Performance Evaluation of Parallel International Data Encryption Algorithm on IMAN1 Super Computer

Dr.mohammad Qatawneh

International Journal of Network Security & Its Applications

PERFORMANCE EVALUATION OF PARALLEL INTERNATIONAL DATA ENCRYPTION ALGORITHM ON IMAN1 SUPER COMPUTER

Ahmad Bany Doumi and Mohammad Qatawneh

Department of Computer Science-King Abdullah II School for
Information technology, University of Jordan, Amman-Jordan.

ABSTRACT

Distributed security is an evolving sub-domain of information and network security. Security applications play a serious role when data exchanging, different volumes of data should be transferred from one site to another safely and at high speed. In this paper, the parallel International Data Encryption Algorithm (IDEA) which is one of the security applications is implemented and evaluated in terms of running time, speedup, and efficiency. The parallel IDEA has been implemented using message passing interface (MPI) library, and the results have been conducted using IMAN1 Supercomputer, where a set of simulation runs carried out on different data sizes to define the best number of processor which can be used to manipulate these data sizes and to build a visualization about the processor number that can be used while the size of data increased. The experimental results show a good performance by reducing the running time, and increasing speed up of encryption and decryption processes for parallel IDEA when the number of processors ranges from 2 to 8 with achieved efficiency 97% to 83% respectively.

KEYWORDS

International Data Encryption Algorithm (IDEA); Plain text; Encrypted data; MPI.

1. INTRODUCTION

The communication between different devices on computer networks can be a serious issue which it motivated various researchers to develop secure communication to protect data exchanged between both sender and receiver. They have proposed cryptography algorithms like DES, AES, RC2 [14][15] and IDEA[1]. There are other issues which focus on improving the speed of cryptography algorithms to reduce end to end delay on networks.

Cryptography is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. IDEA is one of the ciphers which encrypt the text into an unreadable format and makes it secured in order to send it over to the network. The IDEA encryption algorithm provides high-level security which does not based on keeping the algorithm a secret, but rather upon ignorance of the secret key.

Many algorithms used in the encryption field which they are divided into symmetric and asymmetric encryption approaches. International Data Encryption Algorithm (IDEA) is symmetric secret-key cryptography ciphers algorithm to encrypt and decrypt date which has been

DOI: 10.5121/ijnsa.2019.11105

developed in 1991 by James L. Massey and Xuejia Lai [1]. Originally, this approach is used in different applications such as financial applications and named Improved Proposed Encryption Standard (IPES) that was described by [5].

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 [7][8] [9] [10] [11][12][13].

In this paper, the parallel international data encryption algorithm (IDEA) is implemented and the performance of it evaluated in term execution time, speed up, and parallel efficiency according to different data size and different number of processors using Message Parallel Interface (MPI) on IMAN1 supercomputer. 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. It uses 2260 PlayStation3 devices [7] [16].Our work has two limitations. First, it may yield different results with other programming languages and parallel frameworks. Second, it did not take in account that communication time between processors and processing time of processors can be separately calculated, rather, they were calculated by summing them as a single value.

The rest of this paper is organized as follows; Section 2 presents the background and related works. Section 3 presents the experiments and results, and Section 4 presents the conclusion.

2. BACKGROUND AND RELATED WORK

IDEA belongs to a class of secret-key cryptosystems which is characterized by the symmetry of encryption and decryption processes, and the possibility of implying the decryption key from the encryption key and vice versa [5---]. In the encryption process, the 64-bit plain text is divided by IDEA into four portions where each sub-blocks with 16 bits (P1 to P4) as shown in Figure 1. These four sub-blocks will proceed through eight rounds. In each round, every 16-bit of four blocks will be manipulated by different six sub-keys of 52 keys of 128-bits cipher key which will be agreed upon by sender and receiver. After completing eight rounds, the output of four blokes can be manipulated by the OUTPUT TRANSFORMATION phase. Moreover, in each round, data produced from previous round (P1 to P4) are input in current round and processed by logical and arithmetic operations with six sub-keys which are assigned to the round. Finally, OUTPUT TRANSFORMATION phase, contains arithmetic operations and just four sub-keys, where the final output produces cipher data (C1 to C4) divided to 16-bits sizes for each block [2] [3].

