ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

Secure File Transmission Using Parallel Aes Algotrithm

Vedant Digraskar, Anushree Chirde, Ankita Jagtap, Aishwarya Deasi

Diploma Student, Department of Computer, AISSMS, Pune, India

ABSTRACT: Maintain security on the network is a very challenging task. Cryptography provides authenticity and integrity. The cryptography has many advantages such as less memory requirement, requires less execution time and computation power. However, we can't get desired outcomes by using sequential computation. In this paper, we introduce a new approach to computation using multicore processors by parallelizing the execution of the encryption algorithm in multiple cores/processors. We know that Advanced Encryption Standard (AES) is very efficient algorithm so we use effectiveness of the algorithm on dual core processor by using Open MP API to reduce the execution time. We are using JAVA platform for development. Based on our results, we conclude that parallel computation reduced execution time as compared with sequential computation.

KEYWORDS: Cryptography, Encryption Decryption parallel computing, Dual-core processor, AES, Open MP, Parallel computation.

I. INTRODUCTION

Nowadays computer networks are becoming more popular for transmitting and receiving information, but since it's wide use, it has become an important factor to take care of. Most importantly we need to secure the data transmission happens between networks. Data encryption i.e. Cryptography is the best mechanism to protect information and save the system from loss. Cryptography is the best way to secure sensitive information. The main purpose of cryptography is to convert important information into an unreadable form to all other than the intended receiver. These days AES is mostly used as the best encryption algorithm for hiding sensitive information from unintended users but it has some limitations such as memory requirement and computation time. The best way to overcome this problem is to use parallel execution method for AES algorithm. Parallel computing is a type of computation in which many calculations or the execution of processes are carried out simultaneously. Large problems can often be divided into smaller ones, which can then be solved at the same time [9]. It can be performed by using multicore and multiprocessor computers having multiple processing elements in a single computing machine. Open MP (Open multiprocessing) is one of the API which is supported by multicore architectures to provide multithreaded shared memory parallelism. Using this architecture we can surely parallelize execution of AES algorithm by dividing tasks between different cores which reduces total execution time.We formatted our paper as Section II gives an overview of literature survey. Section III represents the proposed system and Section IV gives experimental results and paper is concluded in Section V

II. LITERATURE SURVEY

In this section, we are going to discuss features of Advanced Encryption Standard algorithm and existing system. We developed a system which performs faster than the existing system. One of the conventional methods in enhancing the encryption and decryption of the plain text is by double encryption and double decryption [2]. In this method, to get cipher text from plain text one need to encrypt plain text twice also need to decrypt twice to get original text. Since this takes the huge time it needs to eradicated. The time requires for this is a huge drawback of this system so we need to overcome this problem and same we did into our system. The most common way of implementation of encrypting the data that is converting the plain text into cipher text and decrypting the data is by using the single core system [3]. Only one core is used without considering the size of the file to encrypt or decrypt. So this clears that it will definitely take huge time to complete encryption. It is best for a file with small size but not with big fat files. So to overcome these

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

problems and to get a best possible outcome, a parallel core processing is introduced in this paper. With this method, we made improvements in the conventional methods by reducing the run time of encryption and decryption. AES is a symmetric key block-oriented cryptography algorithm [1].AES block cipher has 128, 192 or 256-bit keys to encrypt or decrypt data in blocks of 128-bits. Advanced Encryption Algorithm has a separate key expansion phase for the expansion of 128, 192 or 256-bit keys so that these keys can be used in multiple rounds of encryption and decryption process [4] [7].

