ISSN:0975-9646

A Novel Approach for Data Security Enhancement Using Multi Level Encryption Scheme

Sairam Natarajan [#1], Manikandan Ganesan [*2], Krishnan Ganesan [#3]
[#] School of Computing, SASTRA University-613402, Tamil nadu, India.

Abstract: **In this Information era, where all the transactions and files are digitized, the need for secure channel is eminent for transactions and confidential files. Information security is vital for many systems like core banking, defence systems, Satellite control systems, etc. wherein breach of secure data can lead to major consequences. Hence there is a demand for a stronger encryption which is very hard to crack. In this paper we proposed a multi level of multiple encryption schemes which enhances the security of the algorithm.**

Keywords: **Cryptography, Encryption, Random Hash Function, Security.**

I. INTRODUCTION

For ensuring the security, the plain text is converted to cipher text and the process is called encryption. Although this conversion idea is old, the way of encryption should not be vulnerable to attacks. Caesar's cipher method, poly alphabetic substitution method, bit-level encryptions like substitution box, permutation box, encoding, and rotation are some of the conventional encryption methods. These methods are easy to implement but can be cracked easily with the high end technologies. The objective of this project is to develop multi-level encrypter software that can be used to encrypt top-secret files including text, images and multimedia files in the secondary storage devices.

II. RELATED WORK

Most of the existing systems are vulnerable to attacks and it is breaked at some point of time by crypt analysing it. There are various cryptanalysis techniques available to break most of the encryption algorithms at one point of time. Each and every algorithm either it may be block cipher or stream cipher or any other cipher types can be easily attacked by performing various cryptanalysis techniques like linear cryptanalysis, n-gram analysis, meet in the middle attack, brute force attack, Man in the middle attack etc... It's pity to say that intruders can intrude any systems even it has a complex algorithmic design. Most of the famous algorithms of all ages are breaked easily by eavesdroppers at one stage and we are evidencing it in our day-to-day daily life. This happens because of its platform dependency and the emerging trend of open software solutions available all over the world. Despite some systems are developed to support cross platform, they do not use multi level encryption. This is because the algorithmic developers always believe in their own encryption formulas and firmly attached to the tradition of modifying or using or creating a single algorithm which is not secure after a period of time. It is quite obvious to digest the fact it is easy to cryptanalysis any algorithm within months as soon as they are adapted to practical use. Even though very few systems support multiple encryptions, they do not use randomized encryption hence can be cracked as soon as they came to know the algorithms used to build multi level encryption. Most of the existing systems support text encryption preferably than other media types. Since the intruders and eavesdroppers had shown their excelling skills towards breaking the encryption algorithms almost in all important and sensible areas like Banking, Military, Defence, Networks, a need for "practically strong and infeasible to get attacked" algorithm becomes vital. This paper suggests one such technique which never ever gives a clue of the encryption pattern adopted, no of encryption algorithms used, their order of execution.

III. PROPOSED SYSTEM

We proposed a system which is different and efficient from the existing systems as follows,

1. Our System is developed in such a way that it is platform independent. Where the existing systems are limited to platform dependent design.
2. It is developed through multiple encryption algorithms whereas the existing systems are always focussed as encryption at single level.
3. We use a Random function generator which generates a n-digit random number based upon the n-number of Encryption algorithms used. Thus generated n-digit number determines the order of selecting Encryption algorithms. Since the number determining the order is completely random it is infeasible to crack the order of execution.
4. Another significant feature of this random generator is, it is totally depends upon the key phrase that we pro-

vide and hence for various phrases it produce different order, which results the intruder in a more worse scene.

5. Moreover the number of encryption algorithm that we use, their order of execution will always remain a secret and hence it don't even leave a single chance for the eavesdroppers to make a guess on our system and hence the security offered is up to the best of ever provided.
6. This proposed system is developed in order to support not only text files but also images and media files. But still many of the existing systems are developed in order to suit basic text formats.

