DES, Triple-DES, Blowfish and AES).

All the implementations were accurate to make sure that the results are relatively fair and truthful.

Variables measured

There is no limit to the number of variables that can be measured, although the more variables, the more complex the study and the more complex the statistical analysis. The following variables were taken care of in this paper: dependent, independent, moderator, Control, and confounding variables (extraneous and intervening).

For the experiment researchers collected the following performance metrics:-

Avalanche effect

A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the cipher text. In, particular a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the cipher texts.

Encryption time

Encryption time is the total time taken to produce a cipher text from plain text. The calculated encryption time is then used to calculate the throughput of the encrypted algorithm. It gives the rate of

encryption. . The more the encryption time more will the power consumption and speed will be less. The powerful processors consume more power in the key generation process as a result node capacitance, charge sharing and leakage current exist in the model. These parameters are responsible for the loss of data and cause station failure. The throughput of the encryption scheme will be calculated as the total encrypted plaintext in bytes divided by the encryption time.

Decryption time

Decryption time is the total time taken to produce the plain text from plain text. The calculated decryption time is then used to calculate the throughput of the decrypted algorithm. It gives the rate of decryption. More decryption time more will the power consumption and speed will be less. The throughput of the decryption scheme is calculated as the total decrypted plaintext in bytes divided by the decryption time.

Memory required for implementation

Different encryption techniques require different memory size for implementation. This memory requirement depends on the number of operations to be done by the algorithm. It is desirable that the memory required should be as small as possible.

The CPU process time

The CPU process time is the time that a CPU is committed only to the particular process of calculations. It reflects the load of the CPU. The more CPU time is used in the encryption process, the higher is the load of the CPU.

The time was in milliseconds and the text file size was in kilobytes.

As state in the literature, execution time of algorithm directly depends on the functionality of the algorithm and it is clearly defines that more complex structure originates poor execution time. Security of the data directly depends on the key length, higher key length will provide higher security but it can increase the execution time of the algorithm so it is very important that what should be the key length and how execution time got controlled, if selected key length is higher[3].

Experimental design for metric of proposed system

For the experiment, authors used a laptop with following the System configuration:-

Operating System	-	Windows 8
Clock Speed	-	1.6GHz
Processor	-	AMD Athlon TF-20
RAM Capacity	-	3.0GB
Hard Disk	-	150 GB
Monitor	-	SVGA Color
System Bus	-	64 Bits

In which performance data is collected. In the experiments, the laptop is used for encryption and decryption of different file size ranges from 20 Kilobytes to 990 Bytes for text data. Authors have used Java SE Run-time Environment ("JRE") 1.7 and Java SDK 1.7 and NetBeans IDE 7.3. These implementations are thoroughly tested and are optimized to give the maximum performance for the algorithms.

Simulation tools for data collections

Netbeans profiler was used for data collection.

[3]Rajni J. and Ajit S., (2012), Design and Implementation of New Encryption algorithm to Enhance Performance Parameters, IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 4, Issue 5, PP 33-39

The simulation was the provided classes in java environment to simulate the performance of the proposed algorithm, DES, 3DES, AES and Blowfish. The implementation uses managed wrappers for the Algorithms available in java.cypto and java.security [CryptoSpec] that wraps unmanaged implementations available in JCE (Java Cryptography Extension) & JCA (Java Cryptography Architecture). The Cipher class provides the functionality of a cryptographic cipher used for encryption and decryption. It forms the core of the JCE framework.

The Simulation program accepts the inputs: Algorithm. After a successful execution, the data generated, encrypted, and decrypted.

Simulation result analysis and discussion

Avalanche effect

The simulation results for this evaluation shown in figure 5 and table 1 below, it is clear that, based on the simulation results; avalanche effect is highest in AES. It is medium in DES, Triple-DES and Blowfish. It is smallest in the propose algorithm. Therefore, if one desires a good avalanche effect; AES is the best option not the proposed algorithm. According to the graph, there is a tendency that the avalanche effect increases with file size.

