IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. VI (Feb. 2014), PP 29-34
www.iosrjournals.org

A Novel Approach to Design Time Efficient and Secure encryption Algorithm (T-SEA)

Prachi Saxena, Sini Shibu
M.Tech. Scholar, Department of CSE NRI, Bhopal (M.P.), INDIA
Asst. Professor, Department of CSE NRI, Bhopal (M.P.), INDIA

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 proposing a new encryption/decryption technique and fulfilling the basic security principle like confidential and authenticity. This paper developed an algorithm with variable 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 different types 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 to provide confidentiality on the data, and the hash functions are one-way functions that don't allow the reverse process, used to provide integrity on the data. Encryption algorithms are used in computer communications or exchanging information in network to provide secure transfers. Whenever there is a requirement to transfer of secret message or data, a file containing a secret message is first translated in to a meaningless cipher text and then transferred. The conversion of secret message called plaintext to cipher text is done using a value called key; at the receiving end, computer uses a key value to translate the cipher into its original form. If key used at both the end are same than it is called symmetric key algorithm and if both the keys are different than the algorithm is called public key algorithm. If the data or file is intercepted before it reaches the receiving end computer, it is in an unusable (or encrypted) form [1]. The complete cryptography process is control through key where it is a piece of information and permits an encrypted string to be decoded. In fact, the key provides the only means to decrypt data that was encrypted, so not only chosen the key carefully is important, keeping it secret is also very important. If someone gains access to the key, the data will be easily decoded [2]. To take maximum advantage of the encryption algorithm, the key should be as random a string as it can concoct, with numbers and uppercase and lowercase letters. Key should not be a simple text string. In order to be cryptographically secure it needs to be as random as possible [2, 3].

This paper proposed a new encryption /decryption algorithm with variable 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. The attractive feature of this algorithm is its time complexity. It is simple and converts any plaintext into cipher text very fast. Another plus point of proposed algorithm is that it protects the cipher text from Brute-force attacks as the large variable key length used in the encryption process. This algorithm uses 8 different key to encrypt the plaintext into cipher text. Thus without knowing the exact sequence of exact key, it will be very hard to attain plaintext from cipher text.