Steps for Encryption; each round consists of the four steps:

A. Sub-Bytes
B. Shift-Rows
C. Mix-Columns
D. Add-Round-Key

Sub-Bytes A non-linear substitution step where each byte is replaced with another according to a lookup table (S-box). This stage is basically a table lookup utilizing a 16×16 matrix of byte values called an s-box. This matrix comprises of all the possible combinations of an 8-bit sequence (2*8 = 16 × 16 = 256) [3] [8]. Then again, the s-box is not only a random permutation of these quantities and there is an all-around characterized method for making the s-box tables [9]. Again the matrix that gets worked upon all through the encryption is known as the state.
This transformation made of two steps:
(i). Multiplicative inverses of each byte in the state.
(ii). The result in this step is obtained from step (i) by transforming y = f(x)

Shift-Rows A transposition step where each row of the state is shifted a cyclically certain number of times. Shift row transforms the line of the state which increasing the offset of circulation moves left, the first line unchanged. Second line loop left 1 byte, third line loop left 2 bytes, similarly fourth line loop 3 bytes [3][1]. The Inverse Shift Rows transformation performs these circular shifts the other way for each of the last three lines.

Mix-Columns A mixing operation which operates on the columns of the state, combining the four bytes in each column. Mixing the data within each column of the State array. It makes confuse transforms to columns in the state. In this the data of state column as 32 bit and then carriers on the matrix multiplication transformation in it. Effectively a matrix multiplication in GF (28) using prime poly m(x) =x8+x4+x3+x+1.

Add-Round-Key Each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule. In this stage, the 128 bits of state are bitwise XOR with the 128 bits of the round key. The operation is seen as a column-wise operation between the 4 bytes of a state column and single word of the round key. This transformation is as straightforward as would be prudent which helps in efficiency however it also affects all of the states. The value of add round key transformation is pseudo C code.

```
Round (state, round key[i])
{
Byte sub (state);
Shift row (state);
Mix column (state);
Add round key (state, round key[i]);
}
```

III. PROPOSED SYSTEM

We used OPENMP API for parallel implementation of AES i.e. Advanced Encryption Standard which enables us to use multicore architecture to operate. By **Vangie Beal** In consumer technologies, multi-core is usually the term used to describe two or more CPUs working together on the same chip. Also called multicore technology, it is a type of architecture where a single physical processor contains the core logic of two or more processors [5] [8]. In this type

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

system, each core has its own execution pipeline and resources which are needed for execution. Figure 1 shows multi-core system, it has N number of processing elements (core) integrated onto a single chip. Each processing core has its own L1 cache and shares a common L2 cache. This system supports simultaneous multithreading. Simultaneous multithreading which permits threads to execute independently on the same core. In multicore systems, a number of threads can execute multiple numbers of tasks simultaneously

