Evolution of New Cryptography Algorithm (An Approach for Data Security)

Mohd. Rashid[#1] , Jayanti Jharia[#2] , Manju Sahu[#3]
[#]Dept. of Computer Science, Career College, Bhopal, India
[1]mohd.rashid09@gmail.com
[2]mehra.jayanti109@gmail.com
[3]manjusahu87@gmail.com

Abstract: **A recent field of study is the security of the information over public network through encryption in efficient way. Even if there are no efficiency gains to be made, there are practical and usability issues. This paper is describing a concept of proposed encryption/decryption technique. In this we are fulfilling basic security principle like confidential, and authenticity. In this we have developed encryption/decryption algorithm with 128 bits key length to improve the security. Presented results are showing the performance of the proposed concept in terms of efficiency and security.**

Keywords: **Computer Security, Network, Encryption, Decryption, Algorithm, Cryptography, Symmetric Key**

I. INTRODUCTION

There are three type of cryptography algorithm like public key, symmetric key algorithms, and hash functions. While the first two algorithms are used for encryption and decryption of the data, and the hash functions are one-way functions that don't allow the reverse processed. As we know that encryption algorithms are used in computer communications or exchanging information in network to provide secure transfers. Whenever an algorithm is used in a transfer, the file is first translated into a meaningless cipher text and then transferred; at the receiving end, computer uses a key value to translate the cipher into its original form. So if the data or file is intercepted before it reaches the receiving end computer it is in an unusable (or encrypted) form [1]. Cryptography process can be control through key where it is a piece of information and permits an encrypted string to be decoded. In fact, the key we chose will provide the only means to decrypt data that was encrypted with that key, so not only must we choose the key carefully, we must never change it if we intend use it for persistent data. It goes without saying that we should guard our key carefully. If someone gains access to our key, the data will be easily decoded [2]. If our server is not totally under our control it's impossible to ensure key security so we may want to think carefully before using it for anything that requires high security, like storing credit card numbers. To take maximum advantage of the encryption algorithm, our key should be 16 characters in length (128 bits). The key should be as random a string as we can concoct, with numbers and uppercase and lowercase letters. Our key should not be a simple text string. In order to be cryptographically secure it needs to be as random as possible [2, 3].

In this paper we propose a new encryption /decryption algorithm with 128 bits key length that focus on the security enhancement of existing algorithm. Certain modifications are proposed in encryption/decryption algorithm where original message will be encrypt by proposed encryption algorithm and at receiving end it will decrypt by proposed decryption algorithm.

Organization of the paper is as follow: section two is the proposed work, section third is the result analysis and finally section four is the conclusion and future enhancement.

II. PROPOSED WORK

The proposed algorithm is a block cipher that divides data into blocks of equal length and then encrypts each block using a special logical operation and Key. This algorithm uses symmetric key technique for encoding and decoding of data i.e. it uses the same key at both ends. We can use additional security measures that used in algorithm to change the format of key while sending it from one end to another. Thus, the key distribution predicament can be handled easily but this is not our task. Another plus point of proposed algorithm is that it protects the cipher text from Brute-force attacks as the 128 bits long key length used in the encryption process. Thus even knowing or decrypting key, it will be very hard to attain plaintext from cipher text.