For simulating the system we adopted four famous cryptographic algorithms to implement multilevel security. The high feasibility with our system is it can suits any algorithm of any type. They are

1. Advanced Encryption System Algorithm,
2. Data Encryption Standard,
3. Rivest Shamir Adleman algorithm
4. Ceaser Cipher

The input file which is to be fed in to our MLIS system is chosen at first. Then, a key phrase is entered which is none other than a secret pass phrase or a password for data authentication. Then there comes a randomizer function which calculates a random number of length n depending up to the key phrase that we provided, where n is the no of encryption algorithms that are to be get used. Our system produces maximum number of combinations that can be made with the number of algorithms chosen. For Example for a system which uses 3 encryption algorithms, then it generates 3! combinations namely 123,231,321,132,213,312. So in general for n bit random number it produces n! combinations. Among the possible combination random number thus generated of n-bit determines the order of execution of those encryption algorithms. For example if the resulted random number is 4-3-1-2 for the multilevel encryption system that we used for simulation (which is specified above) then the order of execution will be ceaser cipher at first and then RSA Algorithm which is followed by AES and DES algorithms. So the cipher text generated from the ceaser cipher is supplied as plaintext for RSA algorithm. The resultant cipher text of RSA algorithm is supplied as plaintext to AES and the cipher text of AES as plaintext for DES. After executing the four algorithms in the order generated by our system randomly we can send the resulted multi cipher text to the receiver via a communication link. At the receiver's end, the reverse of encryption order takes place. As by our case it is 2-1-3-4.so the cipher text is decrypted by DES algorithm which still possesses some scrambled text is supplied is plaintext for AES and it is continued for RSA and Ceaser Cipher where at last we get resulted in the original message that the sender wishes to communicate. The following architecture shows our proposed MLIS System flow design from top to down approach.

