APPLICATION OF CLASSICAL ENCRYPTION TECHNIQUES FOR SECURING DATA- A THREADED APPROACH

Raghu M E[1] and Ravishankar K C[2]

[1]Department of CSE, Government Engineering College, Hassan, Karnataka, India,
[2]Department of CSE, Government Engineering College, Hassan, Karnataka, India,

ABSTRACT

The process of protecting information by transforming (encrypting) it into an unreadable format is called cryptography. Only those who possess secret key can decipher (decrypt) the message into plain text. Encrypted messages can sometimes be broken by cryptanalysis, also called code breaking, so there is a need for strong and fast cryptographic methods for securing the data from attackers. Although modern cryptography techniques are virtually unbreakable, sometimes they also tend to attack.

As the Internet, big data, cloud data storage and other forms of electronic communication become more prevalent, electronic security is becoming increasingly important. Cryptography is used to protect e-mail messages, credit card information, corporate data, cloud data and big data so on... So there is a need for best and fast cryptographic methods for protecting the data. In this paper a method is proposed to protect the data in faster way by using classical cryptography. The encryption and decryption are done in parallel using threads with the help of underlying hardware. The time taken by sequential and parallel method is analysed.

KEYWORDS

Cloud, Data, Cryptography, Parallel cryptography, Threads.

1. INTRODUCTION

Cryptography (or cryptology) means “hidden secret” is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation [5, 6]. Modern cryptography intersects the disciplines of mathematics, Computer Science and Electrical Engineering. Cryptography prior to the modern age was effectively synonymous with encryption, the conversion of information from a readable state to apparent nonsense. The originator of an encrypted message shared the decoding technique needed to recover the original information only with intended recipients, thereby precluding unwanted persons to do the same.

Modern cryptography is heavily based on mathematical theory and Computer Science practices. Cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary.

DOI: 10.5121/ijci.2015.4212

It is theoretically possible to break such a system but it is infeasible to do so by any known practical means. These schemes are therefore termed computationally secure, theoretical advances. Improvements in integer factorization algorithms and faster computing technology require these solutions to be continually adapted [6]. There exist theoretically secure schemes that provably cannot be broken even with unlimited computing power. An example is the one-time pad. But these schemes are more difficult to implement than the best theoretically breakable but computationally secure mechanisms.

Cryptography is the discipline, art and science of ensuring that data is secure from possible attacks, whether these attacks be eavesdropping, impersonation or corruption. Cryptography provides security through a number of mathematical transformations that can be proven to be mathematically secure provided some optimum conditions. However there is a need to cognizant that cryptography on its own is insufficient to ensure a high level of security within an organization, that is to say that cryptography is not the silver bullet to solve all information security issues and should be used in conjunction with good security practices. Cryptography, like the Information Security field itself, is an incredibly broad field involving many existing disciplines such as abstract algebra to provide mathematical proofs for the guaranteed correctness of an algorithm, statistics for analysis of cryptographic.

Securing the information is done in sequential way using existing algorithms. It requires more time to encrypt and decrypt the large data, so there is a need for parallel approach to do the same in faster and secure manner. This paper proposes a method to secure large data using parallel approach.

2. MOTIVATION

Why fast processing/parallel processing is so important for cryptography? The reason is that many applications require fast cryptographic software and that even small speedups justify high effort. Consider for example Internet content providers running large server farms. Encrypting all transmitted data requires many computers that do nothing but perform cryptographic operations. Even a speedup of only 10% of the software saves 10% of hardware and power cost. Also private users benefit from fast cryptographic software. So there is a need for single computer to do all work in parallel to provide cost effectiveness in hardware and fast processing of required work. Here it is performing conversion of plain text to cipher text and vice versa.

The term parallel cryptography used here refers to the design and implementation of secure and fast cryptographic functions for the present good hardware computers to perform cryptography in fast and cost effective manner. The use of parallel processing enhances the speed of system when compared to the traditional crypto systems. As we knew that hardware evolution is faster in comparison with the present software evolution. It is possible to develop algorithm that make use of available hardware facilities to perform the crypto operation more faster and cost effective manner.