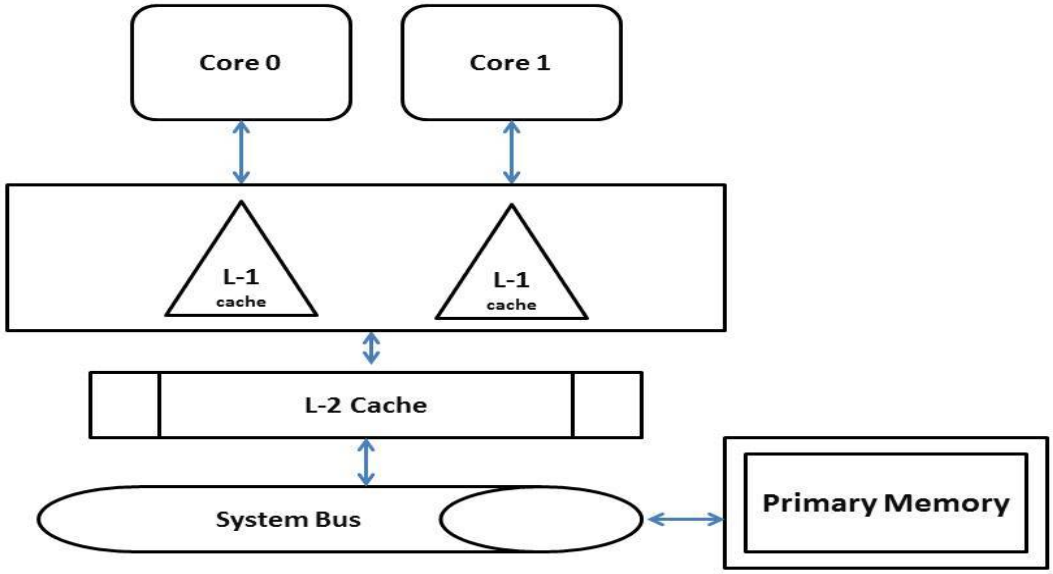

Figure 1 System Flow

Open MP uses Fork-Join model for execution. Every time OpenMP program start its execution it begins with a thread called master thread. This thread will execute in a single region until the first parallel construct is encountered [10]. When parallel construct encounters, the master thread N number of threads. Now those statements which are enclosed in parallel region are allotted to these threads and start execution.After the execution of all the statements within the parallel region, threads will terminate and leave only the master thread. Open MP is mainly comprised of three components: Compiler directives, Runtime library routines, and Environment variables. We used Open MP APIs to implement AES Cryptography algorithm. Here we have parallelized the cryptography mechanism by using OpenMP directives between two core to reduce the execution time of the program.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

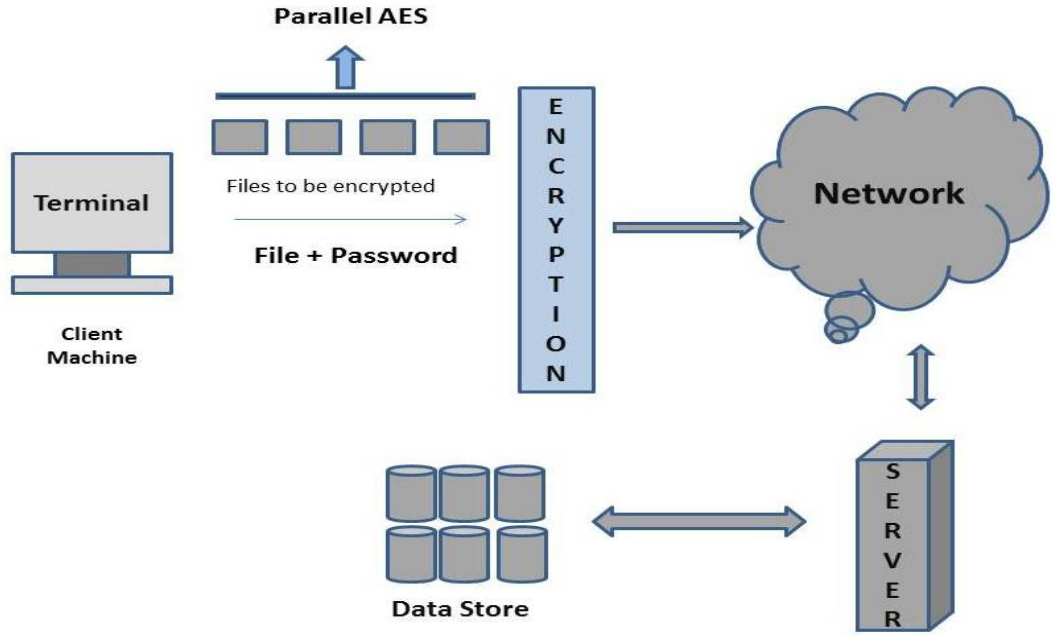

Figure 2 Proposed System Architecture

We have created an application to securely transmit file between network or to store the file on the server via a network. From the client machine, we interact with the system via GUI which is created with JAVA language. We used OpenMP APIs to create parallel AES algorithm. First of all, we logged into the system and then select a file to be transmitted then we upload the file after that file gets passed to our encryption engine which parallelly converts the file into encrypted file and then it gets stored in the database. We use MongoDB as a database since it provides a greater amount of security. When we need to get back file we just need to download that file in this process file gets fetched from M ongoDB to server here it gets decrypted and forwarded to intended user. Above figure shows the flow of system we pass the file to encryption engine then this file gets equally divided and gets assigned to threads for converting plain text to cipher text, each thread performs AES algorithm steps to get ciphertext. Now after encryption, all part gets combined and the whole file is represented as a complete encrypted file. For Decryption we need to follow the same method, we pass encrypted file to decryption engine this engine divides the file into equal parts then master thread assign each part to each thread which is running on the individual core. After decryption, all parts get combined and we get a final file which is a plain text file.

