Performance Evaluation of AES algorithm on Supercomputer IMAN1

Sanad AbuRass
Department of Computer science
King Abdullah II School for Information Technology
University of Jordan, Amman –Jordan

Mohammad Qatawneh
Department of Computer science
King Abdullah II School for Information Technology
University of Jordan, Amman –Jordan

ABSTRACT

Advanced Encryption Standard (AES) is one of the most popular encryption algorithms. The algorithm uses a combination of Exclusive-OR operations (XOR), octet substitution with an S-box, row and column rotations, and a Mix Column. In this paper the parallel implementation of AES cryptography algorithm is evaluated and compared in terms of running time, speed up and parallel efficiency. The parallel implementation of AES is implemented using message passing interface (MPI) library, and the results have been conducted using IMAN1 Supercomputer. The experimental results show that the run time of AES algorithm is decreased as the number of processors is increased. Moreover, the speedup for the data size of 16, 32, 64, 128, 256, and 1024-KB is increased when the number of processors is equal to 2, 4, 8, and 16.

Keywords

AES Encryption, MPI, Supercomputer, Parallel Computing.

1. INTRODUCTION

Advanced Encryption Algorithm (AES) is a symmetric block cryptographic algorithm designed by Rijmen [1]. Because of its high efficiency and the ease of implementation, AES becomes one of the most widely used cryptographic algorithm for several applications, such as high- performance database servers, and secure communication systems. For AES selected three members of Rijndael family, each with a block size of 128 bits, but three different key lengths: 128, 192 1nd 256 bits. AES considered to be an iterative rather than Feistel cipher [2]. In general, AES algorithm processes data as block of 4 columns of 4 bytes operates on entire data block in every round [3].

Parallel and distributed computing systems are high-performance computing systems that spread out a single application over many multi-core and multi-processor computers in order to rapidly complete the task. Parallel and distributed computing systems divide large problems into smaller sub-problems and assign each of them to different processors in a typically distributed system running concurrently in parallel [4][5] [6] [7] [8][9][10].

High speed implementation of AES is crucial to boost performance of security applications. Mainstream implementation of AES including ASIC [11], GPUs [12], and CPUs [13]. In this paper, the Performance Evaluation of AES algorithm on Supercomputer IMAN1 is presented. The evaluation is done in terms of the running time, speed and parallel efficiency according to different data size and different number of processors. The results were conducted using IMAN1 supercomputer which is Jordan's first and fastest supercomputer. It is available for use by academia and industry in Jordan and the region and provides multiple resources and clusters to run and test High Performance Computing (HPC) codes [7].

This paper is organized as follows: AES algorithm is analyzed in Section 2. In section 3, the proposed scheme of the parallel AES implementation is presented. Section 4 presents the experimental results and comparison. Finally, section 5 concludes the paper.

2. AES ALGORITHM

AES is based on a design principle known as a substitution-permutation network, a combination of both substitution and permutation, and is fast in both software and hardware. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification *per se* is specified with block and key sizes that may be any multiple of 32 bits, with a minimum of 128 and a maximum of 256 bits. For encryption, there are four basic blocks of operation, i.e., Byte Substitution (1 S-box used on every byte), Shift rows (permute bytes between groups/columns), Mix columns (subs using matrix multiply of groups), and Add round key (XOR state with key material).

AES encryption starts with an Add Round Key block. Then followed 9 rounds consisting of 4 blocks. The tenth round of 3 blocks without Mix Columns is performed finally to get ciphertext, as shown in Fig. 1.

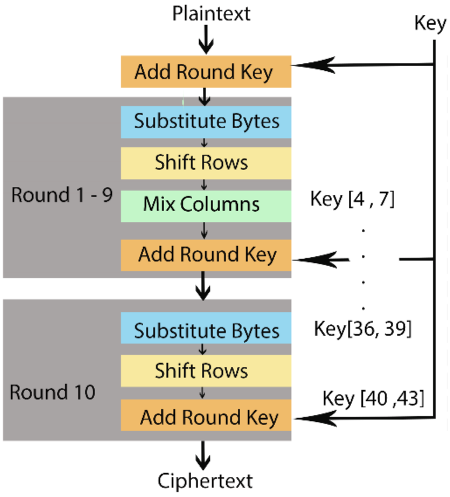

Fig 1. Block diagram of AES encryption.

3. PARALLEL IMPLEMENTATION OF AES

In this section, the IMAN1 is introduced firstly. And then the implementation is demonstrated both sequential and parallel to evaluate the execution time, speedup and efficiency. The IMAN1 is a Jordan's first and fastest High Performance Computing resource, funded by JAEC and SESAME. It is available for use by academia and industry in Jordan and the region.