A. Encryption

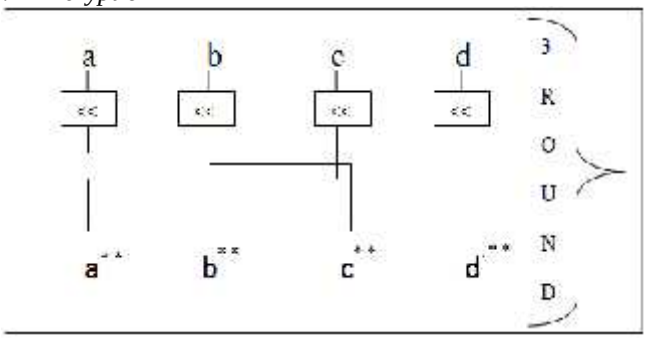

Fig. 1 (a): Proposed Encryption Block-1

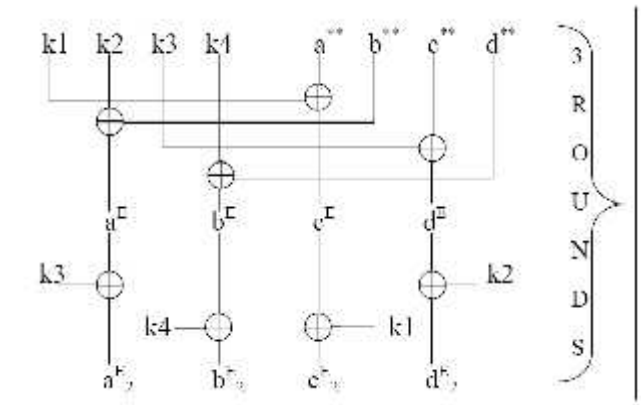

Fig. 1 (b): Proposed Encryption Block-2

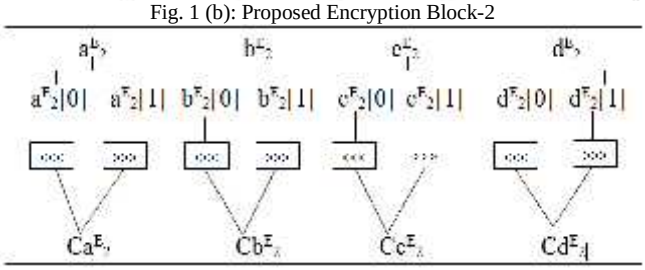

Fig. 1 (c): Proposed Encryption Block-3

B. Proposed Encryption Algorithm

1) Initially divide the input message into blocks, each of length 128 bits, if the message is not a multiple of 128 bits than insert the padding bits zero's at the starting of message and make it equal to multiple of 128 bits block. Now, select a block of 128 bits one by one from input file to be encrypt
2) These block will divide into 4 sub block like a, b, c, d Each sub block is of 32 bits.
3) Rearrange these sub-block like a will become d^* with 2 bits left circular shift, b will become c^* with 2 bits left circular shift, c will become a^* with 2 bits left circular shift and d will become b^* with 2 bits left circular shift.
4) Repeat Step 3rd 3 times. After that result will be a_i^{**} , b_i^{**}, c_i^{**}, d_i^{**} where (i = 0,1,2,3,……n, n is the total number blocks of 128 bits)
5) Select key of 16 byte (Characters, numbers, special characters).
6) Selected will divided into 4 sub-keys like k_1, k_2, k_3, k_4.
7) Apply XOR operation between sub-key and sub-block first time.
   i) K_1 $a_i^{**} \rightarrow a^E$
   ii) K_2 $b_i^{**} \rightarrow b^E$
   iii) K_3 $c_i^{**} \rightarrow c^E$
   iv) K_4 $d_i^{**} \rightarrow d^E$
8) apply XOR operation between sub-key and sub-block second time in different way like
   i) K_1 $c^E \rightarrow a^E{}_2$
   ii) K_2 $d^E \rightarrow b^E{}_2$
   iii) K_3 $a^E \rightarrow c^E{}_2$
   iv) K_4 $b^E \rightarrow d^E{}_2$
9) Step 7th and 8th repeat 3 times. After that result will be $a_i^E{}_2$, $b_i^E{}_2$, $c_i^E{}_2$, $d_i^E{}_2$, where (i = 0,1,2,3,……n)
10) Apply right circular shift operation on $a_i^E{}_2$ [0] , $b_i^E{}_2$[0] , $c_i^E{}_2$[0], $d_i^E{}_2$ [0] and Apply left circular shift operation on $a_i^E{}_2$ [1] , $b_i^E{}_2$[1] , $c_i^E{}_2$[1], $d_i^E{}_2$ [1] with 3 bits.
11) Finally cipher text will produced like $Ca_i^E{}_2$, $Cb_i^E{}_2$, $Cc_i^E{}_2$, $Cd_i^E{}_2$.
12) Exit