Table-1. Evaluation of the techniques based on Avalanche effect

Technique	1 bit variation in key keeping plain text constant	1 bit variation in plain text keeping key constant
Proposed Algorithm	3	5
DES	30	35
Triple-DES	37	34
AES	65	78
Blowfish	36	24

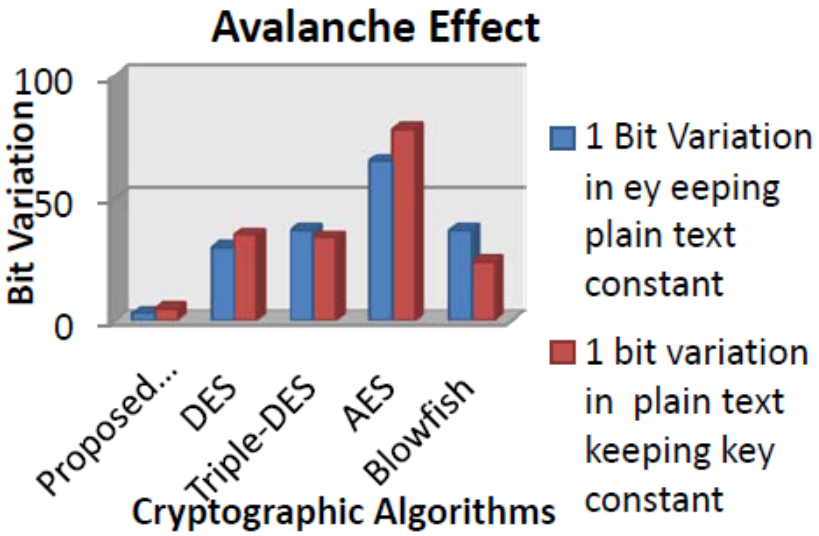

Figure 5. The simulation results avalanche effect

Encryption time

Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. Different text sizes are used in this experiment for the propose algorithm, DES, Triple-DES, Blowfish and AES. The encryption time is recorded for these encryption algorithms. The average data rate is calculated for these algorithms based on the recorded data. The formula used for calculating average data rate is

$$\text{AvgTime} = I/Nb \sum_{I=0}^{Nb} \quad \text{Mi/ti(Kb/s)}$$

Where
AvgTime = Average Data Rate (Kb/s)
Nb = Number of Text
Mi= Text Size (Kb)
ti=Time taken to Encrypt Text Mi

Encryption Throughput:-Encryption throughput and power consumption are inversely proportional to each other. As the encryption throughput value is increased the power consumption of the following encryption is decreased and this "encryption throughput "can be evaluated by dividing the total plaintext in megabytes encrypted on the total encryption time for each algorithm which is ready for encryption or in the process of encryption or encrypted. This can be illustrated from the following

Encryption time is used to compute the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated using the formula

Throughput = $\frac{Tp}{Et}$

Where Tp= Total Plain text and Et= Encryption time

It is very important to calculate the throughput time for the encryption algorithm to known better performance of the algorithm. Encryption Time Comparison between Proposed Algorithm and the Selected Existing Algorithm on Text File is illustrated below.

Table 2. Comparative execution times (in milliseconds) of encryption algorithms with different packet size

Text File Size In Kbytes	**DES**	**Triple-DES**	**Blowfish**	**Propose Algorithm**	**AES**
20	20	34	25	23	42
48	30	55	27	26	55
100	47	81	33	30	90
247	83	111	45	43	112
321	90	167	46	44	164
694	144	226	47	45	210
899	240	230	64	56	256
910	245	299	68	63	213
Average Time	112.375	150.00	44.375	41.25	142.75
Throughput (Kilobyte/sec)	4.16	3.11	11.16	11.33	3.27

The data analyzed in the table is illustrated in the graph below.