In the decryption phase, there are different approach used in divide the Key to sub-keys and same techniques used in the encryption process are used. Logical and arithmetic operations of each round are Multiplication modulo 216 +1, Addition modulo 216 and XOR. But, in final phase is Multiplication module 216 +1 and Addition modulo 216 [3]. These operations will be applied on (P1 to P4) blocks by assigned keys.

Figure 2 shows IDEA steps of logical and arithmetic operations executed in each round to produce encrypted data to be input of next round. The steps are summarized as following:

(1) Multiply X1 and Z1.
(2) Add X2 and Z2.
(3) Add X3 and Z3.
(4) Multiply X4 and Z4.
(5) Bitwise XOR the results of steps 1 and 3.
(6) Bitwise XOR the results of steps 2 and 4.
(7) Multiply the result of step 5 and Z5.
(8) Add the results of steps 6 and 7.
(9) Multiply the result of step 8 and Z6.
(10) Add the results of steps 7 and 9.
(11) Bitwise XOR the results of steps 1 and 9.
(12) Bitwise XOR the results of steps 3 and 9.
(13) Bitwise XOR the results of steps 2 and 10.
(14) Bitwise XOR the results of steps 4 and 10.

But in the output transformation which starts at end the 8th round have four steps as following [4]:

(1) Multiply X1 and Z1.
(2) Add X2 and Z2.
(3) Add X3 and Z3.
(4) Multiply X4 and the Z4.

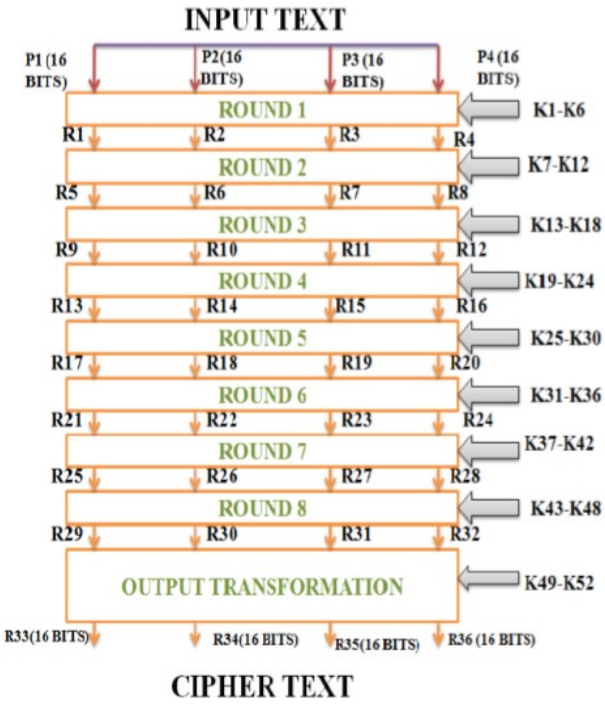


Figure 1. IDEA algorithm [2]

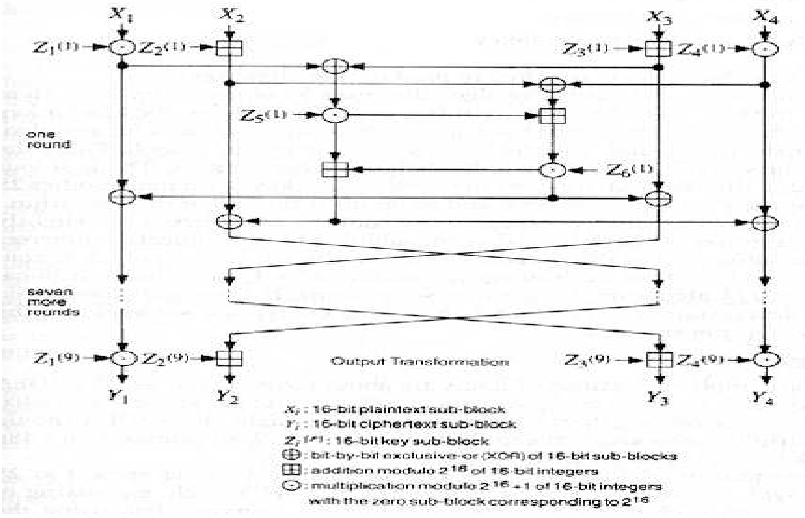

X_1
X_2
X_3
X_4
$Z_1(1)$
$Z_2(1)$
$Z_3(1)$
$Z_4(1)$
$Z_5(1)$
$Z_6(1)$
one round
seven more rounds
$Z_1(9)$
$Z_2(9)$
Output Transformation
$Z_3(9)$
$Z_4(9)$
Y_1
Y_2
Y_3
Y_4
X_i : 16-bit plaintext sub-block
Y_i : 16-bit ciphertext sub-block
$Z_i^{(r)}$: 16-bit key sub-block
⊕ : bit-by-bit exclusive-or (XOR) of 16-bit sub-blocks
⊞ : addition modulo 2^{16} of 16-bit integers
⊙ : multiplication modulo $2^{16}+1$ of 16-bit integers with the zero sub-block corresponding to 2^{16}