C. Decryption

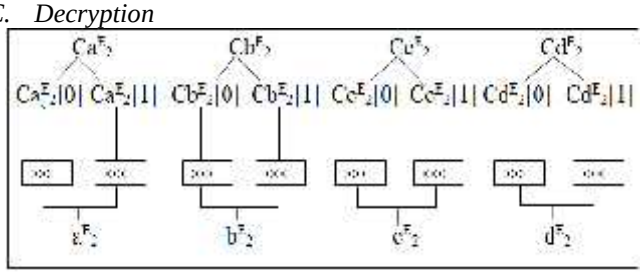

Fig. 2 (a): Proposed Decryption Block-1

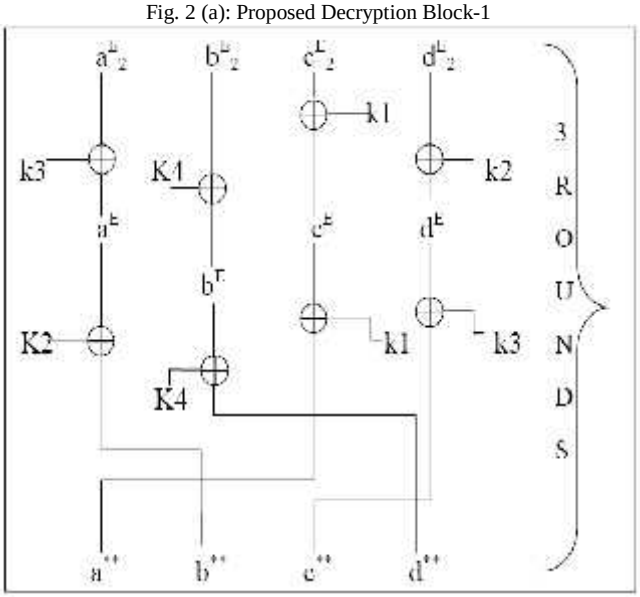

Fig. 2 (b): Proposed Decryption Block-2

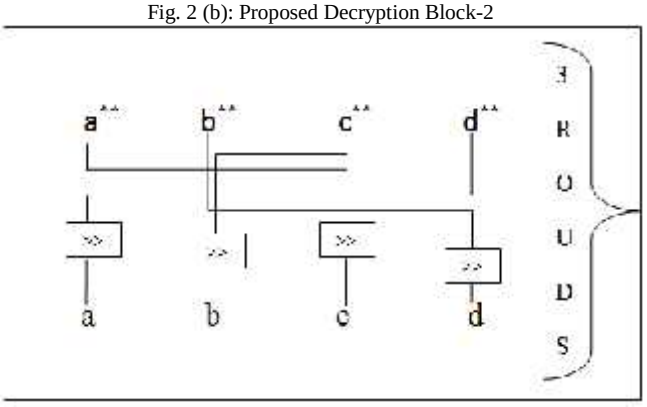

Fig. 2 (c): Proposed Decryption Block-3

D. Proposed Decryption Algorithm

1) Select cipher text of 16 byte from encrypted file. i.e. $Ca_i^E{}_2$, $Cb_i^E{}_2$, $Cc_i^E{}_2$, $Cd_i^E{}_2$.
2) Apply reverse right circular shift operation on$Ca_i^E{}_2$ [0] , $Cb_i^E{}_2$[0] , $Cc_i^E{}_2$[0], $Cd_i^E{}_2$ [0] and Apply reverse left circular shift operation on $Ca_i^E{}_2$ [1] , $Cb_i^E{}_2$[1] , $Cc_i^E{}_2$[1], $Cd_i^E{}_2$ [1] with 3 bits. Results will be $a_i^E{}_2$, $b_i^E{}_2$, $c_i^E{}_2$, $d_i^E{}_2$

3) Apply XOR operation between sub-key and sub-block first time.
   v) K_3 $a^E{}_2 \rightarrow a^E$
   vi) K_4 $b^E{}_2 \rightarrow b^E$
   vii) K_1 $c^E{}_2 \rightarrow c^E$
   viii) K_2 $d^E{}_2 \rightarrow d^E$
4) Apply XOR operation between sub-key and sub-block second time in different way.
   v) K_1 $a^E \rightarrow a_i^{**}$
   vi) K_2 $b^E \rightarrow b_i^{**}$
   vii) K_3 $c^E \rightarrow c_i^{**}$
   viii) K_4 $d^E \rightarrow d_i^{**}$
5) Step 4th and 5th Repeat 3 times. Result will be a_i^{**} , b_i^{**} , c_i^{**} , d_i^{**} where (i = 0,1,2,3,……n)
6) Rearrange these sub-block like d^* will become a with 2 bits right circular shift, c^* will be come b with 2 bits right circular shift, a^* will become c with 2 bits right circular shift and b^* will become d with 2 bits right circular shift.
7) Finally original text block a, b, c, d will be there.
8) Repeat the same for each 128 bits block.
9) Exit.