Simulation results for this compassion point are shown Figure 6 and Table 2 at encryption stage. The results show the superiority of Propose Algorithm over other algorithms in terms of the processing time. An additional position can be noticed here; that Blowfish requires less time than the remaining algorithms: DES, AES and Triple-DES algorithms except the Propose Algorithm. A third point can be noticed here; that AES has an advantage over other DES, DES and the Proposed

Algorithm in terms of time consumption and throughput. A fourth point can be noticed here; that Triple-DES has low performance in terms of power consumption and throughput when compared with DES. It requires always more time than DES because of its triple phase encryption characteristics. Finally, it is found that Triple-DES has low performance and low throughput when compared with other five algorithms in spite of the small key size used.

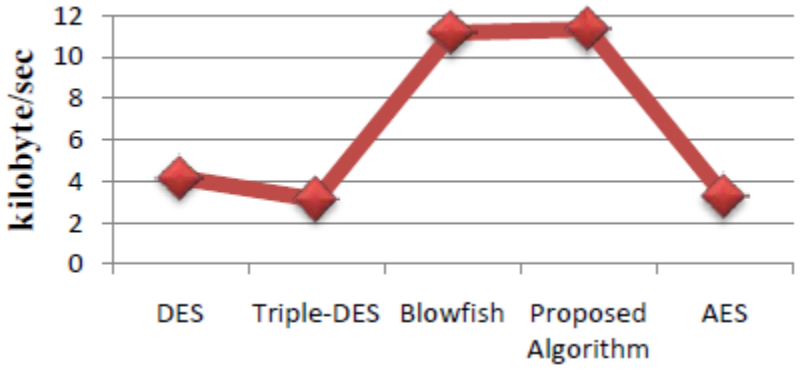

Figure 6. Throughput of each encryption algorithm (Kilobyte/Sec)

Encryption time

Table 3. Comparative Decryption Time (in milliseconds) of various algorithms with different packet size

Text File Size in Kbytes	**DES**	**Triple-DES**	**Blowfish**	**Propose Algorithm**	**AES**
20	34	40	28	25	45
48	50	53	34	29	63
100	57	57	56	54	60
247	72	77	73	70	76
321	87	87	81	79	149
694	120	146	90	88	142
899	152	171	99	99	171
910	160	173	120	112	144
Average Time	91.875	100.875	72.625	69.5	106.25
Throughput (Kilobytes/sec)	5.09	4.63	6.40	6.72	4.40

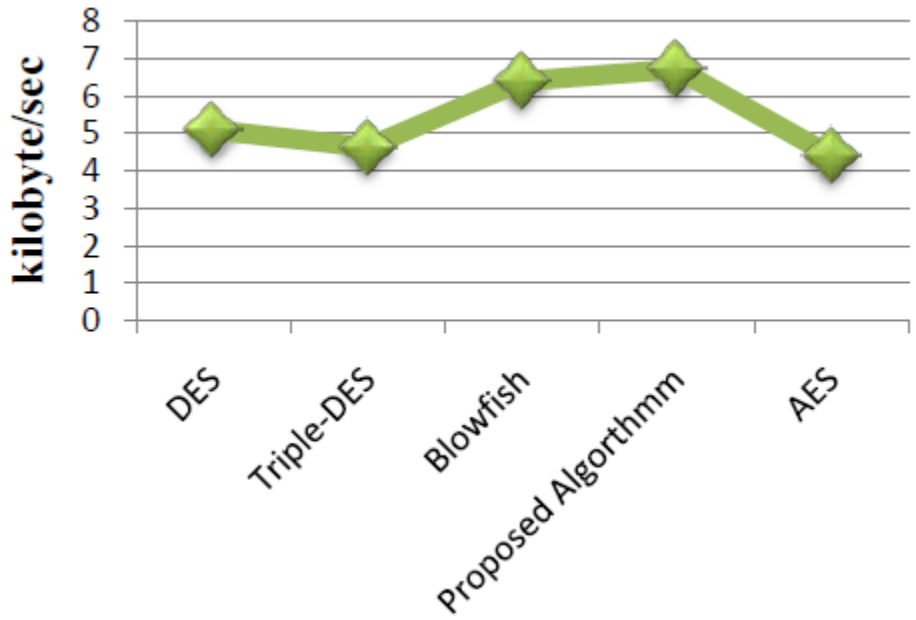

Figure 7. Throughput of each decryption algorithm (kilo-Byte/Sec)

