Journal of Theoretical and Applied Information Technology
15 May 2012. Vol. 39 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A PARALLEL APPROACH FOR IMPROVING DATA SECURITY

[1]KARTHIKEYAN.S, [2]SAIRAM.N, [3]MANIKANDAN.G, [4]SIVAGURU.J

[1]Student,School of Computing,SASTRA University,Thanjavur-613401,TamilNadu, India
[2] Professor,School of Computing,SASTRA University,Thanjavur-613401,TamilNadu, India
[3]Assistant Professor,School of Computing,SASTRA University,Thanjavur-613401,TamilNadu, India.
[4]Student,School of Computing,SASTRA University,Thanjavur-613401,TamilNadu, India

E-mail: [1]karthiswamy@gmail.com, [2]sairam@cse.sastra.edu, [3]manikandan@it.sastra.edu, [4]jsivaguru@gmail.com

ABSTRACT

In today's world internet is being used by almost everyone. Numerous file exchanges take place online including many official documents. These files require some sort of security mechanisms while being transmitted over the internet. Cryptography is one such means by which a person can encrypt the data and send it through the internet. This way the data is safe and unreadable for the intruders. In this paper we have proposed a system which combines the advantages of parallel processing and cryptographic algorithms. The use of parallel processing enhances the speed of system when compared to the traditional crypto systems. In our approach we have divided a file into two slices and have applied a single algorithm with different key for each slice and the processing of the algorithm is done in a parallel environment. From our experiments we found out that the execution time of a cryptographic algorithm is considerably reduced in a parallel environment when compared to the generic sequential methods.

Keywords : *Cryptography, Parallel Processing, File Security, Symmetric-Key.*

1. INTRODUCTION

Using cryptography for the transactions over a network is an age old yet powerful technique for data security. Many algorithms were developed and employed. But the advantages and disadvantages of each algorithm vary. Existing systems use a single algorithm in sequential manner for encryption. This makes the system vulnerable to attacks at a point of time by the cryptanalysts using various techniques available. Some of the cryptanalysis techniques are brute force attack, linear and non linear cryptanalysis, n-gram analysis, meet in the middle attack, man in the middle attack, etc., Enhancements to the existing algorithms help to overcome these problems. This
paper proposes one such technique that uses a single algorithm for encryption of a file in parallel environment.

The cryptography techniques use encryption that scrambles the plain text using some algorithms that produces a text that cannot be interpreted by the intruders. This scrambled text is called as cipher text. At the receiving end the cipher text is again converted to the plain text by using the same algorithm. The actual plain text can be obtained by those who possess a key for decryption. Others, who does not have a key can only get a different scrambled message, actual text cannot be retrieved.

There are two types of cryptographic schemes available [1]. They are,

1. Symmetric Key Cryptography – uses a common key for both encryption and decryption of the message.
2. Asymmetric Key Cryptography – uses two different keys called public key and private key; one is used for encryption and another for decryption.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

One of the best ways to strengthen a cryptographic scheme is to increase the strength of its key. Manikandan etal., (2011) suggest a method to improve the strength of the key in case of RC4 algorithm[2].

The rest of the paper is organized as follows. Section II briefly describes the AES (Advance Encryption standard) algorithm and parallel environment. Section III provides insights about the proposed system. Section IV summarizes the results of our experiment. Section V gives the concluding statement of our work.

2. LITERATURE REVIEW

In the year 2011, Manikandan et al.,has proposed an interesting approach to combine the two basic forms of ciphers, namely block cipher and stream cipher .[7]

In the year 2011, Manikandan et al., suggested a method in which one type of cipher was used with the plain text and the other with the key.[8]

Manikandan et al., 2011 proposed an approach to improve security by integrating cryptography and steganography. [9]

In [11] the combination of steganography and cryptography is further strengthened by including a compression technique before using the encryption process.

A new approach for improving data security by executing the blowfish algorithm

In an iterative manner was proposed [10].

In [12] A Pseudorandomised Key Generation approach for improving the data security was proposed.

The existing systems use a single algorithm in the sequential manner to perform encryption of data. A single key is used for the encryption of the whole data or multiple keys are used for repeated encryptions in sequence [3]. The sequence is maintained in order to preserve the symmetric nature of the encryption technique. It is also possible for an intruder to obtain a key by brute force attack, which is equivalent to the sequence of the keys used. The cipher text produced can be broken by cryptanalysis if a set of plain text and cipher text combinations are known or if the key and the algorithm used are guessed. The sequential processing increases the execution time.