IV. RESULTS AND ANALYSIS

Results mentioned here are based on the different file using AES and Parallel AES algorithm.
Hardware Description for machines used.
Intel Dual Core, 2GB RAM with Ubuntu Operating system.
IDE: NetBeans,
Programming Language: JAVA
Database: MongoDB

Comprative results

Input File (Size in MB)	Sequential execution result	Parallel execution result
1	2010ms	1089ms
10	3125ms	1578ms
150	7829ms	3252ms

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

V. CONCLUSION

In this paper, we explained our work in parallel computing of AES algorithm. We implemented AES algorithm using OpenMP API. We have successfully show that parallel computing always results in faster execution. We tested files with different size and for all files, we got a better result for parallel AES. So we conclude that Parallel AES performs 40% to 55% than sequential AES algorithm.

REFERENCES

[1] https://en.wikipedia.org/wiki/Parallel_computing
[2] http://www.webopedia.com/TERM/M/multi_core_technology.html
[3] J. Pichel, D. E. Singh, and J. Carretero. Reordering algorithms for increasing locality on multicore processors. 10th IEEE International Conference on High Performance Computing and Communications, 2008, pages 123-130, 2008
[4] Yeong Chee Mei; Naziri, S., "The FPGA implementation of multiplicative inverse value of GF(2 8) generator using Extended Euclid Algorithm (EEA) method for Advanced Encryption Standard (AES) algorithm," in computer Applications and Industrial Electronics (ICCAIE), 2011 IEEE International Conference on , vol., no., pp.12-15, 4-7 Dec. 2011 doi: 10.1109/ICCAIE.2011.6162095.
[5] Moh'd, Abidalrahman, Yaser Jararweh, and L. Tawalbeh. "AES-512: 512-bit Advanced Encryption Standard algorithm design and evaluation." In Information Assurance and Security (IAS), 2011 7th International Conference on, pp. 292-297. IEEE, 2011.
[6] Rewagad, P.; Pawar, Y., "Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in 2011 Cloud Computing," in Communication Systems and Network Technologies (CSNT), 2013 International Conference on , vol., no., pp.437-439, 6-8 April 2013 doi:10.1109/CSNT.2013.97
[7] Pearson, S.; Benameur, A., "Privacy, Security and Trust Issues Arising from Cloud Computing," in Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on , vol., no., pp.693-702, Nov. 30 2010-Dec. 3 2010 doi: 10.1109/CloudCom.2010.66
[8] Hussen F, Elleithy K, Razaque A. Implementation of Fault Tolerance Algorithm to Restore Affected Nodes in Scheduling Clusters. International Journal of Computer Networks & Communications. 2012 Jan 1;4(1):1.
[9] A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, New York, 1997,p. 81-83. J. Nechvatal, et. al., Report on the Development of the Advanced Encryption Standard (AES), National Institute of Standards and Technology, October 2, 2000.
[10] amed, E.M.; Abdelkader, H.S.; El-Etriby, S., "Enhanced data security model for cloud computing," in Informatics and Systems (INFOS), 2012 8th International Conference on , vol., no., pp.CC-12-CC-17, 14-16 May 2012 Chih-Chung Lu; Shau-Yin Tseng, "Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter," in Application-Specific Systems, Architectures and Processors, 2002. Proceedings. The IEEE International Conference on , vol., no., pp.277-285, 2002 doi: 10.1109/ASAP.2002.1030726