Simulation results for this compassion point are shown Figure 7 and Table 3 decryption stage. We can find in decryption that the Proposed Algorithm is the better than other algorithms in throughput and power consumption, its throughput is 24.7% as against 23.5%, 18.7%, 17.0% and 16.2% of Blowfish, DES, Triple-DES and AES respectively. Because less the time; less will be the power consumption & more the speed of the algorithm Second point can be notice here that Blowfish has advantage over the other DES and Triple-DES in terms of processing decryption time except the Proposed Algorithm. Third point is to be noticed here that DES has a better performance than Triple-DES in terms of Decryption time. Fourth point which has been observed is that AES has least performance than all. Finally it is concluded that the proposed Algorithm is the best among them all.

Memory required for implementation

This memory requirement depends on the number of operations to be done by the algorithm. It is desirable that the memory required should be as small as possible.

Table 4. Comparison based on memory required for implementation.

Cryptographic Algorithm	Memory required for implementation (KB)
DES	12.8
Triple-DES	14.8
Blowfish	6.88
Proposed Algorithm	5.7
AES	10.6

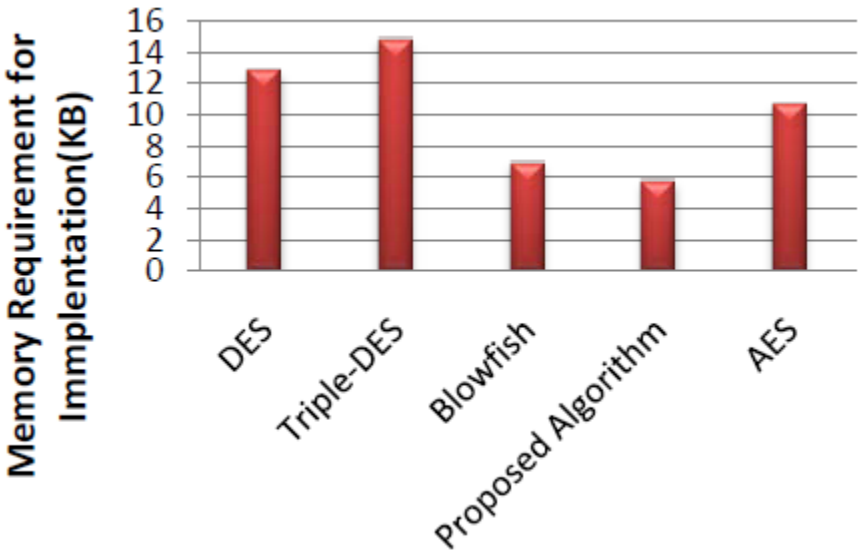

Figure 8. Bar graph illustrating Comparison based on memory required for implementation.

How much memory space is required to execute an algorithm as known as space complexity?[4] Memory space has depended upon program length, that mean memory utilization is directly depend on length of the program; if any program have used large amount of code then we can say that large memory space will required to execute. In the proposed algorithm used simple and small code.

From Table 5 and the figure 8 above, it is clear that the memory required for implementation is smallest in the proposed algorithm whereas it is largest in 3DES. DES, AES and Blowfish require medium size of memory. Therefore, if the demand of any application is the smallest memory size; the proposed algorithm is the best option.

The CPU execution time

The CPU process time is the time that a CPU is committed only to the particular process of calculations. It reflects the load of the CPU. The more CPU time is used in the encryption process, the higher is the load of the CPU. The CPU execution time in seconds for the proposed algorithm and the four algorithms (DES, Triple-DES, Blowfish and AES) were determined on file size of 100 KB.

The experiment was repeated 10 times for each algorithm and for each operation, key generation, the encryption operation and the decryption operation. The average of the 10 runs for each operation was computed for each algorithm.

