Enhanced Data Encryption Algorithm for secured Data in Cloud Computing using Parallel Computing Environment

Nasreen Ahmed Mekki Ismail, Omer Abdel Razag Sharif AbuBaker, and Amin Babiker Abdalnabi Mustafa

Al Neelain University

GCNU Journal

ISSN: 1858-6228

Volume 16 - 2021
Issue: 12

Graduate College
Al Neelain University

Graduate College Journal – NU
Vol.16 -2021, No. (12)
ISSN: 1858-6228, http://www.neelain .edu.sd

مجلة الدراسات العليا - جامعة النيلين
المجلد (16) –2021، العدد (12)
الرقم الدولي الموحد للدوريات: 1858-6228

Enhanced Data Encryption Algorithm for secured Data in Cloud Computing using Parallel Computing Environment

Nasreen Ahmed Mekki Ismail, Omer Abdel Razag Sharif AbuBaker, and Amin Babiker Abdalnabi Mustafa

Faculty of Engineering, Al Neelain University, Khartoum, Sudan.
Corresponding author E-mail: nasreenmekki@yahoo.com

Abstract

Cloud computing environment is a platform gives users in the field of information technology the opportunity for sharing data, information, resources, and services. Sensitive data on cloud needs to be protected from unauthorized users, for this reason security for cloud computing is very important to create secure communications, there are many methods used to protect data, one of them is Data Encryption Standard and Advanced Encryption Standard algorithms. In this paper, a comprehensive investigation on the performance has been compare between parallel computing of Advanced Encryption Standard and Data Encryption Standard and comparison between conventional and parallel computing environment as well as the number of workers (2, 3, and 4 workers). The encryption and decryption time were investigated for different text files sizes using MATLAB platform. The objective of this paper is to enhanced and developed modified encryption algorithm to achieve high level of security for the text files database for the examination office, reduced encryption and decryption time, and speed up the device processor. The obtained results show that the encryption and decryption time of parallel computing of Advanced Encryption Standard is less than the parallel computing of Data Encryption Standard by 52% and 58% respectively, when 4 workers were used for 581 bytes file size, and speed up by 3 times for parallel computing of Advanced Encryption Standard algorithm.

Keywords: DES, AES, Encryption time, Decryption time, parallel computing.

Introduction

Cloud computing is a big platform in the field of information technology for sharing data, resources, services and information by people of the world. This sensitive data on cloud must be secured and hiding from unauthorized user. The big problem in cloud is security, there are many methods used to create secured data in cloud computing, cryptography is one of them. In this paper cryptography is used to create secure data for examination office to protect all text files of exams using symmetric encryption (Marwaha and Bedi, 2013).

Symmetric encryption is a form of cryptosystem that encryption and decryption are performed using the same secret key, which transform plaintext into cipher text for encryption process and the plaintext is recovered from the cipher text for decryption process (Kumari et al., 2017). In this paper a block cipher is used in term of data encryption standard (DES), Advanced Encryption Standard (AES) algorithms.

DES algorithm is most widely used encryption algorithm in the world, this algorithm because it is easy to understand the basic transformations when we follow the steps of the algorithm. DES works on binary number 0's and 1's common to digital computer; it is a symmetric cryptography encryption algorithm which works as a block cipher containing 64 bits, it operates on two inputs; the plain text of a given size 64 bits to be encrypted, and the key and returns cipher text block of same size as output of DES (Sharma, 2017).

AES published by the National Institute of standards and technology (NIST) in 2001, this standard specifies the Rijndael algorithm is a symmetric block cipher of data blocks of 128 bits, using three different keys with lengths of 128, 192, and 256 bits. According to the key length the algorithm referred to as "AES-128", "AES-192", AES-256" (Stallings, 2011).This paper used a block of plain text of 128 bits, and key length of 128 bits.

The important issues for security level is to speed up the encryption time, and decryption time by using parallel computing for DES, AES using MATLAB implementation. To increase security, dependency between the subsequent data block must be introduced for encryption and decryption operation of the parallel computing. In this paper the proposed method encrypted all the files in parallel processing by dividing the file into n-block depend on the file size by uses par for and embed it in a par for loop, parallelization occurs only at the outer level and chooses the number of workers to be used in the parallelization process in MATLAB program (The MathWorks, 2018). The main contribution of this paper is to present the parallelization process of DES, and AES algorithm to speed up measurements of encryption and decryption time for many sizes of encrypted text files and compare between sequential process and parallel computing process execution time for encryption and decryption, then compare between parallel DES, and parallel AES encryption and decryption time.

Various works are made to increase the performance of DES, AES algorithm as it is widely used, and some of them are as follows: Piotr Bilski, et al proposed modified AES algorithm used in any node of the distributed system for the data processing, the efficient approach modification made inside the AES schedule to be able to run on the independent processor core, the operations is divided into separate threads, two approaches are implemented the first one uses time sequences of the particular functions execution and assign the processor cores to these tasks. Finally, the independent rows or columns were put together. The parallelism of the scheme implemented in the lab view, the time analysis of the modified algorithm is more efficient (Bilski and Winiecki, 2010).