E. Strength of Proposed Encryption Algorithm

Apply all the permutation and combination to get actual key value is known cryptanalysis. In the proposed algorithm everything is done in binary format. Proposed key is of 128 bits long. Although, keys and data are conveyed in character mode but keys and operations are actually applied in binary format. Thus, it becomes very challenging for a Cryptanalyst to understand the underlying format and relationship between operations, functions and data. Here a worst case scenario is presented to break the cipher text, even if encryption process is known.

Text is in character format which has been encrypted using 128-bit key using some logical operation. So the conversion time for data= 128 unit.

Possible number of attempts to break the proposed Key**:** First of all, the intruder does not know about the key, as it is hidden from all the data. The work needed to get to this 128 bit long key will be: 2*2*2*2* . . . . . 128 times= 2^{128} units of time.

There are total 7 cycles like this and each contains different functionalities. So it must multiply by 3^7 also. As the total cases are 3^7.

Total effort for the primary key = $3^7 * 2^{128}$.

Potential number of attempts to reverse the Shift left Operation: As the data is 128 bits long and shift left operation is performed on this data, hence there is total 2* 128 cases possible to break the code.

III. PERFORMANCE ANALYSIS

This section is providing analysis of this algorithm on the basis of different parameters like security and efficiency. The basic feature of popular block ciphers is that they all are fully dependent on key and the key remains same for the whole plaintext. Moreover, it does different binary operations on plaintext and making it harder to crack than traditional block ciphers. Dot Net implementation has used to test these algorithms. For experiment, Intel Core i5 2.40 Ghz, 4 GB of RAM and Window-7 Home Basic SP1, have used in which performance data is collected.

A. Time Analysis:

The core advantage of any cryptographic algorithm is the speed of encoding and decoding of data. Proposed algorithm is especially designed for this feature and some time-saving coding is done. Table 1 and 2 is showing encryption and decryption time of the proposed encryption/decryption algorithm on various files size with same key value.

TABLE 1
COMPARISON OF ENCRYPTION TIME OF PROPOSED ALGORITHM ON VARIOUS FILE SIZE

File Size in KB	**Algorithm**		
	Execution Time in Second		
	Proposed Algorithm	AES	DES
5 KB	0.046	0.742	0.139
10 KB	0.081	1.772	0.241
20 KB	0.142	3.817	0.461
40 KB	0.348	8.286	0.924
80 KB	0.920	19.047	2.116

TABLE 2
COMPARISON OF DECRYPTION TIME OF PROPOSED ALGORITHM ON VARIOUS FILE SIZE

File Size in KB	**Algorithm**		
	Execution Time (in Second)		
	Proposed Algorithm	AES	DES
5 KB	0.045	0.793	0.115
10 KB	0.067	1.583	0.221
20 KB	0.117	3.214	0.432
40 KB	0.258	6.721	0.872
80 KB	0.806	13.037	1.883

A graphical representation for the table 1 and table 2 is shown in fig. 3 and fig. 4 with blue line for the proposed algorithm, red line for AES and green line for DES. According to the graph, there is a tendency that execution time for encryption/decryption algorithm, increases with file size. But required time for the execution of proposed encryption/decryption algorithm is much smaller than execution time of compared algorithms.