The sequential and parallel implantation have been conducted with different packet sizes (16, 32, 64, 128, 256, 512 and 1024 KB), and the key size of 16 bytes.

To parallelize the AES computation, we need to partition the text file into n blocks, where n is the number of processors. Each processor will encrypt the block using encryption key and sent it back to the master processor, and finally, the master processor gets all ciphertext blocks from all worker processors and put them in one file. The interval method is used to distribute the input file among several processors. This method will determine a specific segment for each processor using the PROCESSOR_ID and the number of processors that are used. We used two variables to determine the interval: INTERVAL_START = PROCESSOR_ID * NUMBER_OF_PROCESSORS, and INTERVAL_END = INTERVAL_START + NUMBER_OF_PROCESSORS. Each processor will encrypt its segment if and only if the ith iteration is greater than or equal INTERVAL_START and less than INTERVAL_END. In this way each Processor will encrypt specific segment.

4. EXPERIMENTAL RESULTS

The parallel AES algorithm is implemented using open Message Passing Interface (MPI) library, and executed on IMAN1 supercomputer. MPI library provides different functions to support distributing data among different processors to be processed simultaneously. The MPI_Scatterv procedure is used to split and distribute the input data on processors. Every processor executes the same code with the same key for all concurrently on data portion which allocated to it. Parallel AES evaluated by multiple input sizes (16, 32, 64, 128, 256 and 512), and different number of processors (1, 2, 4, 8, 16, 32, 64, and 128).

The execution time of the AES encryption algorithm on different number of processors with various packet sizes shown in Fig. 2. The figure shows that as the number of processors increase the encryption time decrease, due to the data distribution among the processors, and this is clear when the number of processors is equal to 2, 4, 8, and 16 processors.

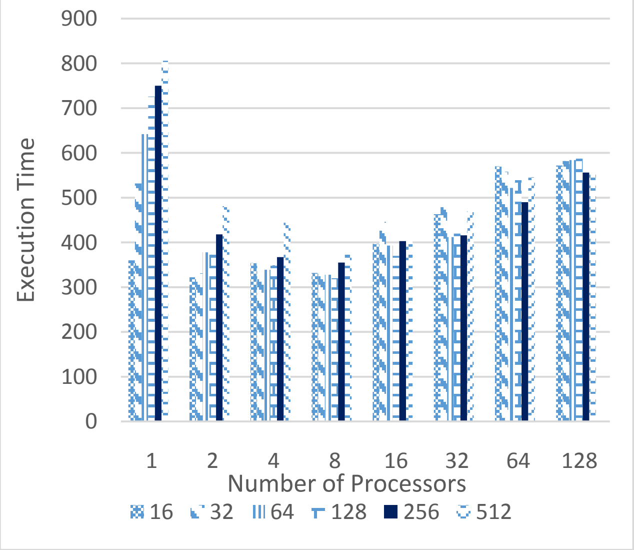

Fig 2. The execution time of encryption algorithm

The speedup is calculated by taking the ratio between the serial and parallel time. According to the results in Fig.3 the speedup for the data size of 512-KB is increased when the number of processors is equal to 2, 4, 8, and16, and the speedup decreases when the number of processors is more than 16.

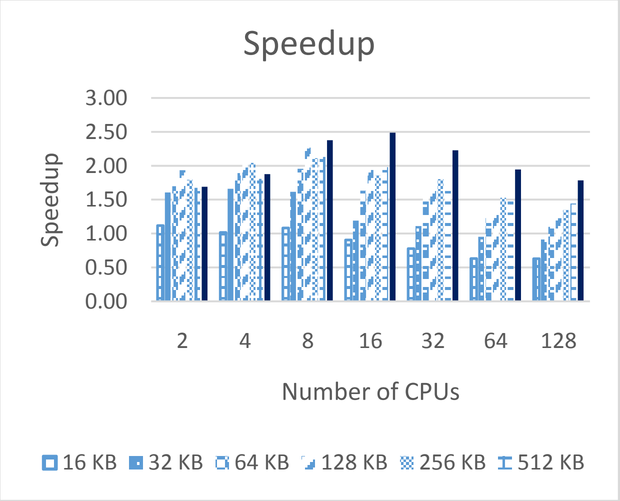

Fig 3. The Speedup of the AES Algorithm

Parallel efficiency is computed by taking the ratio between speedup and number of processors. Fig. 4 describes efficiency of the AES algorithm for different packet sizes. The results show that the efficiency is decreased with increasing the number of processors on all evaluated packet sizes.