Vishal Pachori et al proposed modified AES algorithm, which has two approaches data parallelism and control parallelism, data parallelism is divided into many parts which send to different independent nodes to execute the function or same procedure. The numbers of nodes depend upon the number of processing units; the implementation of this approach is done on Java Parallel Programming Frameworks (JPPF) by passing different data sets to different JPPF nodes. In control parallelism the function or operation divided instead of data, the different operation sends to different nodes, at last the output send to the server, the performance analysis improved the execution time for parallel processing (Pachori et al., 2012).

Piotr Bilski et al proposed symmetric algorithms of multi-core implementation of AES algorithm, the analysis of the operations for the parallelism of whole scheme can be broken into independent parts, so that they can be used inside one program function, there is no need to separate them as a sub procedure, then do the procedure of AES algorithm (Bilski and Winiecki, 2010).

Włodzimierz Bielecki, et al proposed parallelization process of AES algorithm by applied Petit program to find the data dependences in loops and the OpenMP API to present parallelized source code. The OpenMP Application Program Interface support multi-platform shared memory programming in C/C++ and Fortran on architecture OpenMP convert sequential programming language with single program multiple, when the statement with in the parallel region the master threat create a team of threat to execute and verify code on the parallel processing to find the parallelized AES algorithm, the total running time of AES algorithm consist of data reading from input file, data writing to an output file, data encryption, and data decryption (Bielecki and Burak, 2005).

Muhammad Khan, et al proposed a new Fast-DES algorithm, which works in three rounds of block of 32 bits plaintext with key size of 32 bits to produced 32 bits encryption cipher text, this algorithm was design for high speed and provide good security in fast time (Khan et al., 2012) .

Amab Rahman Cow, et al proposed Modified Advanced Encryption Standard a light weight version of AES, which presented a new one dimensional substitution box is proposed by formulating a novel equation for constructing a square matrix of affine transformation, This algorithm has more energy efficiency, less number of packets at the same voltage degradation about 18.35%, and less average transmission time than the original Advanced Encryption Standard (AES), but the algorithm can't use other multimedia data (Chowdhury et al., 2018).

The objective of this paper is to enhanced and developed modified encryption algorithm to achieve high level of security for the text files data base for the examination office, reduced encryption and decryption time, and speed up the device processor.

The paper is organized as follows: Section two briefly review the DES, AES algorithms for parallel computing , section three describes the implementation of DES, AES source code for sequential process and parallel computing. In section four present experimental results and discussion of the application efficiency of a parallelization algorithm of DES and AES, section five shows the conclusion and recommendations.

DES and AES for parallel computing

The proposed method splits the file to be encrypted into n blocks, and then encrypt it by DES algorithm or AES algorithm in series block by block; this process takes time, and lower speed for encryption and decryption process. The research goals are to speed up the algorithm and minimize time of the encryption and decryption process by using DES, and AES algorithm in parallel computing, and compare between two algorithms in sequential form and parallel computing using MATLAB to apply an implementation for DES, and AES algorithms in sequential and parallel computing. To programming an application in parallel, choose the tools of which program used to done this job. In this research MATLAB program was chosen as a platform to provide parallelization process. Parallel computing toolbox provides several high levels programming, which convert your application to take many advantages of computers tools and equipped with multi-core processors. Parallel for-loop (par for) is constructs as special array types for distributed processing to simplify parallel code development by abstracting away the complexity of managing data and computation between the computing resource used and your MATLAB session.

In parallel computing process the same application on different computing resources are running without reprogramming it. The parallel constructs function in the same way, but in which the applications runs on the resource of the computer which sharing the works on a multi-core desktop. For parallel processing you must choose carefully whether you want to convert either the inner or the outer for-loop to par for-loop to avoid parallel overhead.

A par for loop in MATLAB executes a series of statements in loop body in parallel. Each execution of the body of a par for-loop is iteration; MATLAB workers evaluate iterations in no particular order and independently of each other as shown in Figure (1). A par for-loop can provide significantly better performance than for-loop, because many MATLAB workers can compute on the same loop (The MathWorks, 2018). A parallel pool is a set of MATLAB workers on a compute cluster or desktop. Parallel pool starts automatically when the program needed by parallel language feature such as par for, you can choose the parallel preference from the option of menu to display the preference panel to specify the default pool size and cluster in your parallel preferences you can change the pool size and cluster in the parallel menu. Alternatively you can choose cluster and pool size using par cluster and par pool respectively on the MATLAB command as shown in figure (2). The worker in a parallel can used interactively and communicate with each other during the life time of the job (The MathWorks, 2018).