3. LITERATURE REVIEW

The literature review introduces and defines concepts relating to cryptography, issues relating to cryptography need for parallel approach in the cryptography and the development of software frameworks.

Karthikeyan S, Sairamn, Manikandan G, Sivaguru J [1] 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 this approach

they 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 the experiments it is found out that the execution time of a cryptographic algorithm is considerably reduced in a parallel environment when compared to the generic sequential methods.

Osama Khalifa [2] addresses the problem of enhancing the performance of strong cryptographic algorithms, which are widely used and executed by almost all Internet users. The author used the parallel computing as a means to improve the performance. Especially nowadays multi-core computers are commonly available. Since the security level provided by most cryptographic algorithms depends on the difficulty of solving some computational problems, the developments in computer systems manufacturing will threaten people's security. Thus, it is very important to cope with this development and increase the security level by using stronger cryptographic algorithms with longer keys which in return will take longer to encrypt and decrypt data but also a much longer time to hack the cipher text. The resulted parallel algorithm(s) will be assessed by measuring the scalability and speedup features, moreover, it will be able to adapt to the increasing number of cores in a dynamic way.

Vinodh Gopal , Jim Guilford, Wajdi Feghali [3] Cryptographic algorithms, such as secure encryption, occur in networking, storage and other applications. Since the amount of data being processed is large and increasing at a rapid rate, there is an ever-increasing need for very high performance implementations of these algorithms. The introduction of the 2nd Generation Intel® Core™ processor family brings an additional substantial boost in performance on cryptographic algorithms.

H. Naveen, M. Ramesh [4] By exploring different granularities of data-level and task-level parallelism, they mapped implementations of an Advanced Encryption Standard (AES) cipher with both on-line and off-line key expansion on a fine-grained many core system. The smallest design utilizes only six cores for off-line key expansion and eight cores for on-line key expansion, while the largest requires 107 and 137 cores, respectively. In comparison with published AES cipher implementations on general purpose processors, the design has 3.5-15.6 times higher throughput per unit of chip area and 8.2-18.1 times higher energy efficiency.

According to Salem Sherif Elfard[7] Cryptography is the only practical means to provide security services and it is becoming a powerful tool in many applications for information security. The intension of author is to give a parallel algorithm based on the linear Fibonacci forms method and applying it on a modular reduction on addition machines and also a modular exponentiation based on linear Fibonacci forms.

The method proposed by K C Ravishankar and M G Venkateshmurthy [8] on region permutation deals with scrambling the regions of an image based on symmetric key. This is done in order to introduce the disorderness in the visibility of an image. By doing permutation the regions are trans-positioned to the new locations. The algorithm introduces the randomness and simple to retrace back if key is known.

The literature review summarizes the need for fast and efficient crypto algorithm for present applications. Due to present need it is necessary to have fast algorithm which performs the crypto process in time efficient and secure manner, for this the best method is performing the crypto operation in parallel using available hardware technology.

4. DESIGN

The encryption is a process of making plain text to cipher text and decryption is converting cipher

text to plain text. Doing the same in sequential way is always time consuming. This paper proposed a method where the encryption and decryption is done in parallel way using threads.
In the proposed system the subdivision method is used. Here the data is divided into number of small units called chunks using the subdivision algorithm and each chunk is of same size. Now each chunk is moved to the threads, each thread takes chunk of data as input and encrypt the data and gives the output. Output of each thread is collected to form the output file. Figure 1 shows the whole process in the pictorial way.