The result is illustrated in the figure 9 and the table 5. In the experiment, the CPU execution time is computed for the five algorithms CPU time includes: system (kernel) time and user time. The system time is the execution time in kernel mode, and the user time is execution time in user mode.

Table 5. CPU Execution Time in Seconds for 100KB Text

Operation	Cryptographic Technique				
	DES	Triple-DES	Blowfish	Proposed Algorithm	AES
Key Generation	0.316	0.32	0.32	0.23	0.31

[4]Yan Wang and Ming Hu ”Timing evaluation of the known cryptographic algorithms” 2009 International Conference on Computational Intelligence and Security

Encryption	0.36	0.351	0.32	0.2	0.332
Decryption	0.36	351	0.32	0.2	0.332

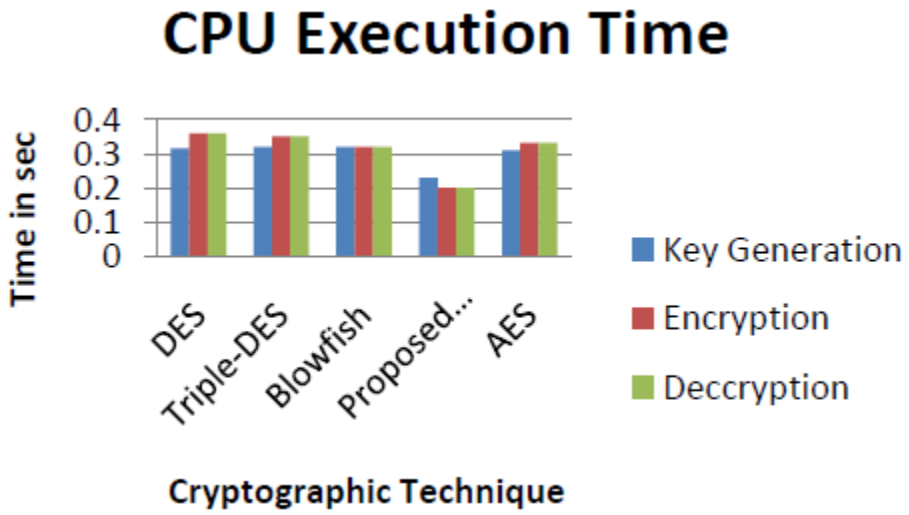

Figure 9. The CPU execution time in seconds for the five algorithms on 100KB text

The CPU execution time for the proposed Algorithm is shorter than the CPU execution time of the other algorithms by the factors shown in the graph.

Simulation time

The time required by the algorithm for processing completely a particular length of data is called the simulation time. It depends on the processor speed, complexity of the algorithm etc. The smallest value of simulation time is desired. From Table 6 the text files of different sizes/text lengths are taken.

Table 6. Comparison based on simulation time required for different length of plaintexts (in sec.)

Text Length (KB)	DES	Triple-DES	Blowfish	Proposed Algorithm	AES
0.1	2.0	6.0	4.2	0.18	3.2
0.7	5.1	16.2	6.1	0.24	5.4
3.0	59.0	332.0	10.0	0.87	14
6.0	243	1301	14.5	2.00	21.9

These are encrypted and decrypted using all the Cryptographic techniques one by one. The simulation time taken by different techniques is recorded in sec. It is clear that the Proposed Algorithm is the fastest Cryptographic technique. Triple-DES is the slowest technique. The speed of DES, AES and Blowfish is average. This is graphically shown in figure 10, where with text length 6KB the bar of Triple-DES indicate remarkable height far above other cryptographic techniques. The graph depicts that the simulation time is best on the proposed algorithm for the four types of text length.