Figure 2. IDEA Structure of single round [4]

The cipher speed is one of the major functional features in cryptographic techniques, this feature is significantly important when they usually work on huge datasets, there are many studies and researchers striving to increase the speed of encryption techniques using parallelism.

In [16], the authors showed the performance evaluation of the blowfish algorithm in the parallel platform, the algorithm is implemented using the MPI library, and the experiment is performed on an IMAN1 supercomputer. The experimental results showed that the parallel algorithm achieved the best value when the number of processors is 32 for a plaintext size of 160 Mbyte.

The author in [17] used the parallel implementation for the encryption algorithm by using eight Quad-Core (32) Intel Xeon Processors 7310 Series - 1.60 GHz and the Intel C++ Compiler, the experimental results showed that the parallel encryption algorithm by multiprocessor from 2 to 32 processors improved the time of the data encryption and decryption.

The authors of [6] presented results of parallelizing the IDEA on 1 to 4 processors. The OpenMP standard was chosen for presenting the parallelism of the algorithm. The efficiency measurement for a parallel program was presented. They did not find the best number of processors to be used for different sizes of data.

In this paper, we implemented the IDEA algorithm on a parallel platform which is different from the above researches in architecture and the number of processors used. This work evaluated the parallel IDEA to find the best number of processors to be used for different sizes of data.

3. Experiments And Results

In parallel computations, the number of processors should be defined to run concurrently by writing special instructions in a programming language. In this section, Parallel IDEA results are evaluated according to performance in terms of execution time, speedup and efficiency of serial and parallel IDEA.

All experiments obtained by using IMAN1 cluster as hardware is "Dual Quad-Core Intel Xeon CPU with SMP, 16 GB RAM" and Scientific Linux 6.4 [7] with MPI library which implemented by C++ is used in parallel IDEA implementation.

MPI library provides different functions to support distributing data through different available machines to be processed by these machines simultaneously, MPI_Scatterv procedure is one of them which used in our implementation to split data in the same sizes where it responses to allocate each specific size of data to one processor. The IDEA implementation based on portioning data where every processor executes the same code with the same key for all concurrently on data portion which allocated to it. Parallel IDEA evaluated by multiple input sizes (0.09, 0.19, 0.39,0.78, 1.56,3.12, 6.25,12.5, 25,50 MByte) and different numbers of processors (1, 2, 3, 4, 8, 16, 32, 64, 128).

The assumptions in our work as following: first, all processors have same capacity and throughput. Second, execution time for plaintext size on assigned processors is taken from the processor which consumes larger time. Finally, data will be evenly partitioned on assigned processors.

3.1 ENCRYPTION AND DECRYPTION TIME EVALUATION

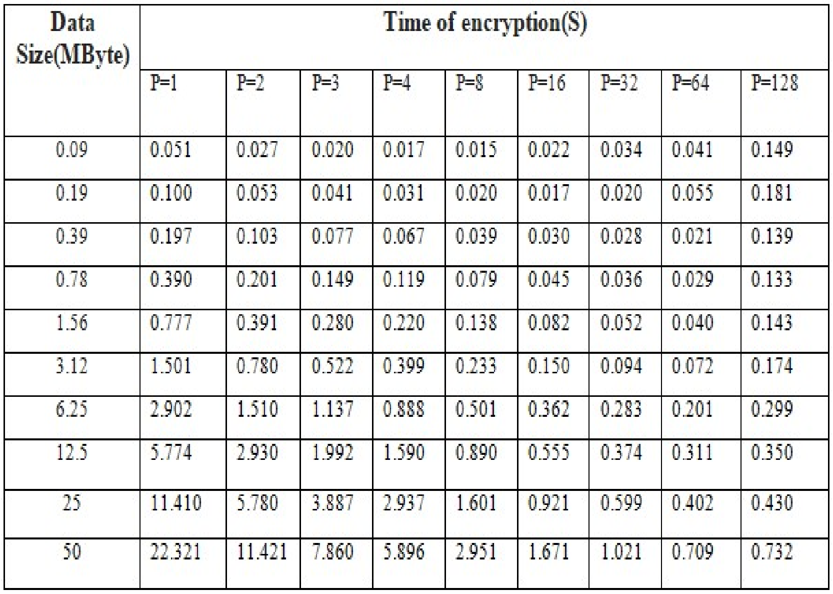
The experimental results were repeated 10 times for every number of processors then the average is taken and they were recorded in Table1 where it shows the encryption and decryption time of serial and parallel IDEA according to multiple input sizes.