A. Encryption:

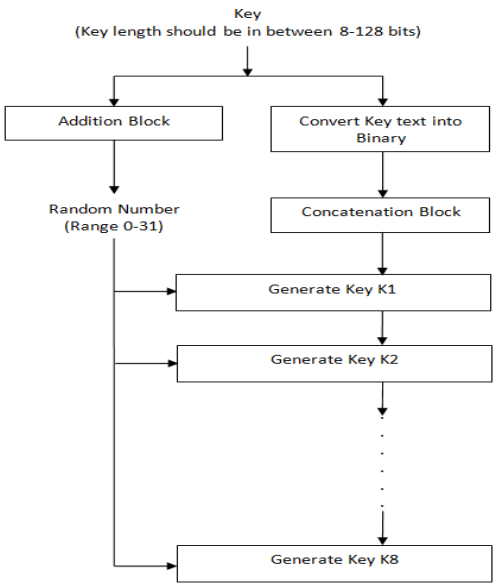

Figure 1 (a): Generation of eight different keys

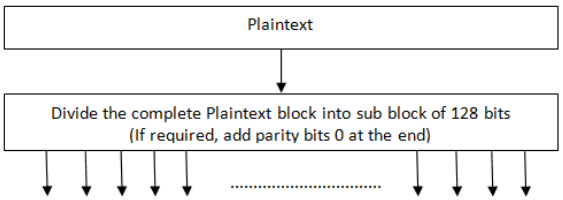

Figure 1 (b): Proposed Encryption Block-1

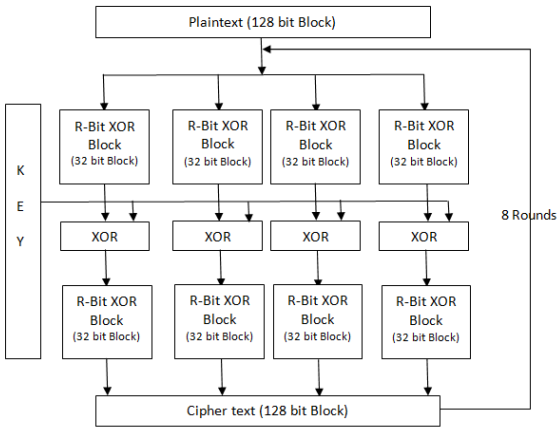

Figure 1 (c): Proposed Encryption Block-2

B. Proposed Encryption Algorithm:-

1) Initially, prepare 8 random 128 bit length key from arbitrary length key. Steps for preparing keys are as follows:

a) First generate a Random Number by passing the arbitrary length key into Addition block, which generate a random number having range 0 to 32. Addition Block is a block which performs addition of ASCII values of all the character of a key and generates a number. A 32 modulus of number is calculated and the resultant value is a Random Number generated from the key.

b) Now, Convert all the characters of a key into binary value which than passes to the concatenation block. The result of concatenation block is a 128 bit length key.

Concatenation Block is a block which takes arbitrary length binary key and performs XOR operation of all bits to its next bit.

The resultant value is than append with the original key.

c) Repeat step b) till the length of key become equal to 128 bits.

d) Now, for generating eight different keys from 128 bit key are as follow: First perform left circular shift on generated 128 bit key at random number times and the resultant value is the first key K1, Repeat the

same step on the key K1 and the resultant value is K2, similarly generate K3, K4, K5, K6, K7 and K8. Block diagram of generating 8 different keys is shown in Figure 1 (a).

2) Now, divide the input message (Plaintext) 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 end of message and make it equal to multiple of 128 bits block. As shown in Figure 1 (b). Now, select a block of 128 bits one by one from input file for encryption.
3) These blocks will divide into 4 sub block. Each sub block is of 32 bits.
4) All these 32 bits block is passed to the R-Bit XOR Block. This block perform xor operation on each bit with the bit at random number position to the right side of each bits
5) Now, Taking first key K1 and divide it into four sub block of 32 bits and perform xor operation with the result coming from step 4.
6) Result of step 5, is again pass to the R-Bit XOR Block.
7) Step 4 to 6 repeat 8 times with 8 different keys generated in step 1.
8) Result coming after the 8 round is the cipher text of first 128 bit plaintext block.
9) Repeat steps 3 to 7 for each 128 bit Plaintext Block.
10) Exit

C. Proposed Decryption Algorithm:-

1) Initially, generate the same 8 different keys from the arbitrary length key, similarly as done in encryption process.
2) Now, divide the cipher text into blocks, each of length 128 bits, as shown in Figure 2 (a). Now, select a block of 128 bits one by one from input file for decryption.
3) These blocks will divide into 4 sub block. Each sub block is of 32 bits.
4) All these 32 bits block is passed to the Reverse R-Bit XOR Block. This block performs xor operation on each bit with the bit at position random number to the left side of each bit.
5) Now, Taking key K8 and divide it into four sub block of 32 bits and perform xor operation with the result coming from step 4.
6) Result of step 5, is again pass to the Reverse R-Bit XOR Block.
7) Step 4 to 6 repeat 8 times with 8 different keys in reverse order (means from key K8 to K1) generated in step 1.
8) Result coming after the 8 round is the plain text of first 128 bit cipher block.
9) Repeat steps 3 to 7 for each 128 bit Cipher Text Block.
10) Exit