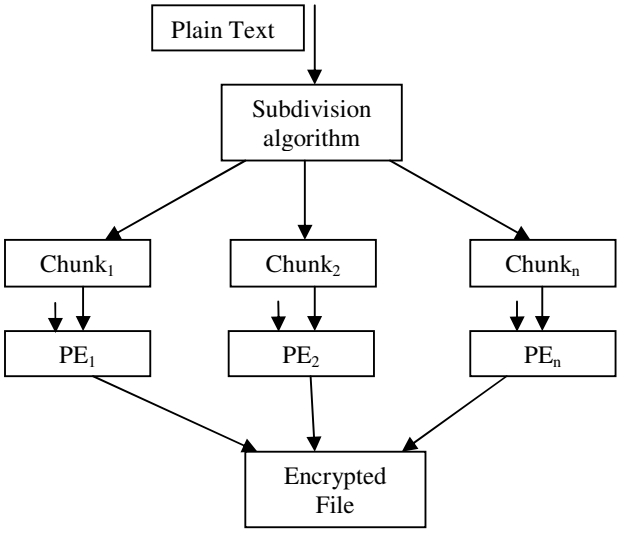

Figure 1: Parallel Encryption using threads

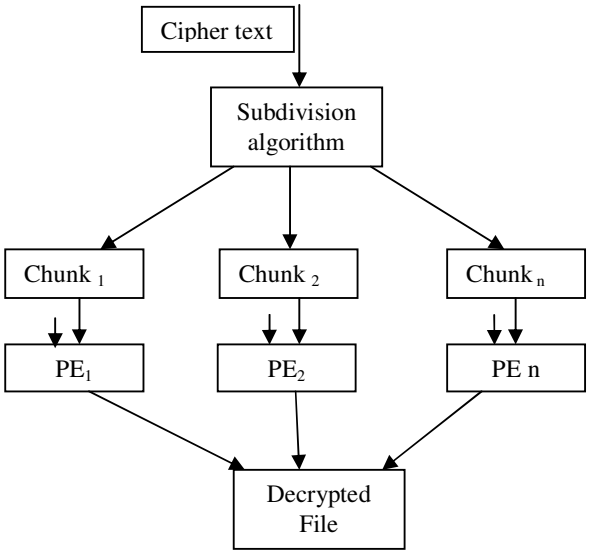

Figure 2: Parallel Decryption using threads

The decryption is done in the reverse order. The cipher text is taken as input. First input is divided into number of chunks and each chunk is taken as an input to thread. Thread decrypts the chunk of cipher data using reverse process of encryption. Output of each thread is collected to form the output file. Figure 2 shows the whole process in the pictorial way.

Here the method is tried on Caesar cipher method and transposition method on text file. The implementation is done on sequential and parallel way for the same input file. The key selection is done using user input.

C_p = Plain text + K_c (Key) Eq…. (1) for encryption process.

Plain text = $C_p + K_c$ (Key) Eq… (2) for decryption process.

The above equation (1) and (2) shows how encryption and decryption in done using single key on the given input text data.

5. IMPLEMENTATION

The implementation is done using python as a programming language. The main reason to select python is, it open source, supports for parallel programming and easy to implement.

In sequential implementation the whole file is taken as input along with the key. The Caesar cipher algorithm is applied and result of the same is written to output file. For the same output file decryption is applied and obtained the plain text output. The time taken for both the processes is recorded for comparison with parallel method.

The parallel implementation is done for encryption using threads. The implementation steps are as given bellow.

Step 1: Read the given plaintext file.
Step 2: Split the file into number of chunks.
Step 3: Create the threads and assign each chunks to the created threads.
Step4: All the chunks are encrypted in parallel using threads.
Step5: Write the result to a new file which is an encrypted file.

Given plaintext file is divided into number of chunks and each chunk is assigned to individual thread. Content of each thread is encrypted separately. Finally, outputs of all the threads are combined together and the result is written to a new file which is an encrypted file (cipher text).
The parallel implementation is done for decryption using threads. The implementation steps are as given bellow.

Step 1: Read the given cipher text file.
Step 2: Split the file into number of chunks.
Step 3: Create the threads and assign each chunks to the created threads.
Step 4: All the chunks are decrypted in parallel using threads.
Step 5:Write the result to a new file which is a decrypted file.

The cipher text file is divided into number of chunks and each chunk is assigned to individual thread. Content of each thread is decrypted separately, finally outputs of all threads are combined together and the result is written to a new file which is a decrypted file (plain text). Time taken is recorded for comparison.