Figure 3 shows that the execution time for the IDEA algorithm, using a single processor (sequential), increased while the plaintext size increased.

Figure 4, figure 5 and figure 6 illustrate execution time of different number of processors, we choose ten different data sizes from 0.09 up to 50 MB, which covers small and large data size, the figures show the behavior of parallel IDEA is the same for all input sizes and can be described in the following:

•When the number of processors increases, the encryption and decryption time decreases due to the work distributed among the processors. This is obvious when moving from 2, 3, 4, 8, 16, 32, or to 64 processors.

• The encryption and decryption time increases when using 128 processors due to the increase in overhead of communication of processors and needing for more time to split particular data size. Therefore, more processors on particular input size.



Data Size(MByte)	Time of encryption(S)								
	P=1	P=2	P=3	P=4	P=8	P=16	P=32	P=64	P=128
0.09	0.051	0.027	0.020	0.017	0.015	0.022	0.034	0.041	0.149
0.19	0.100	0.053	0.041	0.031	0.020	0.017	0.020	0.055	0.181
0.39	0.197	0.103	0.077	0.067	0.039	0.030	0.028	0.021	0.139
0.78	0.390	0.201	0.149	0.119	0.079	0.045	0.036	0.029	0.133
1.56	0.777	0.391	0.280	0.220	0.138	0.082	0.052	0.040	0.143
3.12	1.501	0.780	0.522	0.399	0.233	0.150	0.094	0.072	0.174
6.25	2.902	1.510	1.137	0.888	0.501	0.362	0.283	0.201	0.299
12.5	5.774	2.930	1.992	1.590	0.890	0.555	0.374	0.311	0.350
25	11.410	5.780	3.887	2.937	1.601	0.921	0.599	0.402	0.430
50	22.321	11.421	7.860	5.896	2.951	1.671	1.021	0.709	0.732

Table 1. Execution Time (s) of parallel IDEA Compared by Num. of Processor and Size of data

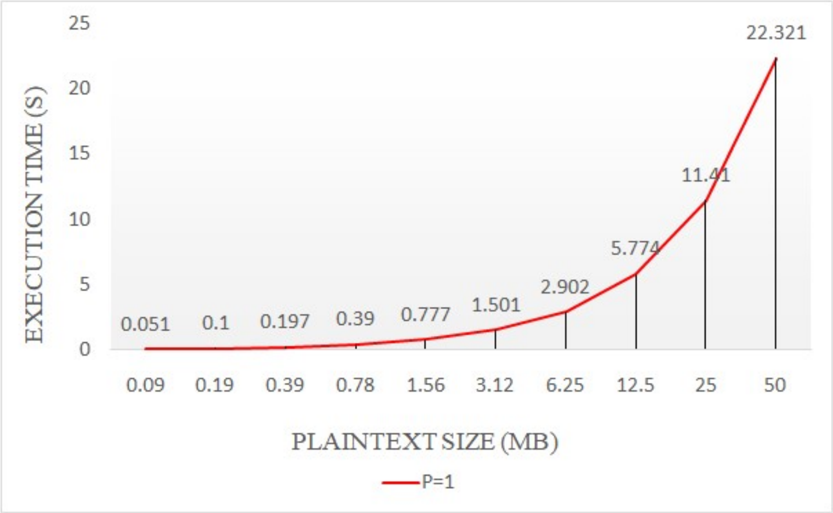


Figure 3. The execution time (s) for sequential IDEA for different plaintext sizes.