D. Decryption Block Diagram:-

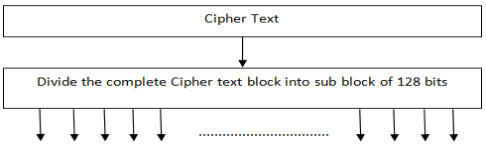

Figure 2 (a): Proposed Decryption Block-1

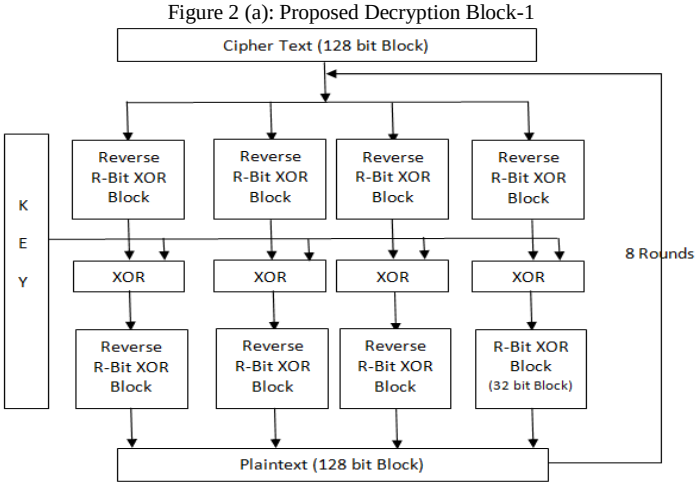

Figure 2 (b): Proposed Decryption Block-2

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 arbitrary length (in between 1 to 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.
- 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 arbitary bit long key will be: $2^{128}+2^{127}+...+2^{1} = 2^{1}(2^{128}-1)$ (therefore time complexity to break the key will be $O(2^{129})$).
- There are total 8 different keys are used which is dependent on random number having range 0-32, so each key can apply 8 *32 different manners. So the time complexity will increases to $O(2^{129+3+5}) = O(2^{137})$.

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.

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. Table 1 and 2 is showing encryption and decryption time of the proposed encryption/decryption algorithm on various files size with same key value with standard AES algorithm.

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

File Size in KB	**Algorithm**	
	Execution Time in Second	
	Proposed Algorithm	AES
5 KB	0.140	0.742
10 KB	0.608	1.772
20 KB	2.277	3.817

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
5 KB	0.146	0.793
10 KB	0.570	1.583
20 KB	2.278	3.214

A graphical representation for the table 1 and table 2 is shown in figure 3 and figure 4 with blue line for the proposed algorithm, red line for AES. 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.