The times recorded for sequential and parallel method are compared. The comparison results that, parallel process is much faster than the sequential process. It is observed that the parallel process is faster, but it works fine for limited number of threads. If thread number exceeds some limit, it

gives worst performance when compared with the sequential processing in limited number of cores. It is possible to overcome this by using more number of cores.

6. RESULTS AND ANALYSIS

The sequential method takes more time as the size of file is increased. When it goes to some particular stage, it will become constant and it will not change. The same method has been implemented using threads, where it is observed that performance is improved compared to sequential method. Using threads execution time can be reduced such that it reduces half of time compared to the sequential execution up to some extent, but as the number of threads increases the performance will be degraded. The input text is as shown in the figure 3 and the encrypted text is as shown in the figure 4 and the figure 5 shows the decrypted text after the decryption. This is shown in the figure 6 and more threads will leads to thread overhead. The figure 6 shows the same performance when the numbers of threads are 32 and if threads are 64 it takes more time than sequential execution.

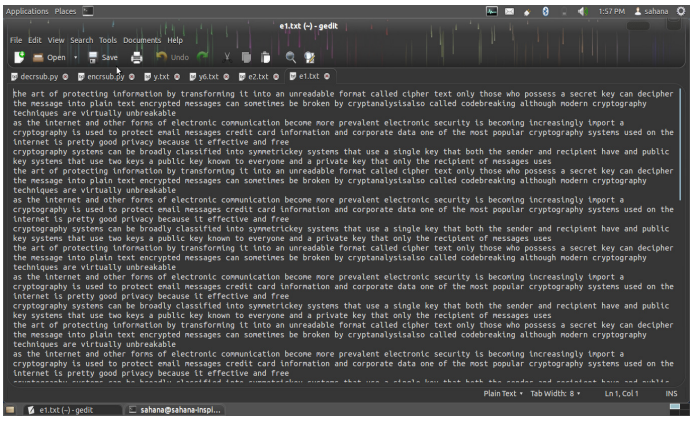

Figure 3: Plain text

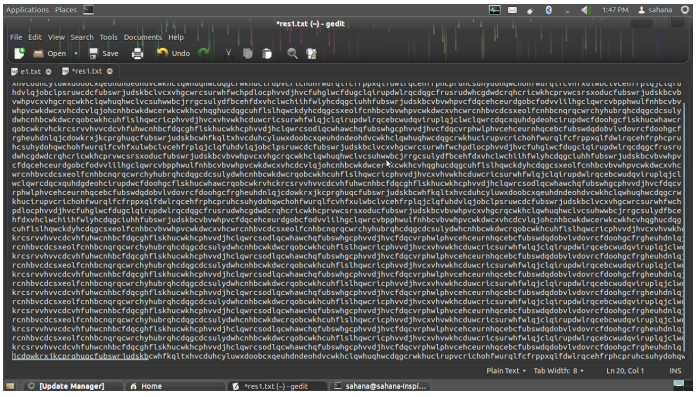

Figure 4 : Cipher text

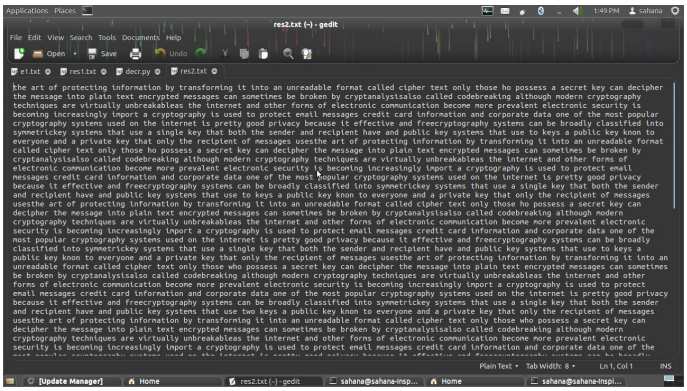

Figure 5: Decipher text

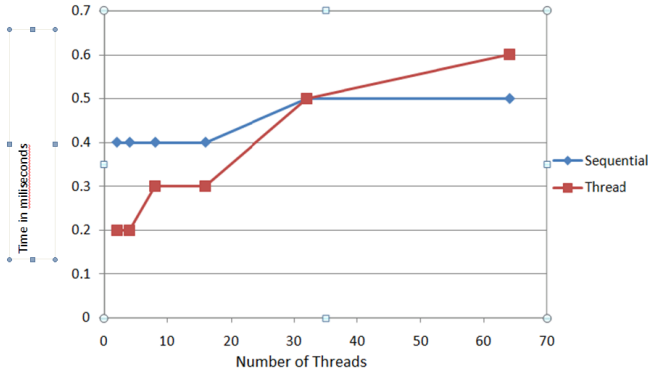

Figure 6: Comparison of Sequential vs parallel crypto

7. CONCLUSION

This paper provides implementation of encryption and decryption algorithm for text file using different cryptographic method using python as programming language. The encryption and decryption are implemented for Caesar cipher and subdivision algorithm. The time taken by sequential and parallel methods suggests that, using threads it is possible to achieve parallelism to improve the performance of encryption algorithms. The same comparison may be done for different algorithms and for different input formats.

REFERENCES

[1] Karthikeyan .S, Sairamn, Manikandan .G, Sivaguru J, “A Parallel Approach for Improving Data Security”, Journal of Theoretical and Applied Information Technology , Vol. 39 No.2, 15 May 2012, p . no 119-125.