2.1 AES(Advanced Encryption Standard) Algorithm

AES provides flexibility, simplicity and hardware and software sustainability. This algorithm can be used in a variety of platforms and applications. This can also be used as a stream cipher, message authentication code, hash function, pseudorandom number generator [4].

AES uses different key sizes of 128, 192 or 256 bits but restricts the block length to 128 bits. This provides greater security when compared with other algorithms of same key size and block length.

The algorithm works as follows [5]:

1. The key provided is expanded into an array of forty-four 32 bits words. 4 words (128 bits) is used as a round key for each round.
2. Four different operations are performed in each round, namely
   a. Substitute bytes : Performs byte-by-byte substitution using an S-box
   b. Shift rows: A simple permutation technique to change the values by shifting the rows.
   c. Mix columns: Uses arithmetic operation and performs substitution.
   d. Add round Key: Bitwise XOR of current block with the portion of the expanded key.
3. For encryption and decryption the process begins with an Add round key stage and followed by 9 rounds, each round consists of four stages mentioned above. The last round contains only 3 stages excluding add round key stage.

2.2 Parallel Processing using Message Passing Interface

The process of solving a problem at a greater computational speed by programming multiple computers or a computer with multiple internal processors is called parallel processing[5]. The need for parallel processing is increasing with the necessity for high-speed processing. Message Passing Interface (MPI) is the standard specification for message passing library. Programs developed using MPI can be reused in newer, faster parallel computers.

Using parallel processing reduces time complexity and cost as parallel computers can be

Journal of Theoretical and Applied Information Technology
15 May 2012. Vol. 39 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

built from cheap components. Many large problems which are infeasible to solve using a single processor computer can be solved using parallel processing. Parallel computer use non local resources i.e, resources of a wide area network or even internet when the local resource is scarce. [6]

3. PROPOSED SYSTEM

Our aim is to design a system that incorporates the advantages of parallel processing in cryptographic algorithms. Apart from this it also opens a new gate in the way a file is encrypted. Since the encryption process is done in parallel it is possible to use different keys for different slices which will further enhance the security of the data.

In our proposed system, we divide a file into two slices and apply the same cryptographic algorithm on both the slices using different keys in different processor. The key for the second slice is obtained from the key that has been used to encrypt the first slice using the transposition technique. For experimental purpose we have reversed the original key given by the user to generate the new key. Both the encryption processes take place in parallel hence decreasing the processing time.

Following are the steps involved in the proposed system:

1. The plain text file and a key are provided as inputs to the system.
2. The plain text file is divided into two slices based on the size of the file to enable parallel processing.
3. A new key is generated using the key provided as input by using transposition technique. The generated key is not even known to the user.
4. An Encryption algorithm used for encrypting each part of the file with different keys, a provided key and a generated key.
5. After encryption all the cipher text are combined to generate a single cipher text file.
6. For decryption the file is again divided into two and each slice is decrypted with their respective keys, which are communicated to the receiver through a secured means.

The flow diagram for the proposed model is given below:

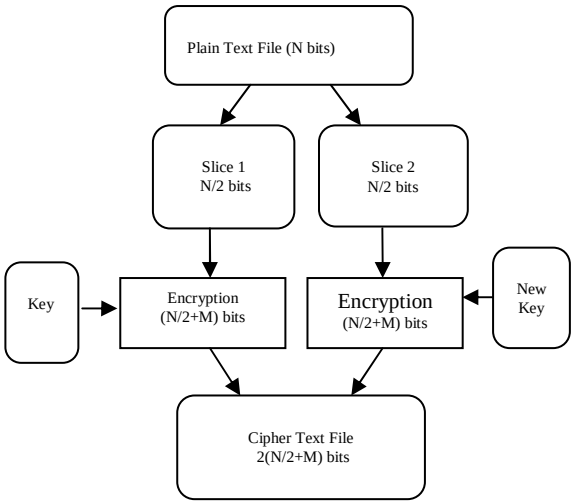

Figure 1: Encryption

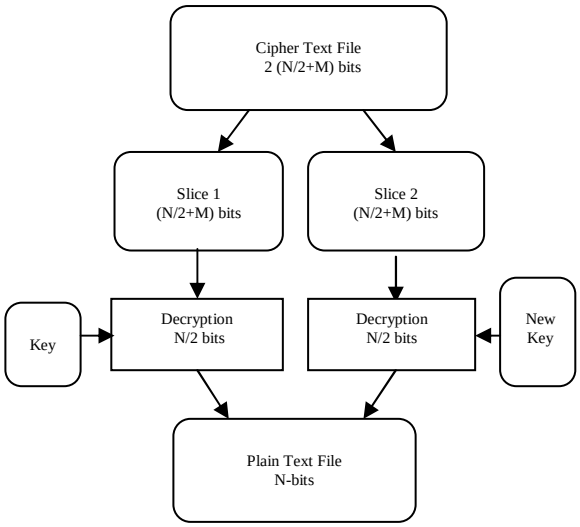

Figure 2: Decryption

4. SIMULATION AND RESULTS

For the purpose of simulation of this system, we divide the file into two slices and both the slices have been encrypted using the AES algorithm. A new key is generated using the key provided. The actual key is used for encrypting the first slice and the new key is used for second slice. For decrypting the cipher text file, the file is again divided and the same mechanism is used to generate a new key from the provided key. The decrypted files are