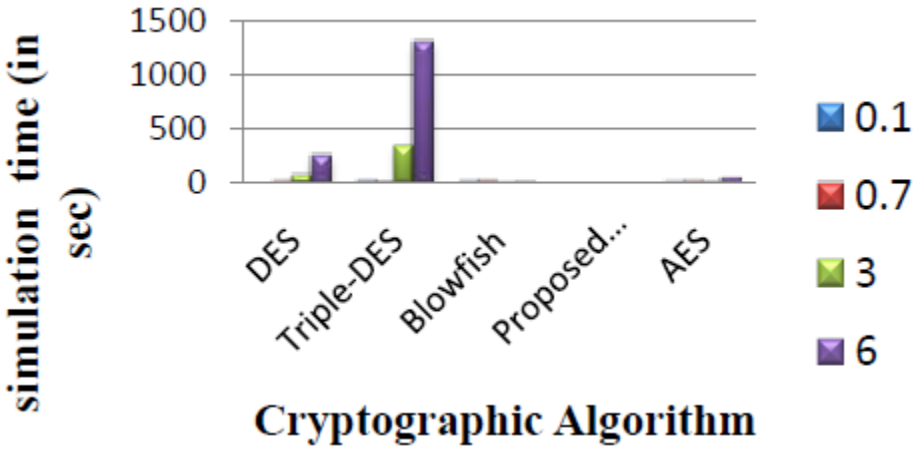

Figure 10. Comparison based on simulation time required for different length of plaintexts (in sec.)

From the simulation analysis it can be observed that it is possible to reduce the complexity of algorithms mathematical manipulations and still ensure maximum security, however the more the algorithms mathematical manipulations is simplified the less secure it becomes.

On the question; can the new encryption algorithms design and implementation to enhance performance? It has been proved yes not only that simulation results shows the possibility of comparing some of the proposed algorithm's variables with that of the tested and proving algorithms like: DES, Triple-DES, Blowfish and AES.

Conclusion

This paper presents a performance evaluation of selected Proposed Enhanced Simplified Symmetric Key Encryption Algorithm and other selected symmetric encryption algorithms on avalanche effect, time consumption for encryption and decryption. The selected algorithms are AES, DES, Triple-DES and Blowfish. Several points will include in the simulation results. First; in the case of Avalanche effect, it can be concluded that proposed algorithm produces smallest avalanche effect as compared with DES, Triple-DES, Blowfish and AES, and AES produced the highest avalanche effect, therefore AES is the best option not the proposed algorithm. Second; in the case of Encryption and Decryption Time the proposed Algorithm is the best among as compared with DES, Triple-DES, Blowfish and AES. Therefore it has found proposed algorithm produced better performance than other selected encryption algorithms used in terms of time consumption.

On Memory required for implementation, comparatively, DES, Triple-DES, Blowfish and AES each uses more memory space than the proposed algorithm. From the above stimulation results, there is possibility to reduce the complexity of algorithms mathematical manipulations and still ensure maximum security; however, the proposed algorithm is not all that simple.

Limitations of the study

Technology constriction:- The problem encountered here is securing the data through data encryption algorithms like DES, Triple-DES, AES and Blowfish Algorithms and another problem is since cryptography is depends on complex mathematical manipulations the more you are simplifying mathematical principles the weaker the resulted algorithm becomes.

Resource constriction: There was not adequate fund and computing resources to carried out cryptanalysis.

Testing constriction: Future researches can test to break the security of the proposed algorithm by writing their own code and using highly classified testing/hacking approach to perform more complex cryptanalysis.

Future enhancement

This system can be enhanced by developing a standard formula for generating the number N which determines the number of iterations that is to be carried out. Though the system is designed for storage level but the modules can be used in web services also. Security can also be enhanced by using more complex operation to increase security level.

The algorithms can be implemented by iterative model instead of using steady algorithm not only that it can use random order algorithm for compression and encryption.

References

[1]. Agrawal, M., & Mishra, P. (2012). A Comparative Survey on Symmetric Key Encryption Techniques. International Journal on Computer Science and Engineering, 877-882.