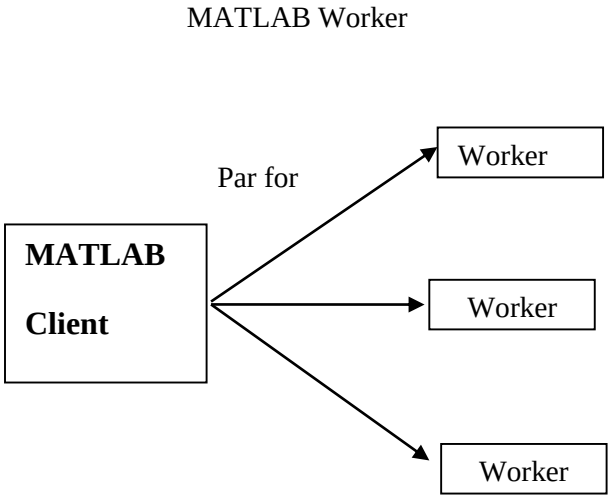

Figure (1) Parallel pool

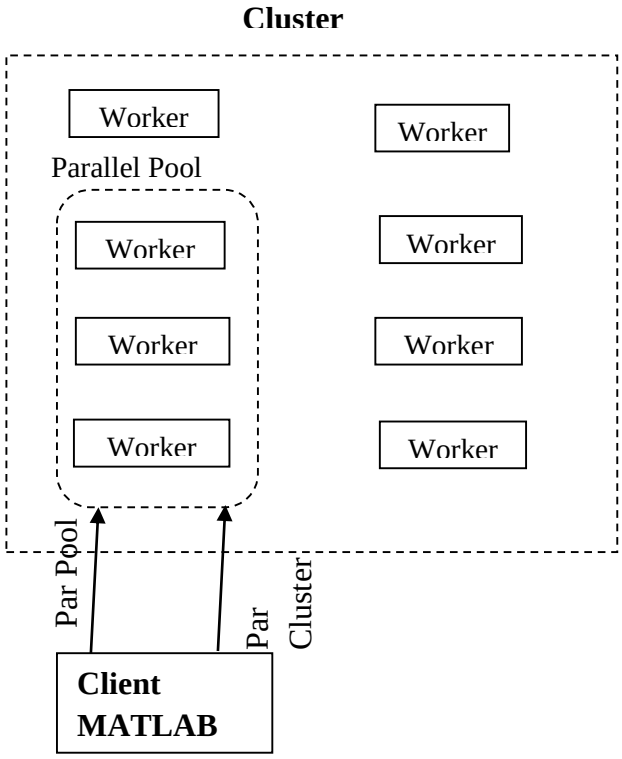

Figure (2) Workers on Cluster

Implementation Scenarios

In this section, the proposed method related to research work is parallel computing DES, AES algorithm, this algorithm is presented for many text files, the data base used in this research is a samples of six text exam files of many size started from size file of 581 bytes up to 3468 bytes, the steps between files size is 581 bytes to find the characteristics of the parallel computing of DES algorithm, and AES algorithm for many selected files size for 2, 3 and 4 workers.

For the proposed method parallel computing process, the implementation of the MATLAB program was done by using the following steps:

1. Reprogram the same application using MATLAB program by converting the outer for-loop to par for loop to avoid parallel overhead.
2. From the option menu choose the parallel preference to display the preference panel to specify the default pool size and cluster in your parallel preference to specify the numbers of workers needed to run the program and take the reading for encryption and decryption time.
3. The results reading are taking for DES and AES algorithms for sequential and parallel computing 2, 3, and 4 workers.

Find the relation for the two parameters, encryption time, and decryption time for the following relations:

- Comparison between DES/AES algorithm encryption and decryption time for sequential and parallel computing for 2, 3 and 4 workers for many files.

- Comparison between the proposed method parallel DES algorithm, and parallel AES algorithm for 2, 3 and 4 workers for encryption, and decryption times.

- Speed up for DES/AES sequential to parallel computing for 2, 3, and 4 workers.

Results and Discussions

DES Algorithm

DES Algorithm Encryption Time for Sequential and Parallel Computing

Table (1) DES Encryption Time for Sequential and Parallel Computing

Process			**Encryption Time (sec.)**					
	File size (bytes)		**581**	**1164**	**1747**	**2320**	**2895**	**3468**
	No. Workers							
Sequential (S)	**S**	-	17.8786	32.8210	49.2639	65.1550	82.0866	97.8809
Parallel Computing(P), Speed up = S/P	**P2**	**2**	10.4331	17.8018	25.6769	33.7526	42.0650	50.6055
	S/P2	**2**	1.71	1.84	1.91	1.93	1.95	1.93
	P3	**3**	7.3349	12.7840	18.7023	24.5747	30.5070	36.4165
	S/P3	**3**	2.43	2.56	2.63	2.65	2.69	2.68
	P4	**4**	5.9400	10.7320	15.6096	21.0298	24.1022	29.8808
	S/P4	**4**	3	3	3.1	3.1	3.4	3.3