Journal of Theoretical and Applied Information Technology
15 May 2012. Vol. 39 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

combined to a single text file. The screenshots are given below:

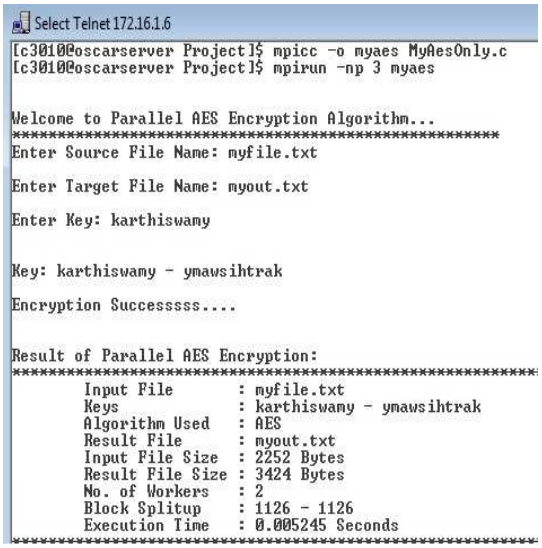

Figure 3: Parallel Encryption

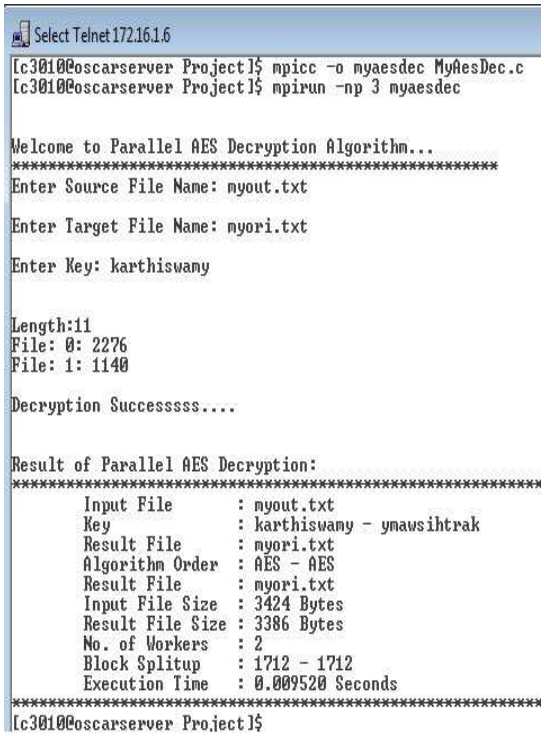

Figure 4: Parallel Decryption

Figures 3 and 4 show the parallel encryption and decryption of the input text file in our proposed system. The input key and the new key generated from the input key are also shown.

The input text file (myfile.txt) before encryption is shown in the figure 5. The cipher text file (myout.txt) after encrypting is shown in figure 6. Figure 7 shows the text file (myori.txt), which is decrypted from the cipher text file.

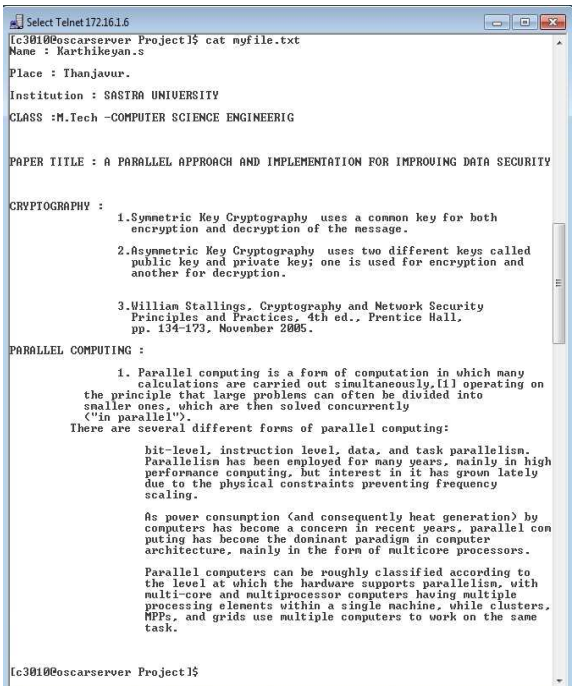

Figure 5: Input(Plain Text) File

Figure 6: Encrypted File (Cipher Text)