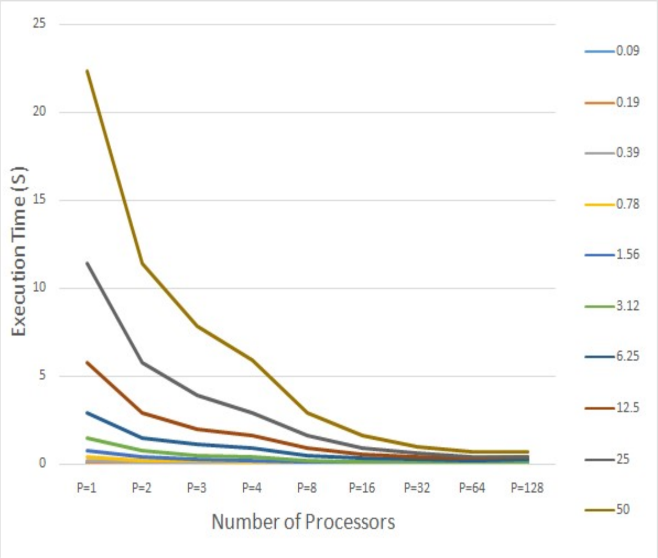


Figure 4. Comparison the effect on execution time with varying the number of Processors on different plaintext sizes

3.2 Speedup Evaluation

Is calculated by taking the ratio between the serial and parallel time [7]. According to Figure 7 which shows the speed up to five different plaintext sizes on2, 3, 4, 8, 16, 32, 64, and 128 processors. The results show as following:

• The speedup increases when the number of processors increases, and the increments are the same for all plaintext sizes on processors 2 to 16.

• When the number of processors equals 32 or 64, the speedup for these plaintext sizes (1.56, 3.12, 25, and 50 MByte) are increased while the speed up for 0.19 MByte is decreased.50 MByte size achieved the best speedup value as compared with other plaintext sizes on all number of processors.

• When the number of processors equals 128 the speed up is smaller for all evaluated plaintext sizes. So the speed up for the large size of data is better when using the number of processors larger than 16.

3.3 Parallel Efficiency

Is computed by taking the ratio between speedup and number of processors [6]. Figure 8 shows the parallel efficiency of parallel IDEA algorithm for different plaintext sizes on a different number of processors. The results show that the parallel efficiency of a parallel IDEA algorithm is the best when the number of processors equals 2, 3, 4, and 8. The parallel efficiency decreases when the number of processors increases from 16 to 128. Finally, the large plaintext sizes achieved the best and kept on high-efficiency values across the different number of processors.

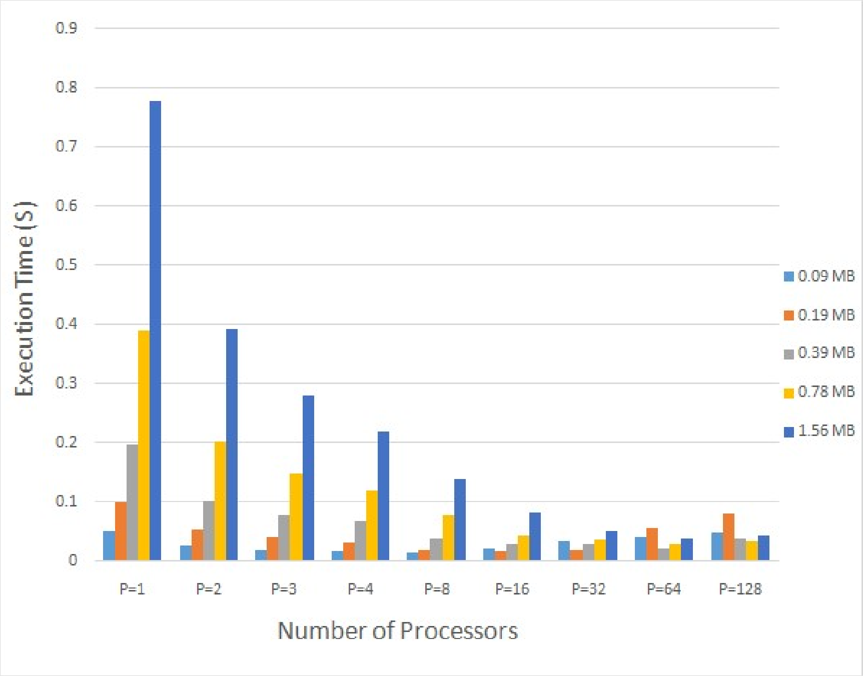


Figure 5. The encryption and decryption time for different number of Processors for small plaintext sizes