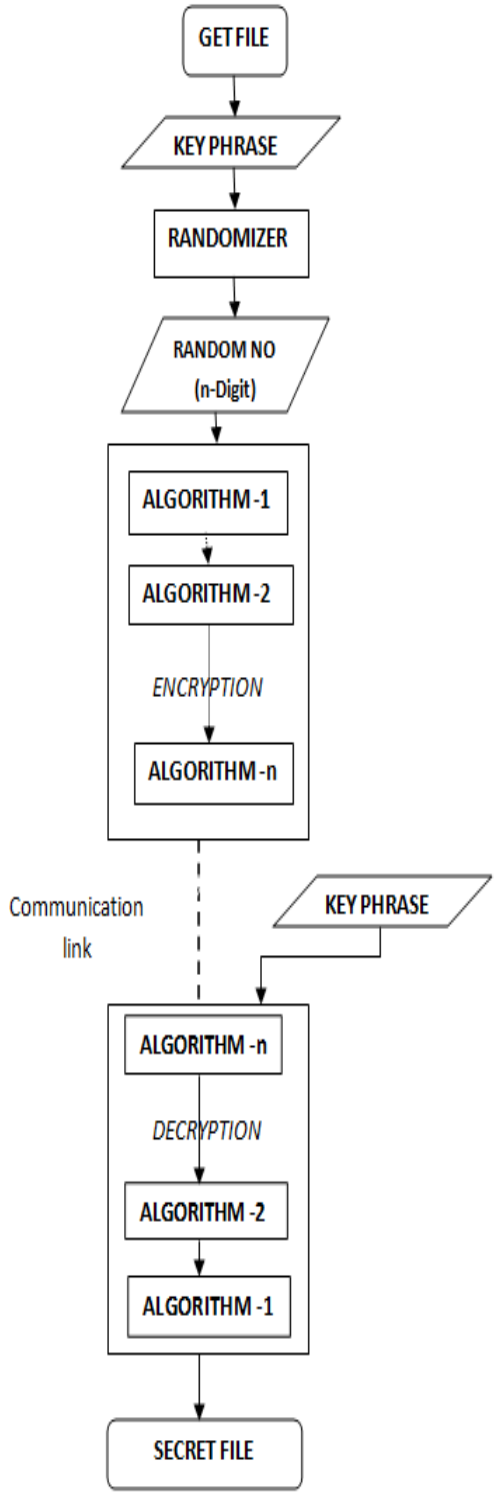

Fig., 1. Proposed Architecture

A. AES Algorithm

This algorithm is flexible in supporting any combination of data and key size of 128, 192, and 256 bits. However, AES merely allows a 128 bit data length that can be divided into four basic operation blocks. These blocks operate on array of bytes and organized as a 4×4 matrix that is called the state. For full encryption, the data is passed through Nr rounds (Nr = 10, 12, 14) [4, 6]. These rounds are governed by the following transformations:

(i) Byte Substitution: This is a non linear byte Substitution, using a substation table (s-box), which is constructed by multiplicative inverse and affine transformation.

(ii) Shifting the rows: This is a simple byte transposition, the bytes in the last three rows of the state are cyclically shifted; the offset of the left shift varies from one to three bytes.

(iii) Mixing of columns: Is equivalent to a matrix multiplication of columns of the states. Each column vector is multiplied by a fixed matrix. It should be noted that the bytes are treated as polynomials rather than numbers.

(iv) Adding round key: Is a simple XOR between the working state and the round key. This transformation is its own inverse. The following diagram shows pictorial view of overall algorithm.

B. DES Algroithm:

DES is a block cipher which takes a fixed-length string of plaintext bits and transforms it into cipher text bit string of the same length. The key length of DES is 64 bits. DES Algorithm uses Fiestal structure which performs the following operations.

1. *Expansion:* By duplicating some of the bits, 32-bit block is expanded into 48 bit block by using the expansion permutation.
2. *Key mixing:* Expanded block is mixed up with a substitution key by using an XOR operation. Sixteen 48-bit sub keys are derived from the main key as one key for each round.
3. *Substitution:* After Key Mixing is over, the block is further divided into 6-bit pieces of eight blocks. By following Non liner transformation, each of the S-boxes replaces its six input bits with four output bits. S-boxes plays a important role in determining the security of the algorithm and without them the algorithm becomes linear and easily breakable.
4. *Permutation*: Thus after substitution 32 outputs from the S-boxes is rearranged by using the concept of permutation.

C. RSA Algorithm:

RSA is an asymmetric or public key cryptographic algorithm i.e., it uses two different keys. It uses two keys namely public key and private key. The public key is a key which can be known to anyone and it is used to encrypt messages. But the other key called private key is the secret key which is used to decrypt the enciphered message. The key generations for the RSA algorithm are done by following steps of procedure:

1. Pick two relatively prime numbers x and y
2. Calculate the value of n which is the modulus for the public and private keys by using the formula n=x*y
3. Determine the quotient: Q (n)= (x-1) (y-1).
4. Choose an integer i such that 1< i <<q(n),and integer i is co-prime to Q(n) that is 1is the only greatest common factor which divides i and Q(n) .
5. I and D are the public key and private key exponent respectively.
6. Calculate D which satisfies the congruence relation

$$DI=1+Q(n)*K \text{ where K is a integer.} \quad (1)$$

RSA Algorithm involves two steps namely Encryption and Decryption respectively.

Encryption: Receiver provides its public key N and I to Sender and keeps its private key secret. Sender wants to transmit the secret message to receiver. At first Sender turns the secret message into a number which is smaller than n by adopting any padding scheme or any reversible protocol .hence the ciphertext can be calculated as

$$C=M^{I} \bmod N. \quad (2)$$

Decryption: Decryption is an inverse to the RSA's Encryption process. Receiver can recover the secret message M from cipher text C by using its private key d in the following formula

$$M= C^{D} \bmod N. \quad (3)$$

D. Ceaser Cipher:

It is the monoalphabetic cipher devised by Julius Ceaser the Great for his wartime communications.It consists of two parts encrypting the messsage and decrypting the encrypted message.