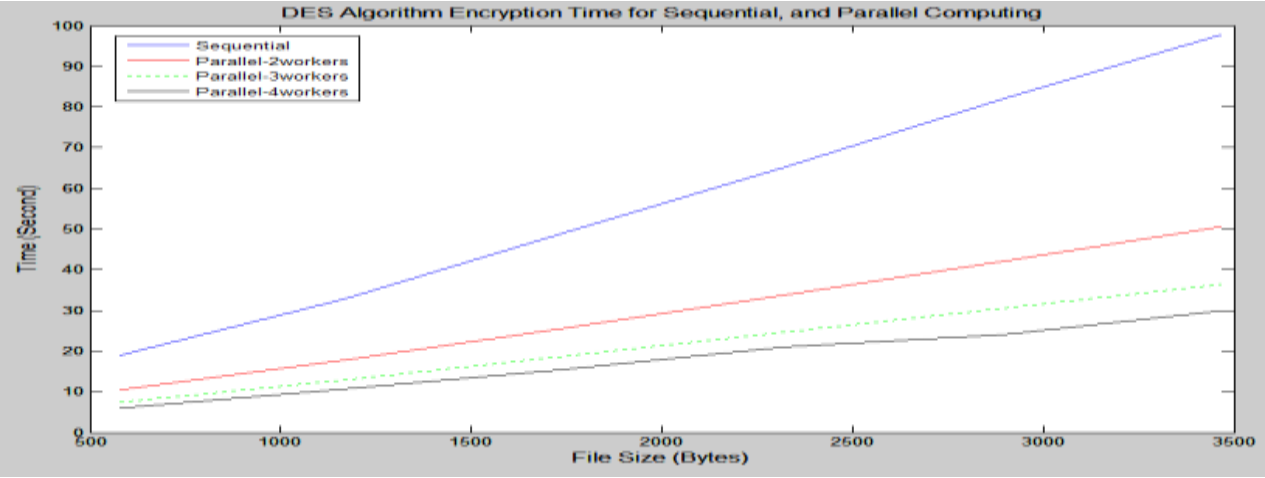

Figure (3) DES Encryption Time for Sequential and Parallel Computing

DES Algorithm Decryption Time for Sequential and Parallel Computing

Speed up = (Time taken by the serial algorithm) / (Time taken by the parallelism algorithm) (Pachori et al., 2012).

Table (2) DES Decryption Time for Sequential and Parallel Computing

Process			Decryption Time (sec.)					
	File size (bytes)		581	1164	1747	2320	2895	3468
	No. Workers							
Sequential(S)	S		16.8204	33.3861	49.9870	66.3303	83.8460	99.5888
Parallel Computing(P), Speed up = S/P	P2	2	9.2213	17.7451	26.2096	34.3891	43.1065	51.8102
	S/P2	2	1.82	1.88	1.9	1.92	1.94	1.92
	P3	3	7.0553	12.8824	19.0132	25.1822	31.5118	37.5862
	S/P3	3	2.38	2.59	2.62	2.63	2.66	2.65
	P4	4	5.5978	10.6496	16.0486	21.2526	25.5930	32.1292
	S/P4	4	3	3.1	3.1	3.1	3.3	3.1

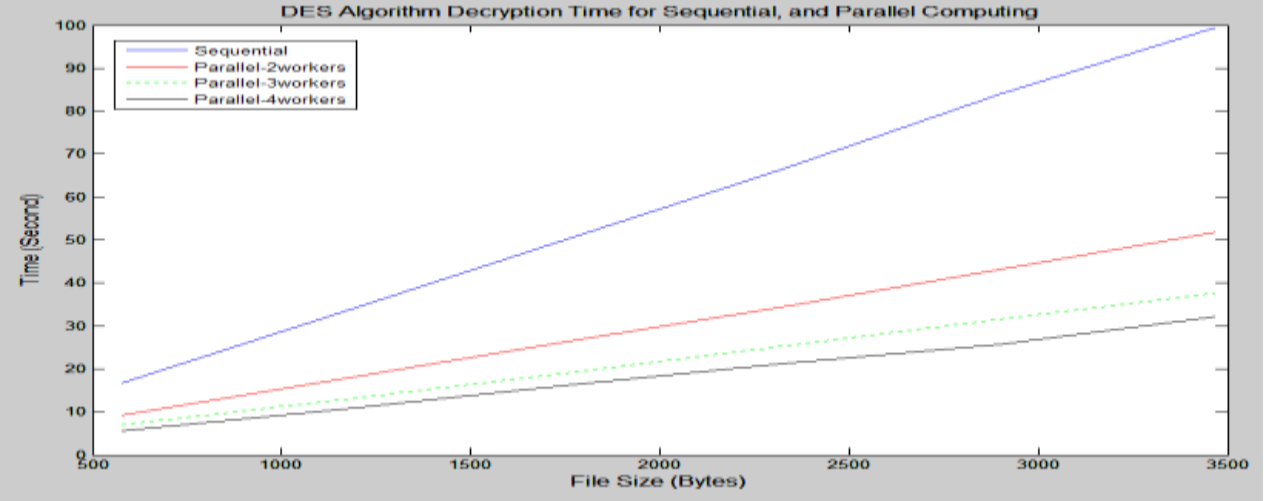

Figure (4) DES Decryption Time for Sequential and Parallel Computing

AES Algorithm

AES Algorithm Encryption Time for Sequential and Parallel Computing

Table (3) AES Algorithm Encryption Time for Sequential and Parallel Computing

Process			Encryption Time (sec.)					
	File size (bytes)		581	1164	1747	2320	2895	3468
	No. Workers							
Sequential (S)	S	-	6.9305	13.4143	20.3356	26.9096	33.6032	40.1093
Parallel Computing (P), Speed up = S/P	P2	2	3.9015	6.8539	9.9448	13.0475	16.0184	18.8529
	S/P2	2	1.77	1.96	2.0	2.1	2.1	2.1
	P3	3	3.0506	5.0942	7.4896	9.3051	11.4438	13.5786
	S/P3	3	2.3	2.6	2.7	2.9	2.9	2.9
	P4	4	2.8245	4.4393	7.0510	7.9035	9.8988	11.3264
	S/P4	4	2.5	3.0	2.88	3.4	3.4	3.5