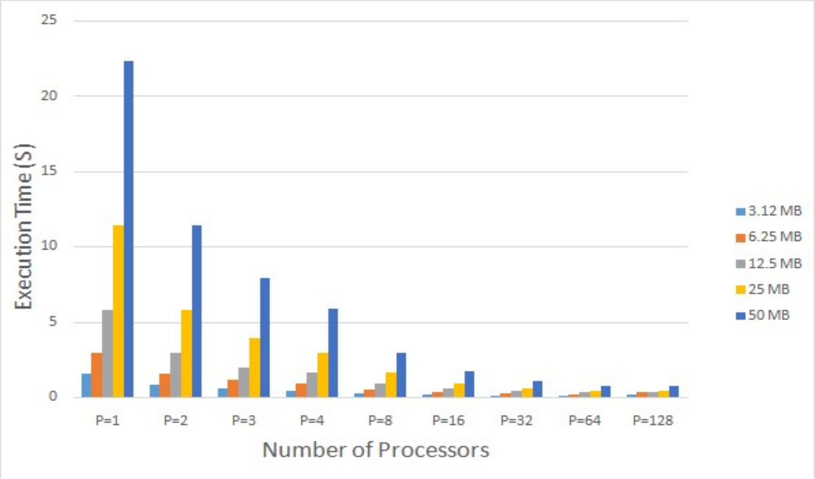


Figure 6. The encryption and decryption time for different number of Processors for large plaintext sizes

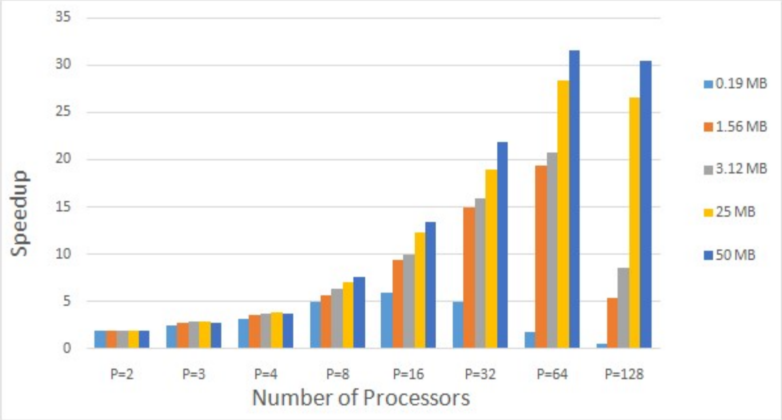


Figure 7. The speedup of the parallel IDEA algorithm on different number of processors and different plaintext size

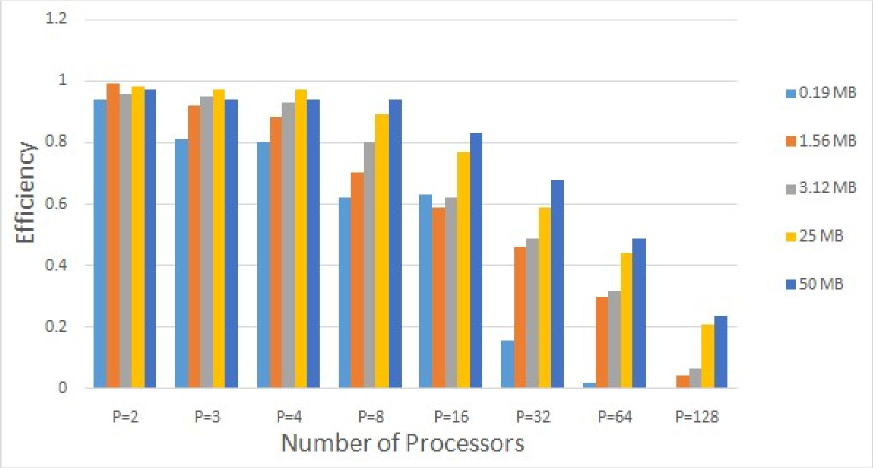


Figure 8. The Efficiency of the parallel IDEA algorithm on different number of processors and Different plaintext size

4. CONCLUSION

In this paper, Performance of parallel IDEA was evaluated according to execution time, speedup and efficiency for different sizes of data and the various number of processors. Parallel IDEA was implemented by C++ using open MPI library and executed on IMAN1 supercomputer. The experimental results show that execution time and speedup of parallel IDEA decreases when the increasing number of processors. When a large number of processors are used to manipulate small data size will increase run time because the amount of communication between processors will be huge. The best value of speedup was achieved when the number of processors equals 64 for data of 50 MByte. Moreover, the best values of the parallel efficiency is when the number of processors is 2, 4, or 8. It achieve up to 99% , 97% , 94%,respectively ,whereas, when number of processors 16 , 32, 64, 128 the parallel efficiency achieve up to 83% ,68% ,49% ,23% respectively.