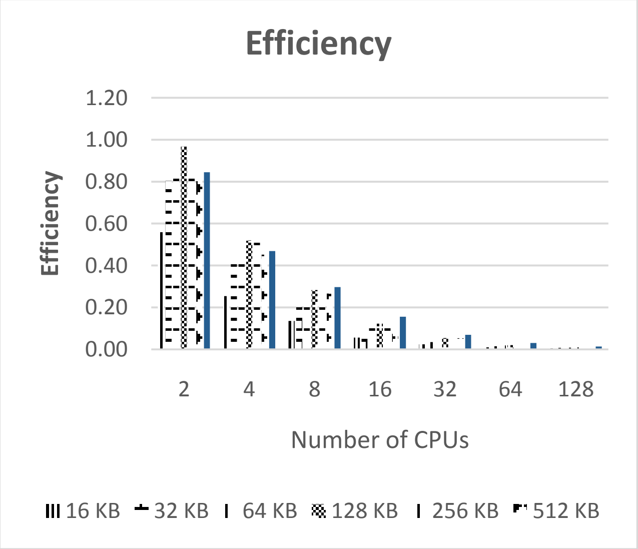

Fig 4. The Efficiency of the AES algorithm

As the number of CPUs increase the speed up and the efficiency will decrease and the reason behind that is the communication overhead between the CPUs. In parallel computing, the CPUs must communicate with each other in order to solve the task. When using huge number of CPUs on relatively small tasks, the communication between the CPUs will be more than the computation of the task, which will result in a decrease in the speed up and the efficiency [14]. So, it is recommended to use suitable number of CPUs relative to the size of the tasks.

5. CONCLUSION

Performance of parallel AES was evaluated according to execution time, speedup and efficiency for different packet sizes of data on different number of processors. According to results, the parallel AES achieves better execution time for large date size. The experimental results show that the running time will be decreased, and the speed-up of the algorithm encryption will be increased when the number of processors ranges from 2 to 16. In the future, this method might be used in cloud environments to encrypt and decrypt the data.

6. REFERENCES

[1] Joan Daemen, Vincent Rijmen, The Design of Rijndael: AES – The Advanced Encryption Standard, Springer, 2002. ISBN 3-540-42580-2.

[2] Le, D., Chang, J., Gou, X., Zhang, A. and Lu, C., 2010, April. Parallel AES algorithm for fast data encryption on GPU. In Computer Engineering and Technology (ICCET), 2010 2nd International Conference on (Vol. 6, pp. V6-1). IEEE.

[3] Di Natale, G., Doulcier, M., Flottes, M.L. and Rouzeyre, B., 2009. A reliable architecture for parallel implementations of the advanced encryption standard. Journal of Electronic Testing, 25(4-5), pp.269-278.

[4] Maha Saadeh, Huda Saadeh, and Mohammad Qatawneh. Performance Evaluation of Parallel Sorting Algorithms on IMAN1 Supercomputer, International Journal of Advanced Science and Technology, 95, 2016, pp. 57-72.

[5] Qatawneh Mohammed. Embedding Linear Array Network into the tree-hypercube Network, European Journal of Scientific Research, 10(2). 2005, pp. 72-76.

[6] Mohammad Qatawneh, Ahmad Alamoush, Ja'far Alqatawna. Section Based Hex-Cell Routing Algorithm (SBHCR), International Journal of Computer Networks & Communications (IJCNC), 7(1). 2015.

[7] Mohammad Qatawneh. Multilayer Hex-Cells: A New Class of Hex-Cell Interconnection Networks for Massively Parallel Systems, International journal of Communications, Network and System Sciences, 4(11). 2011.

[8] Mohammad Qatawneh. Embedding Binary Tree and Bus into Hex-Cell Interconnection Network, Journal of American Science, 7(12). 2011.

[9] Qatawneh Mohammad, Hebatallah Khattab. New Routing Algorithm for Hex-Cell Network, International Journal of Future Generation Communication and Networking, 8(2). 2015.

[10] Mohammad Qatawneh. New Efficient Algorithm for Mapping Linear Array into Hex-Cell Network, International Journal of Advanced Science and Technology, 90, 2016.

[11] S.K. Mathew, et al. 53 Gbps Native GF (24)2 Composite-Field AES-Encrypt/Decrypt Accelerator for Content-Protection in 45 nm High-Performance Microprocessors, JSSC, vol.46, no.4, p.767-776, April (2011).

[12] J. Ortega, et al. Parallelizing AES on multicores and GPUs, Electro/Information Technology (EIT), 2011 IEEE International Conference on, May (2011).

[13] M. Kumar and A. Singhal. Efficient implementation of Advanced Encryption Standard (AES) for ARM based platforms, Recent Advances in Information Technology, 2012 1st International Conference on, March (2012).

[14] Quinn, M.J. and Quinn, M.J., 1994. Parallel computing: theory and practice (Vol. 2). New York: McGraw-Hill.