Journal of Theoretical and Applied Information Technology
15 May 2012. Vol. 39 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

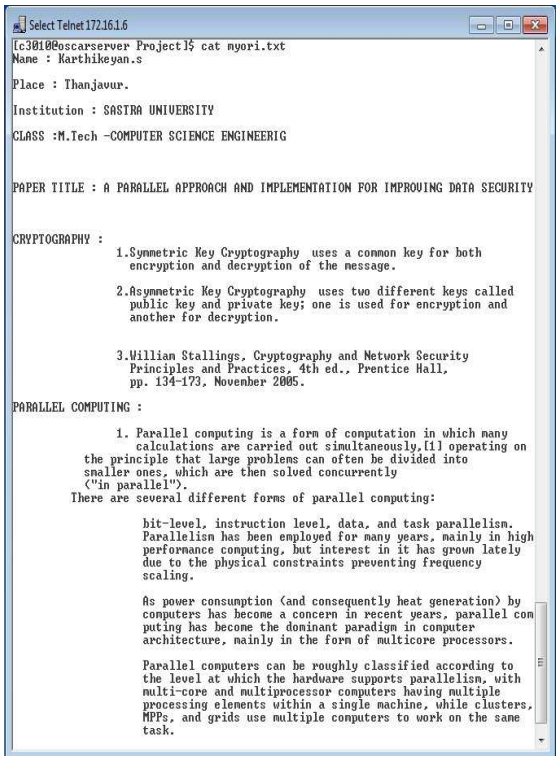

Figure 7: Original File

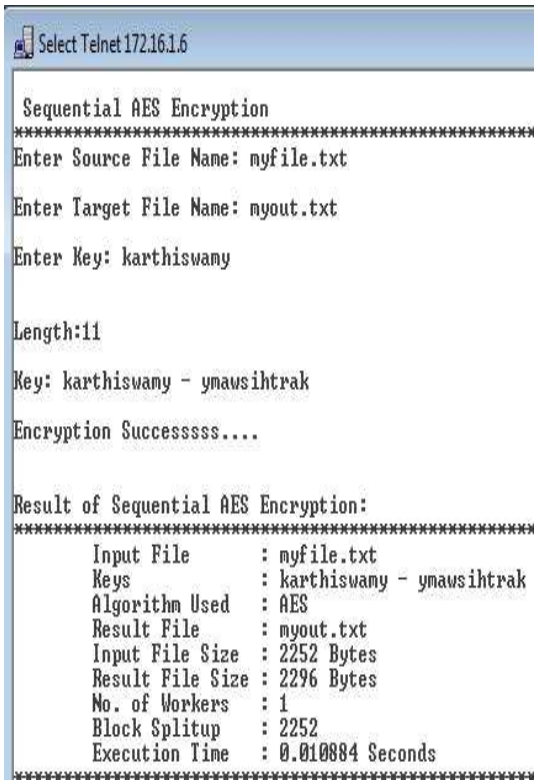

Figure 8: Sequential Encryption

Figure 9: Sequential Decryption

Figures 8 and 9 show the sequential encryption and decryption of the same file and also the execution time for both the processes.

Different sized files are used for encryption and decryption in both sequential and parallel environment and the time taken for execution are calculated and the observations are shown in the Table1.

Table 1: Comparision of sequential and parallel Execution

S.No	File Size (Bytes)	Process	Execution Time (sec)
1	575	Sequential	0.008346
		Parallel	0.005456
2	2252	Sequential	0.029159
		Parallel	0.014765
3	7648	Sequential	0.09432
		Parallel	0.047585
4	10836	Sequential	0.130561
		Parallel	0.066875
5	16709	Sequential	0.198252
		Parallel	0.102362