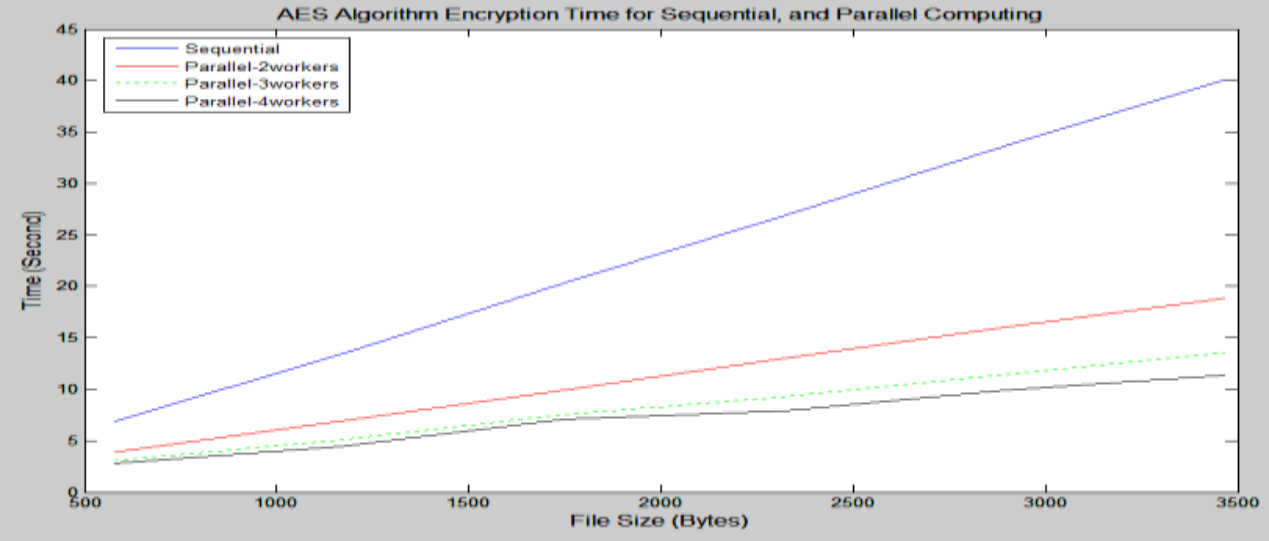

Figure (5) AES Encryption Time for Sequential and Parallel Computing

AES Algorithm Decryption Time for Sequential and Parallel Computing

Table (4) AES Algorithm Decryption Time for Sequential and Parallel Computing

Process			Decryption Time (sec.)					
	File size (bytes)		581	1164	1747	2320	2895	3468
	No. Workers							
Sequential (S)	S	-	7.1464	14.1357	21.1728	32.5046	35.1032	41.8212
Parallel Computing (P), Speed up = S/P	P2	2	3.6262	6.7347	10.0234	13.2320	16.3179	19.1848
	S/P2	2	1.97	2.1	2.1	2.4	2.1	2.1
	P3	3	2.6857	4.8479	7.2309	9.2929	11.4290	13.6670
	S/P3	3	2.7	2.9	2.9	3.5	3.1	3.1
	P4	4	2.3574	4.1080	6.8827	7.6571	9.7151	11.3911
	S/P4	4	3.0	3.4	3.1	4.2	3.6	3.6