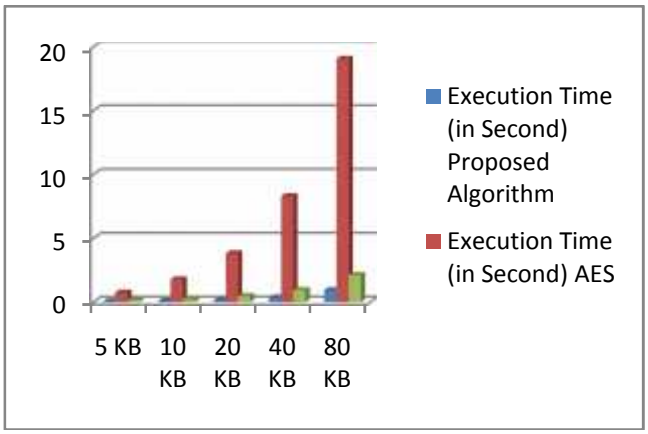

Fig. 3: Encryption time of the proposed algorithm

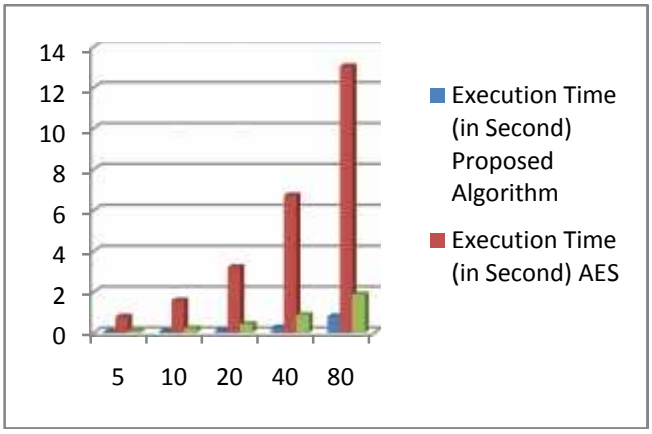

Fig. 4: Decryption time of the proposed algorithm

B. Memory Requirements:

The following table shows that memory prerequisite of proposed encryption algorithm with existing AES and DES encryption algorithms.

Algorithms Name	Key Length in Bits	Plain Text Length in Bits (Input Block)	Cipher Text Length in Bits (Output Block)
Proposed Algorithm	128	128	128
AES	128	128	128
DES	56	64	64

From the above tables it is concluded that the memory requirements of proposed algorithm are almost same to popular security algorithms AES, for the text file with addition to better defense of data.

C. Cryptanalysis:

Decryption of cipher text without prior knowledge of the key is known as Cryptanalysis [6]. In the proposed algorithm everything is done in the binary format as well as character format. Primary key is actually of 128 bits long. Although, keys and data are conveyed in character mode but keys and operations are actually applied in binary format. Thus, it becomes very challenging for a Cryptanalyst to understand the underlying format and relationship between operations, functions and data. Here a worst case scenario is presented to break the cipher text, even if encryption process is known.

1) Possible number of attempts to break the Key:

First of all, the hacker does not know about the key, as it is hidden from all the data. Even if hacker comes to know that a guessing key of 128 bits is used, still the work needed to get to this 128 bit long key will be: 2*2*2*2* . . . … . . . . . 128 times= 2^{128} units of time

Hence, total time requirement to break the key is:

- There are total 7 round processes and each 3 blocks contain the different functionalities. So we must multiply by 2^7 also.
- Total effort for the key = $2^{128}*2^7$.

2) Potential number of attempts to reverse the Shift left Operation:

As the data is 128 bits long and shift left operation is performed on this data, hence there are total 2* 128 cases possible to break the code.

IV. CONCLUSION

The proposed algorithm has been designed in a proficient approach but of- course not sacrificing the security issues. It has been successfully implemented on the text data. We have also tried to benchmark the performance of proposed algorithm against some well-known Symmetric Key Algorithms like DES, AES algorithm. The proposed algorithm is a time-efficient encryption/decryption algorithm which transfers data comparatively faster and it offers the enhanced security features than the other symmetric key algorithms. Hence this algorithm proves to be a very efficient technique for transferring messages from sender to the receiver, achieving confidentiality as well as message authentication. The proposed algorithm provides high security during the transmission, and making it least vulnerable to different attacks.

Future development will include:

- Implementation of Proposed Algorithm for different type of data.
- Hardware compatibility of the proposed algorithm.

REFERENCES

[1] Ashwak M. AL-Abiachi, Faudziah Ahmad, Ku Ruhana A Competitive Study of Cryptography Techniques over Block Cipher" IEEE UKSim 13th International Conference on Modelling and Simulation 2011

[2] Akhil Kaushik, Manoj Bamela and AnantKumar "Block Encryption Standard for Transfer of Data" IEEE International Conference on Networking and Information Technology 2010

[3] G. RAMESH and Prof. Dr. R. UMARANI "UMARAM: A Novel Fast Encryption Algorithm for Data Security in Local Area Network" IEEE ICCCCT'2010

[4] P.P Charles & P.L Shari, "Security in Computing: 4th edition", Prentice-Hall, lnc.,2008.

[5] Cryptography and network Security Principles and Practices, Charles Fleeger

[6] William Stallings, "Network Security Essentials (Applications and Standards)", Pearson Education, 2004.