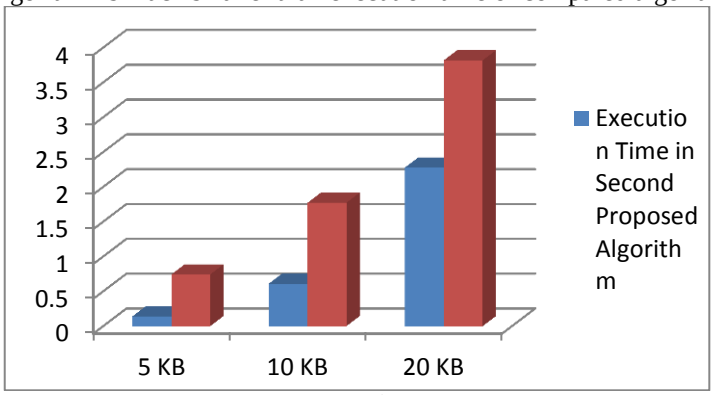

Figure 3: Encryption time of the proposed algorithm

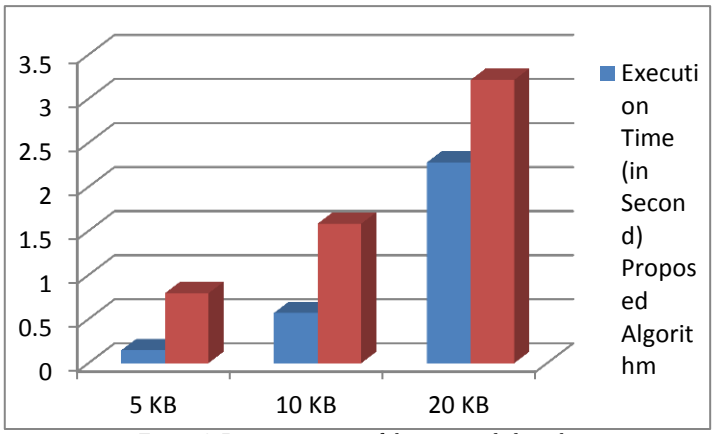

Figure 4: Decryption time of the proposed algorithm

A. Encryption Security

Encryption security considers the strength of encryption algorithm. Avalanche Effect is used to calculate the strength of any cryptographic algorithm. According to the avalanche effect, on changing the single bit in key 50% bits of cipher text must change. The algorithm close to avalanche effect is more secure against cryptanalysis. Table 3 shows the avalanche effect proposed algorithm.

On processing different 50 files, it is found that average avalanche effect comes out between 49 to 50%. This shows the strength of proposed encryption algorithm

TABLE 3 AVALANCHE EFFECT OF PROPOSED ALGORITHM

File Size in KB	**Proposed Algorithm**
	Avalanche Effect
Key-1 (ABCD)	49.4375%
Key-2 (ABCE)	

B. Memory Requirements:

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

TABLE 4: MEMORY COMPARISON BETWEEN PROPOSED ALGORITHM AND AES

Algorithms Name	Key Length in Bits	Plain Text Length in Bits (Input Block)	Cipher Text Length in Bits (Output Block)
Proposed Algorithm	1-128	128	128
AES	128	128	128

From the above tables it is concluded that the memory requirements of proposed algorithm are almost same to popular security algorithms AES.

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 arbitrary bits long (in between 1-128). 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.

- **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 arbitary bit long key will be: $2^{128} + 2^{127} + ... + 2^1 = 2^1(2^{128}-1)$ (therefore time complexity to break the key will be $O(2^{129})$).
- There are total 8 different keys are used which is dependent on random number having range 0-32, so each key can apply 8 *32 different manners. So the time complexity will increases to $O(2^{129+3+5}) = O(2^{137})$

IV. CONCLUSION

The proposed algorithm has been designed in a proficient approach but off- course not sacrificing the security issues. It has been successfully implemented on the text data. The proposed work also tried to benchmark the performance of proposed algorithm against some well-known Symmetric Key Algorithms like 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. Avalanche effect of proposed algorithm shows the internal strength of proposed encryption and decryption algorithm. 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] Neeraj Khanna, Joel James,Joyshree Nath, Sayantan Chakraborty, Amlan Chakrabarti and Asoke Nath : *"New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJJSAA symmetric key algorithm"* Proceedings of IEEE CSNT-2011 held at SMVDU(Jammu) 03-06 June 2011, Page 125-130.
[6] Dragos Trinca, *"Sequential and Parallel Cascaded Convolution Encryption with Local Propagation: Toward Future Directions in Cryptography"*, Proceedings of The third International Conference on information Technology-New Generations. (ITNG.06), 0-7695-2497- 4 / 2006, IEEE Computer Society.
[7] Data Encryption Standard : http://csrc.nist.gov/publications/fips/fips 46-3 /fips- 46-3.pdf
[8] Advanced Encryption Standard http://csrc.nist.gov/publications/fips /fips197/fips- 197.pdf
[9] Adam J. Elbirt, Christof Paar. *"An Instruction-Level Distributed Processor for Symmetric-Key Cryptography"* IEEE Transactions on Parallel and distributed Systems, Vol. 16, No. 5, May 2005.
[10] Cryptography and network Security Principles and Practices, Charles Fleeger
[11] William Stallings, *"Network Security Essentials* (Applications and Standards)", Pearson Education, 2004.