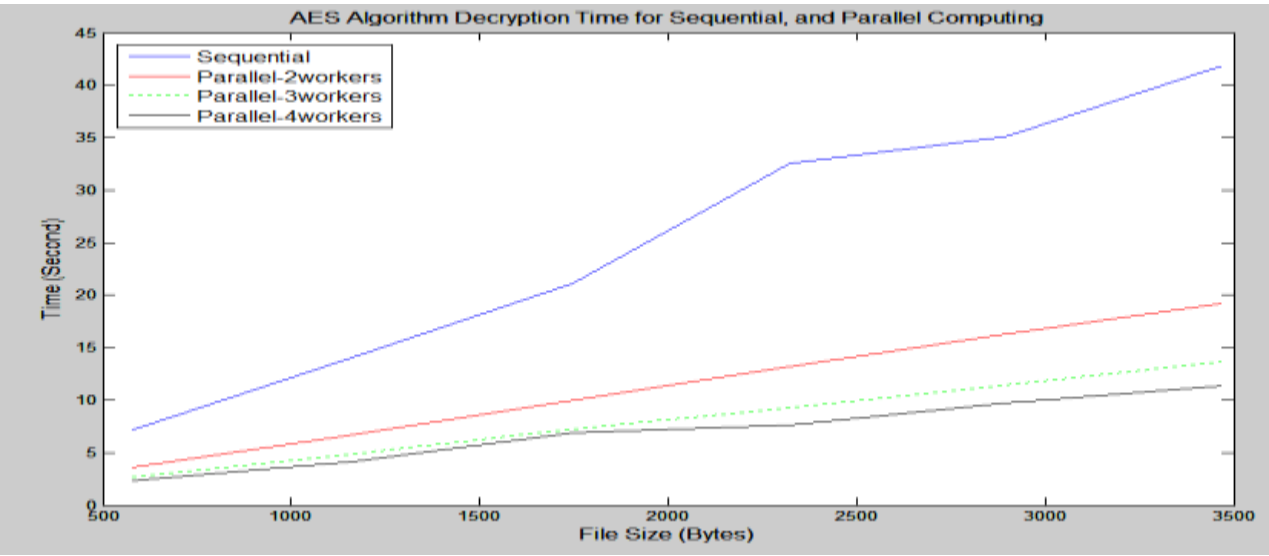

Figure (6) AES Decryption Time for Sequential and Parallel Computing

From the result tables (1), (2), (3), (4) for DES, AES Encryption and decryption time for sequential and parallel computing' when the file size is increase the encryption and decryption time are increase for sequential and parallel computing workers 2,3,4. For parallel computing when the number of workers are increase the encryption and decryption time are decrease for all files size as shown in figures (3), (4), (5), (6), because the data was sharing to more workers using par-for loop, which compute on the same loop to provide better performance, and less encryption, and decryption time, the relation between file size and encryption, and decryption time is linear as shown in figures (3), (4), (5), (6).

The highest speed up DES/AES parallel computing was shown in tables (1), (2), (3), (4) for 4 workers.

Comparison between the proposed method parallel DES/AES algorithm

Comparison between Parallel DES/AES Algorithm 2 workers encryption and decryption time

Table (5) Comparison between Parallel DES/AES Algorithm 2 workers encryption and decryption time

Process		**Encryption Time**					
	File Size (bytes)	581	1164	1747	2320	2895	3468
	No. workers						
Parallel DES (P1)	**2**	10.4331	17.8018	25.6769	33.7526	42.0650	50.6055
Parallel AES (P2)	**2**	3.9015	6.8539	9.9448	13.0475	16.0184	18.8529
[(P1-P2)/P1]*100%		62%	62%	**61%**	**61%**	62%	62%
		Decryption Time					
Parallel DES (P3)	**2**	9.2213	17.7451	26.2096	34.3891	43.1065	51.8102
Parallel AES (P4)	**2**	3.6262	6.7347	10.0234	13.2320	16.3179	19.1848
[(P3-P4)/P3]*100%		61%	62%	62%	62%	62%	62%

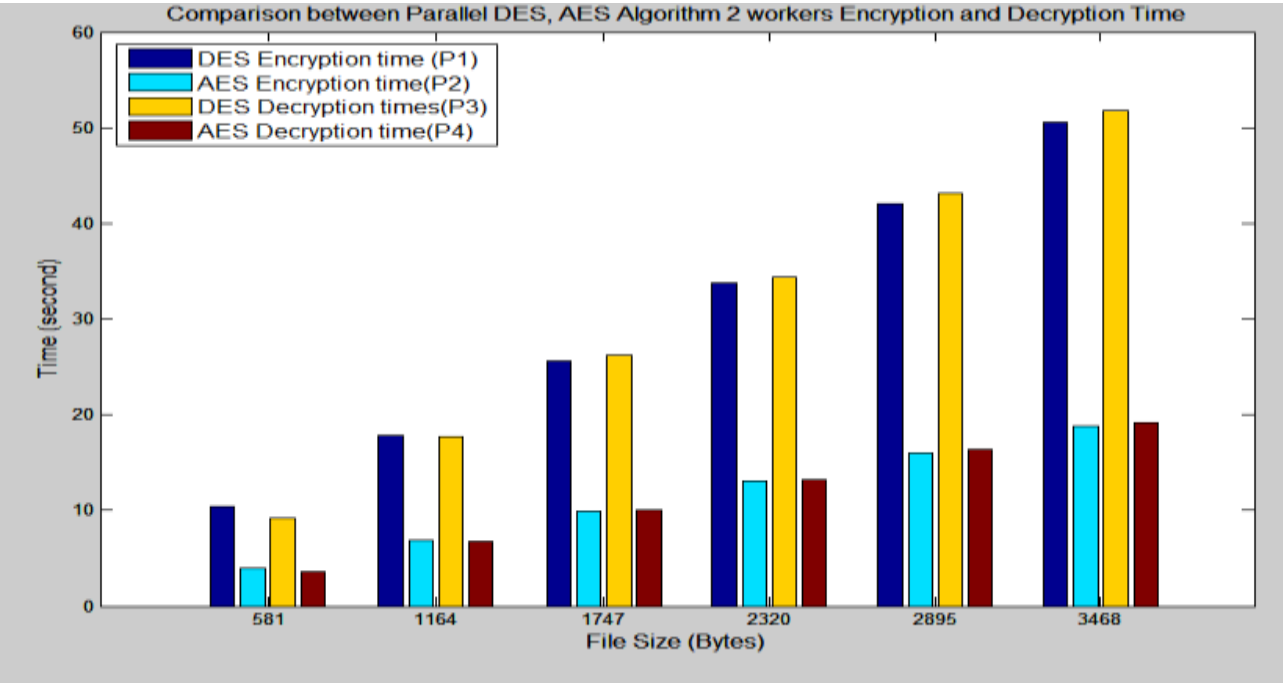

Figure (7) Comparison between parallel DES, AES Algorithm 2-workers encryption and decryption time