REFERENCES

[1] Leong, M.P., Cheung, O.Y., Tsoi, K.H. and Leong, P.H.W., 2000. A bit-serial implementation of the international data encryption algorithm IDEA. In Field-Programmable Custom Computing Machines, 2000 IEEE Symposium on (pp. 122-131). IEEE.

[2] Basu, S., 2011. International Data Encryption Algorithm (Idea)–A Typical Illustration. Journal of global research in Computer Science, 2(7), pp.116-118.

[3] Patil, S. and Bhusari, V., 2014. An enhancement in international data encryption algorithm for increasing security. Intl. J. of Application or Innovation in Engineering & Management, 3(8), pp.64-70.

[4] Almasri, O. and Jani, H.M., 2013. Introducing an Encryption Algorithm based on IDEA. International Journal of Science and Research (IJSR), India, 2(9).

[5] Lai, X. and Massey, J.L., 1990, May. A proposal for a new block encryption standard. In Workshop on the Theory and Application of of Cryptographic Techniques (pp. 389-404). Springer, Berlin, Heidelberg.

[6] Beletskyy, V. and Burak, D., 2004, June. Parallelization of the IDEA Algorithm. In International Conference on Computational Science (pp. 635-638). Springer, Berlin, Heidelberg.

[7] Saadeh, M., Saadeh, H. and Qatawneh, M., 2016. Performance Evaluation of Parallel Sorting Algorithms on IMAN1 Supercomputer. International Journal of Advanced Science and Technology, 95, pp.57-72.

[8] Mohammed, Q., 2005. Embedding linear array network into the tree-hypercube network. European Journal of Scientific Research, 10(2), pp.72-76.

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

[10] Qatawneh, M., 2011. 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), p.704.

[11] Qatawneh, M., 2011. Embedding Binary Tree and Bus into Hex-Cell Interconnection Network. Journal of American Science, 7(12), p.0.

[12] Mohammad, Q. and Khattab, H., 2015. New Routing Algorithm for Hex-Cell Network. International Journal of Future Generation Communication and Networking, 8(2), pp.295-306.

[13] Qatawneh, M., 2016. New Efficient Algorithm for Mapping Linear Array into Hex-Cell Network. International Journal of Advanced Science and Technology, 90, pp.9-14.

[14] Mathur, M. and Kesarwani, A., 2013. Comparison between Des, 3des, Rc2, Rc6, Blowfish And Aes. In Proceedings of National Conference on New Horizons in IT-NCNHIT (Vol. 3, pp. 143-148).

[15] Mandal, A.K., Parakash, C. and Tiwari, A., 2012, March. Performance evaluation of cryptographic algorithms: DES and AES. In Electrical, Electronics and Computer Science (SCEECS), 2012 IEEE Students' Conference on (pp. 1-5). IEEE.

[16] Asassfeh, M.R., Qatawneh, M. and AL-Azzeh, F.M., 2018. PERFORMANCE EVALUATION OF BLOWFISH ALGORITHM ON SUPERCOMPUTER IMAN1.International Journal of Computer Networks & Communications (IJCNC), 10(2).

[17] Burak, D., 2015. Parallelization of an encryption algorithm based on a spatiotemporal chaotic system and a chaotic neural network. Procedia Computer Science, 51, pp.2888-2892.

AUTHORS

Ahmad Bany Doumi, is an admitted PhD candidate in Computer Science in University of Jordan, he received his Master degree in computer science from Yarmouk University, his research interests in network security and parallel computing.

Mohammad Qatawneh, is a Professor at computer science department, the University of Jordan. He received his Ph.D. in computer engineering from Kiev University in 1996. Dr. Qatawneh published several papers in the areas of parallel algorithms, networks and embedding systems. His research interests include parallel computing, embedding system, and network security.