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

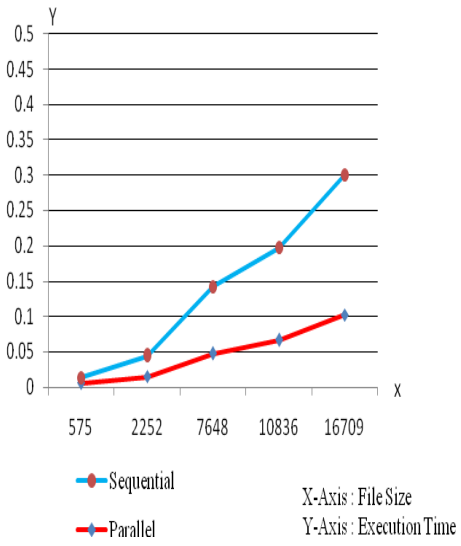

Figure 10: Comparison of Parallel and sequential Encryption

The results of the simulation shows that the file contains cipher text of the plain text but encrypted with two different keys. This makes it infeasible for the cryptanalysts to guess the keys, even a set of cipher text and plain text combinations with the algorithm implemented are known as the key generating mechanism is only known to the sender and the receiver. The new key cannot be guessed even if the key provide is trapped. The cipher text thus produced is very complex as it uses different key for different part of the file.

5. SECURITY AND CORRECTNESS ANALYSIS

Considering N bits are given as input to the encryption algorithm it gives N+M bits as output. During the decryption process N+M bits are given as input and N bits are obtained as output. The correctness of the proposed system is based on the above statement. This is shown in the Figure 1 and Figure 2. with the number of bits specified for each and every step.

In our proposed system no changes is made to the actual functionalities of AES algorithm (i.e., we did not add or remove any operations, rather we execute AES in parallel environment) so performing cryptanalysis is not necessary. Since this approach uses two different keys for the cryptographic process it enhances the overall security of the system and hence it is difficult for an intruder to gain access to the plain text.

6. CONCLUSION AND FUTURE WORK

From the simulation results it is observed that execution time by using parallel processing is very low when compared to that of sequential execution and the cipher text quality is also increased, providing greater security to the information. File can be divided into more than two splices based upon the number of processors available. Further, each processor can implement a different algorithm instead of using the same algorithm for encrypting all the parts.

REFERENCES:

[1] William Stallings, "Cryptography and Network Security Principles and Practices", 4th ed., Prentice Hall, November 2005, pp. 134-173.

[2] G. Manikandan, R. Manikandan, G. Sundarganesh, "A New approach for generating Strong Key in RivestCipher4 Algorithm", Journal of Theoretical and Applied Information Technology, 2011, pp.113-119.

[3] N.Sairam, G.Manikandan, G. Krishnan, "A Novel Approach for Data Security Enhancement using Multi Level Encryption Scheme", International Journal of Computer Science and Information Technologies, Vol.2(1),2011, pp.469-473.

[4] Bruce Schneier, "Applied Cryptography, Protocols, Algorithms and Source Code in C", 2nd ed., John Wiley& Sons, 1996.

[5] Deguang Le, Jinyi Chang, Xingdou Gou, Ankang Zhang, Conglan Lu, "Parallel AES Algorithm for Fast Data Encryption on GPU", 2nd International Conference on Computer Engineering and Technology, Vol.6, ,2010, pp.v6-1-v6-6.

[6] Michael J.Quinn, "parallel Programming in C with MPI and Open MP", 1st ed, McGraw Hill Higher Education.,2008.

[7] G. Manikandan, G. Krishnan, N.Sairam, A Unified Block and Stream Cipher Based File Encryption, Journal of Global Research in Computer Science, Vol. 2 (7), 2011, pp 53-57.

[8] G.Manikandan, R.Manikandan,P. Rajendiran, G.Krishnan,G.SundarGanesh, An Integrated Block and Stream Cipher Approach for Key Enhancement , Journal of Theoretical and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

applied information Technology, 2011, Vol 28 (2), 83-87.

[9] G. Manikandan, M.Kamarasan,P.Rajendiran, R.Manikandan, A Hybrid Approach for Security Enhancement by modified Crypto-Stegno scheme, European Journal of Scientific Research, vol.60(2) , 2011, 224-230.

[10] G.Manikandan, N.Sairam, M.Kamarasan, A New Approach For Improving Data Security Using Iterative Blowfish Algorithm in Journal of Applied Sciences, Engineering and Technology,Vol 4(6) , 2012,603-607.

[11] G.Manikandan, N.Sairam, M.Kamarasan, A Hybrid Approach For Security Enhancement by Compressed Crypto-Stegno Scheme in Research Journal of Applied Sciences, Engineering and Technology,Vol 4(6) ,2012, 608-614.

[12] B.Karthikeyan, V.Vaithiyanathan, B. Thamotharan, M.Gomathymeenakshi and S.Sruthi, LSB Replacement Stegnography in an image Using Psudorandomised Key Generation. Research Journal of Applied Sciences, Engineering and Technology, 4(5), 2012, 491-494.