Comparison between Parallel DES/AES Algorithm 3 workers encryption and decryption time

Table (6) Comparison between Parallel DES/AES Algorithm 3 workers encryption and decryption time

Process			**Encryption Time**				
	File Size (bytes)	**581**	**1164**	**1747**	**2320**	**2895**	**3468**
	No. workers						
Parallel DESP1	**3**	7.3349	12.7840	18.7023	24.5747	30.5070	36.4165
Parallel AESP2	**3**	3.0506	5.0942	7.4896	9.3051	11.4438	13.5786
[(P1-P2)/P1]*100%		58%	60%	60%	62%	62%	62%
		Decryption Time					
Parallel DES (P3)	**3**	7.0553	12.8824	19.0132	25.1822	31.5118	37.5862
Parallel AES (P4)	**3**	2.6857	4.8479	7.2309	9.2929	11.4290	13.6670
[(P3-P4)/P3]*100%		62%	62%	62%	63%	63%	63%

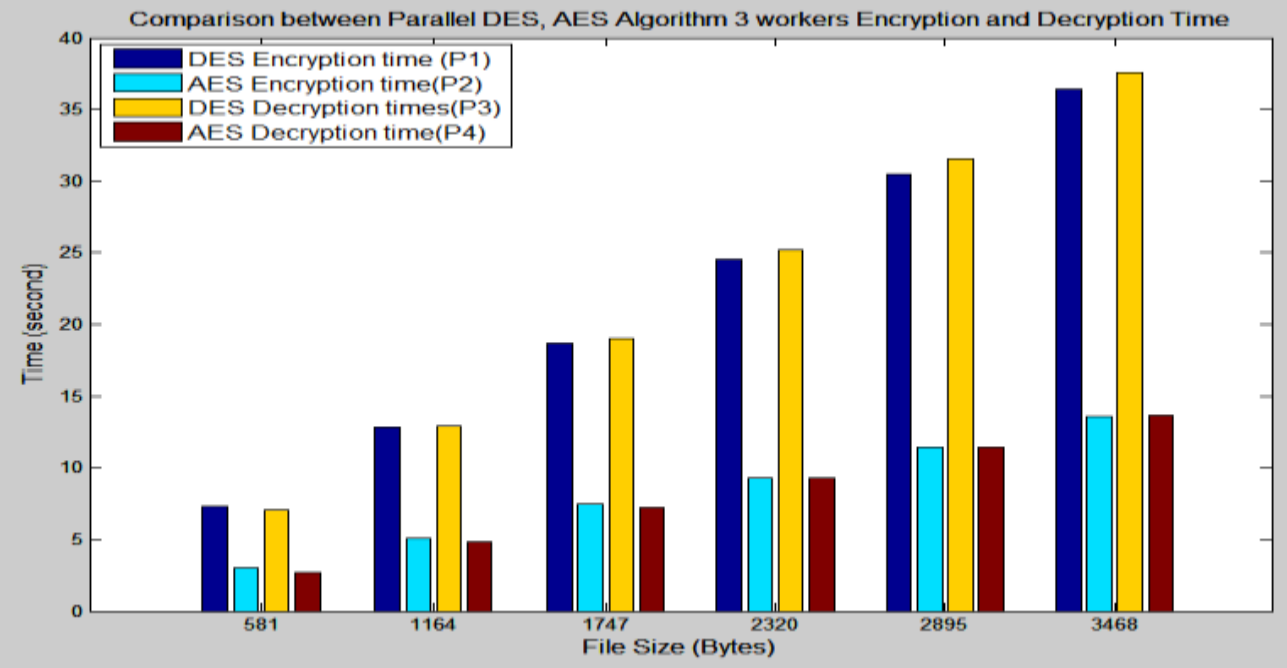

Figure (8) Comparison between parallel DES/ AES Algorithm 3-workers encryption and decryption time

Comparison between Parallel DES Algorithm 4 workers, and Parallel AES Algorithm 4workers Encryption Time

Table (7) Comparison between Parallel DES/AES Algorithm 4 workers encryption and decryption time

Process		Encryption Time (sec.)					
	File Size (bytes)	581	1164	1747	2320	2895	3468
	No. workers						
Parallel DES(P1)	4	5.9400	10.7320	15.6096	21.0298	24.1022	29.8808
Parallel AES(P2)	4	2.8245	4.4393	7.0510	7.9035	9.8988	11.3264
[(P1-P2)/P1]*100%		52%	58%	55%	62%	59%	62%
		Decryption Time (Sec.)					
Parallel DES(P3)	4	5.5978	10.6496	16.0486	21.2526	25.5930	32.1292
Parallel AES(P4)	4	2.3574	4.1080	6.8827	7.6571	9.7151	11.3911
[(P3-P4)/P3]*100%		58%	61%	57%	64%	62%	64%