[2] Osama Khalifa [2] “ The performance of cryptographic algorithms in the age of Parallel computing”, M.sc thesis, August-2011, Heriot Watt University School Of Mathematical and Computer Science.

[3] Vinodh Gopal , Jim Guilford, Wajdi Feghali, “Cryptographic Performance on the 2nd Generation Intel® Core™ processor family”, white paper - 2011.

[4] H. Naveen, M. Ramesh [4] “Parallel AES Encryption Engines for Many-Core Processor Arrays”, International Journal of Innovative Research in Computer and Communication Engineering, (An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

[5] M. Tahghighi, S. Turaev, R. Mahmod, A. Jafaar and M. Md. Said, "The Cryptanalysis and Extension of the Generalized Golden Cryptography", IEEE conference on open system, September 2011, Lankawi, Malaysia.

[6] Joseph Raphael, Dr. V. Sundaram, "Secured Communication through Fibonacci Numbers and Unicode Symbols", International Journal of Scientific & Engineering Research, Volume 3, Issue 4, April-2012, ISSN 2229-5518.
[7] Salem Sherif Elfard. "University Bulletin – ISSUE " No.- 15 – Vol . 2- 2013
[8] K C Ravishankar and M G Venkateshmurthy, " Pixel Compaction and Encryption for Secure Image Transmission" , National Conference on Intelligent Data Analytics and Pattern Discovery -2007, BIT Sathyamangalam, March 15-16, 2007.

AUTHORS

Mr. Raghu M E has got B.E. from UBDTCE, Davangere in 1998, M.Tech from JNNCE, Shivamogga, in 2003. He served at BCE from 2000-2003 and in JNNCE, Shimoga, Karnataka India from 2003-2010. He is currently serving as Associate Professor of CSE in GEC, Hassan. His areas of interest include Cryptography, Compiler Designs, Image Processing and Computer Graphics. He has 3 International and 3 national publications to his credit.

K. C. Ravishankar has got his B.E. from MCE, Hassan in 1990, M.Tech from IIT, Delhi in 1998 and Ph.D. from Visvesvaraya Technological University in 2009. He has served at Malnad College of Engineering, Hassan, and Karnataka India from 1990-2010. He is currently serving as Professor and Head of CSE in GEC, Hassan. His areas of interest include Databases, Image Processing and Cryptography. He has 4 International and 12 national Publications to his credit. He is Guiding 4 PhD students.