Ceaser cipher Encryption will pick a letter and replaces it with an another letter which is k th to the actual alphabet.i.e., if k is 3 then a is replaced by d,b is replaced by e and so on.So,the encryption formulae foe Ceaser cipher can be depicted as,

$$C = P + K \quad (1)$$

Where C is the cipher text, P is the plain text, and k is the key, an integer.

Decryption is the reverse of encryption i.e., an alphabet is replaced by the kth alphabet present before it.such that,

$$P = C - K \quad (2)$$

where P is the plain text,C is the cipher text and k is the key, an integer.

IV. SIMULATION*:*

The following set of figures shows the encryption and decryption of text files. We simulated tested and verified it using Java platform

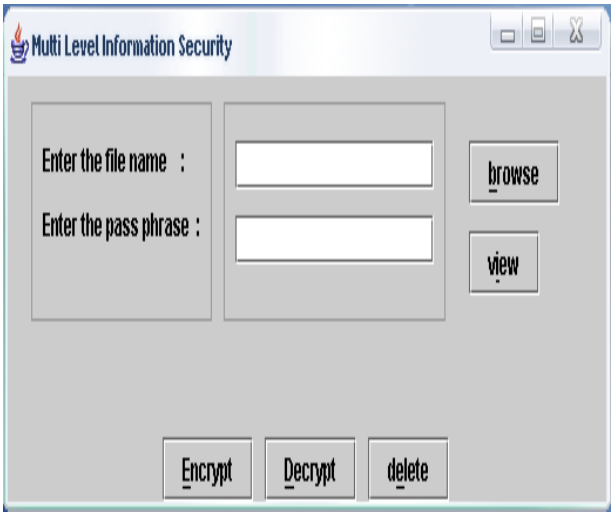

Fig., 2. Home screen

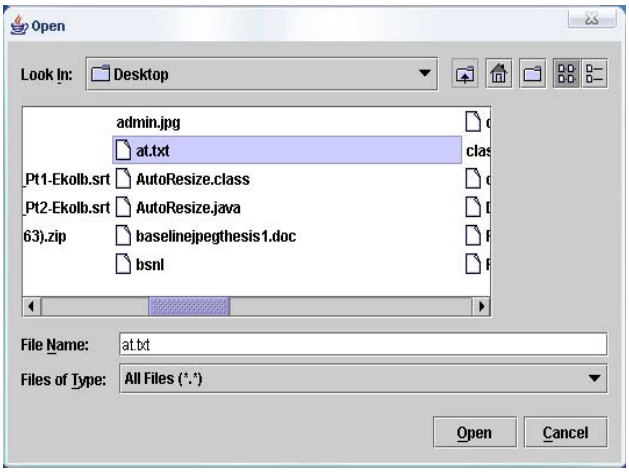

Fig., 3. Browse window

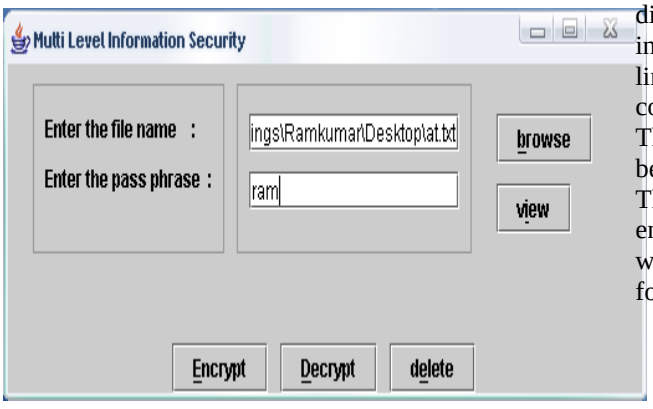

Fig., 4. Encrypting text file

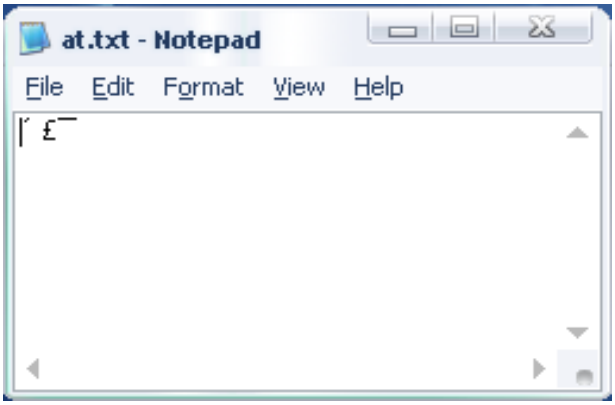

Fig., 5. Cipher Text

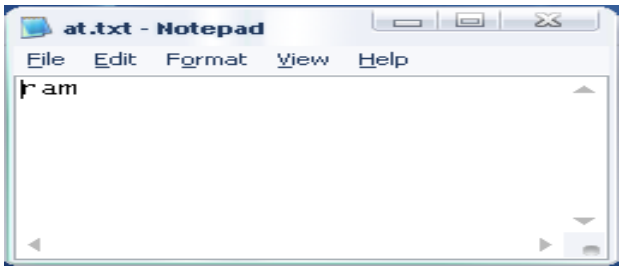

Fig., 6. Text file after decryption

V. Future Enhancements

The system can be easily modified to accept any encryption algorithm which is framed in future. Just by adding or removing another module in the main function and also by increasing or decreasing the hash limit, any number of algorithms can be included or reduced. Though the system is designed for storage level but the modules can be used in web services also. By adding a new button with a server and client sockets, the system can also be improved to work as secure LAN File messenger. Moreover, we currently concentrate on our next work which adopts Parallelism where we can run various Encryption Algorithms in parallel environment which enhances the performance and time taken for Encryption/Decryption.

VI. Conclusion