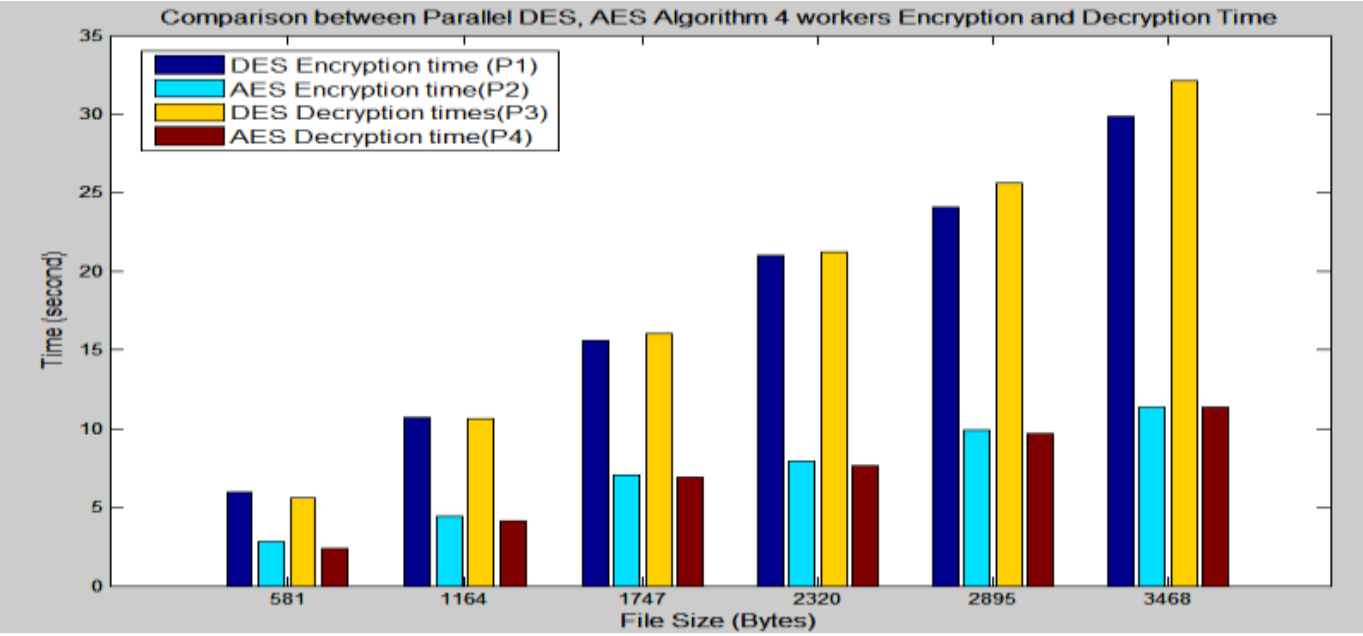

Figure (9) Comparison between Parallel DES/AES Algorithm 4 workers encryption and decryption time

From the comparisons on tables (5), (6), (7) between parallel DES/AES, the encryption and decryption time of parallel AES (P2),(P4) is less than parallel DES (P1), (P3) according to the results in tables (5), (6), (7), This result refers to the mechanisms of AES algorithm works, the numbers of digits used and the numbers of rounds used.

Conclusions and Recommendations

The modified version of DES and AES algorithm is successfully implemented in MATLAB programming and performance is measured in terms of execution of encryption and decryption time using parallel computing of 2, 3, and 4 workers. Parallel AES has low encryption and decryption time (2.8245, 2.3574 seconds) respectively for file size 518 bytes and highest speedup of 3 times shown in parallel computing process using 4-workers. For the Future work other cryptography algorithm can also be implemented using parallel computing in order to increase their performance or using other fast framework available, and different media files as well as other encryption algorithms can be investigated.

References

BIELECKI, W. & BURAK, D. Parallelization of the AES Algorithm. Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, 2005. 224-228.

BILSKI, P. & WINIECKI, W. 2010. Multi-core implementation of the symmetric cryptography algorithms in the measurement system. *Measurement*, 43**,** 1049-1060.

CHOWDHURY, A. R., MAHMUD, J., KAMAL, A. R. M. & HAMID, M. A. MAES: modified advanced encryption standard for resource constraint

environments. 2018 IEEE Sensors Applications Symposium (SAS), 2018. IEEE, 1-6.

KHAN, M. N., WAHID, I. & IKRAM, A. A. 2012. The FastDES: A New Look of Data Encryption Standard. *International Journal of Computer Applications,* 975**,** 8887.

KUMARI, P., BALA, M. & SHARMA, A. 2017. A Comparative Study of Symmetric Key Algorithm DES AES and Blowfish for Video Encryption and Decryption. *International Journal of Advance Engineering and Research Development,* 4.

MARWAHA, M. & BEDI, R. 2013. Applying encryption algorithm for data security and privacy in cloud computing. *International Journal of Computer Science Issues (IJCSI),* 10**,** 367.

PACHORI, V., ANSARI, G. & CHAUDHARY, N. 2012. Improved performance of advance encryption standard using parallel computing. *International Journal of Engineering Research and Applications (IJERA),* 2**,** 967-971.

SHARMA, N. 2017. A Review of Information Security using Cryptography Technique. *International Journal of Advanced Research in Computer Science,* 8.

STALLINGS, W. 2011. *Cryptography and Network Security Principle and Practice*

THE MATHWORKS, I. 2018. *Parallel Computing |Toolbox* [Online]. USA. [Accessed june 2018].