[2]. Ahmad, S., Alam, K. M., Rahman, H., & Tamura, S. (2015). A comparison between symmetric and asymmetric key encryption algorithm based decryption mixnets. Networking Systems and Security (NSysS), 2015 International Conference on (pp. 1-5). Dhaka, Bangladesh: IEEE. doi: DOI: 10.1109/NSysS.2015.7043532

[3]. Ajay K., S. M. (January, 2012). Comparison of Various Encryption Algorithms and Techniques for Secured Data Communication in Multinode Network. International Journal of Engineering and Technology Volume 2 No. 1, ISSN: 2049-3444 © 2011 – IJET Publications UK., 87-92.

[4]. Alabaichi, A., Ahmad, F., & Mahmod, R. (2013). Security analysis of blowfish algorithm. Informatics and Applications (ICIA), 2013 Second International Conference on (pp. 12-18). Malaysia: IEEE. doi:10.1109/ICoIA.2013.6650222

[5]. Al-Hazaimeh, O. M. (March 2013). A New Approach for Complex Encrypting and Decrypting Data. International Journal of Computer Networks & Communications (IJCNC) Vol.5, No.2, 95-103.

[6]. Ankita P. B., L. S. (April - 2013). A Comparative Literature Survey On Various Image Encryption Standards. International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 4, ISSN: 2278-0181, 1444-1450.

[7]. Klinc, D., Hazay, C., Jagmohan, A., Krawczyk, H., & Rabin, T. (2012, 11). On Compression of Data Encrypted With Block Ciphers. IEEE Transactions on Information Theory, Vol-58(Issue-11), 6989 - 7001. doi: DOI: 10.1109/TIT.2012.2210752

[8]. Mandar M. K., P. B. (March 2013). Encryption Algorithm Addressing GSM Security Issues- A Review. International Journal of Latest Trends in Engineering and Technology (IJLTET), Vol. 2 Issue 2, ISSN: 2278-621X, 268-273.

[9]. Paar C. and Pelzl J. (2010). Understanding Cryptography: A Textbook for Students and Practitioners. ISBN 978-3-642-04101-3.

[10]. Shah K. R., B. G. (March, 2012). New Approach of Data Encryption Standard. International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-2, Issue-1, 322-325.

[11]. Sharma, H. A. (2010). Implemenentation and analysis various symmetric cryptosystems. Indian Journal of Science and Technology Vol. 3 No. 12 ISSN:0974-6846, 1173-1176.

[12]. Shrivas, M. K., & Singh, S. V. (2014, June). Added Advanced Encryption Standard (A-Aes): With 512 Bits Data Block And 512, 768 And 1024 Bits Encryption Key. International Journal of ICT and Management, Vol-II (Issue-1), 65-71. Retrieved 2016, from http://www.ijictm.org/admin/html/mail/attach/2014-08-06-03-48-23.pdf

[13]. Shrivas, M. K., & Singh, S. V. (2015). Implementing ADDED ADVANCED ENCRYPTION STANDARD (A-AES) to Secure Data on the Cloud. 3rd International Conference on Management, Communication and Technolog (ICMCT) (pp. 17-24). Accra, Ghana: IJICTM. Retrieved 2016, from http://www.ijictm.org/admin/html/mail/attach/2015-08-07-09-55-57.pdf

[14]. Shrivas, M. K., Dr. Amoako, A., Boateng, S. O., & Dr. Yeboah, T. (2015, 10). Migration Model for unsecure Database driven Software System to Secure System using Cryptography. The International Journal of ICT and Management, Vol-II (Issue-2), 1-8. Retrieved 2016, from http://www.ijictm.org/admin/html/mail/attach/2015-12-10-08-23-09.pdf

[15]. Vikendra S., a. S. ((2013)). ANALYSING SPACE COMPLEXITY OF VARIOUS ENCRYPTION ALGORITHMS. International Journal of Computer Engineering and Technology (IJCET), ISSN 0976-6367(Print), ISSN 0976 – 6375(Online) Volume 4, Issue , 414-419.

[16]. Yogesh K., R. M. (Oct 2011). Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures. IJCSMS International Journal of Computer Science and Management Studies, Vol. 11, Issue 03, ISSN (Online): 2231-5268, 60-63.