So far we have discussions on various algorithms in cryptography. Each algorithm having its own advantages and disadvantages, this system proposed a good strategy of making most out of their advantage while trying to eliminate the limitations. The developed system ignoring the front end could be used in any network services for network security. The system also supports 64 bit operating systems which will be of future concern of all Operating System manufacturers. The concept of multi level encryption along with randomizer enhances the security of files. The system also proposed a way of encrypting media files. Thus the system is justified for its use in securing files.

REFERENCES

1. W. Stallings, *Cryptography and Network Security: Principles and Practices*, 2nd ed., Prentice Hall, 1999.

2. Walter Tuchman , "*A brief history of the data encryption standard*",Internet besieged:countering cyberspace scofflaws. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, pp. 275–280,1997.

3. William E. Burr, "*Data Encryption Standard", in NIST's anthology*",A Century of Excellence in Measurements, Standards, and Technology: A Chronicle of Selected NBS/NIST Publications,2000.

4. Joan Daemen, Vincent Rijmen, "*The Design of Rijndael: AES - The Advanced Encryption Standard*." ,Springer, 2002.

5. Nicolas Courtois, Josef Pieprzyk, "*Cryptanalysis of Block Ciphers with Overdefined Systems of Equations*", pp267–287, ASIACRYPT 2002.

6. Christof Paar, Jan Pelzl, "The Advanced Encryption Standard", Chapter 4 of "*Understanding Cryptography, A Textbook for Students and Practitioners*", Springer, 2009.

7. Rivest, R.; A. Shamir; L. Adleman ,"*A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*", Communications of the ACM 21 (2): 120–126,1978.

8. Sinkov, Abraham; Paul L. Irwin," *Elementary Cryptanalysis: A Mathematical Approach*", Mathematical Association of America. pp. 13–15,1966.