Uniport Journal of Engineering and Scientific Research (UJESR)

Vol. 5, Special Issue, 2020, Page 1-10
ISSN: 2616-1192

(www.ujesr.org)

Simulation- Based Comparative Analysis of Cryptographic Algorithms

Omijeh, B.O*[1] and Agughalam, D.M[2]
[1,2]Centre for Information and Telecommunication Engineering, University of Port Harcourt, Port Harcourt.
*Corresponding author's email: omijehb@yahoo.com

Abstract
In this paper, the simulation-based comparative analysis of Cryptographic Algorithms was achieved. The need to protect messages passed over the internet is high. One of the ways of achieving this is cryptography. This research takes a look at some of the algorithms facilitating this method of securing information. A simulation-based comparative analysis was carried out on symmetric algorithms (Advanced Encryption Standard (AES), Triple Data Encryption Standard (3DES) and Blowfish) and asymmetric algorithms (Rivest-Shamir-Adleman (RSA) and Diffie-Hellman) in terms of encryption and decryption times. This was achieved by building a simulation environment on the C#.NET programming language using the Microsoft visual studio 2010 application for implementing the cryptography algorithms. It was found that for smaller file sizes, the 3DES algorithm was fastest in terms of encryption and decryption compared to the AES and blowfish algorithm. For larger file sizes, the AES algorithm was the most efficient in terms of encryption and decryption times. Comparing the asymmetric encryption techniques, it was found that the Diffie-hellman algorithm encrypted faster but decrypted slower than the RSA algorithm.

Keywords: *Cryptography, Encryption, Symmetric Algorithms, Asymmetric Algorithms*

Received: 14th March, 2020 *Accepted: 3rd April, 2020*

1. Introduction

In recent times, the need for computer systems and computer based applications in all works of life cannot be overemphasized. These include banking applications, online shopping, and other forms communication over wired or wireless networks which help make daily life easier. As sensitive information is sent over the communication medium, there is need to protect the information being sent to avoid falling into the hands of unwanted parties owing to the increase in hacking attempts and the ongoing cyber warfare. This brings about the need for information and data security. Information security entails protecting the confidentiality, Integrity and availability of the data and the communication channel. These are regarded to as the CIA of information security and can be achieved by encryption, hashing and the use of firewalls respectively.

As is mostly the case, computer based applications are built based on the comfort of the user without acknowledging so much of the security aspect till the application is done. Then implementing the security requires trading off some of those comfort features introduced while building the application. These could include extra processing and execution time for simple commands in the application due to the need to encrypt and decrypt information being sent by the system. A glaring example of this is in the use of banking application. The user is always asked to provide some sort of validation to ensure that the user is authorized.

One major way of ensuring confidentiality of data in networks is through cryptography (encryption). In more technological terms, cryptography which involves using specific alphanumeric characters called keys for encrypting data to ensure security and this is an adept way of achieving information security. Encryption is a process of scrambling a message so that only the intended recipient can read it. Over the years, there have been several algorithms created for encryption of data. All these algorithms, while having the same goal, function differently and vary in potency of security, needed resources, the time taken to encrypt and further decrypt the data amongst other factors. These encryption algorithm are generally classified into symmetric and asymmetric algorithms due to the way they function.

In the use of symmetric algorithms for encryption, the same key is used for encryption and decryption of data. The key is hidden and transmitted in a secure form to the intended receiver and is usually encrypted with another key for further protection as interception of the key can lead to quick breach of the message. They can be further divided into stream ciphers and block ciphers. Stream ciphers generate a key stream and parse through the data encrypting a single bit at a time while block ciphers group a block of bits together and encrypt them as a single unit. Block ciphers make use of initialization vectors due to the fact that some chunks of the data might be the same and without the initialization vector, the cipher text would be the same. This further randomization of the encryption process makes block ciphers more efficient for encrypting larger amount of data than stream ciphers.

The symmetric algorithms considered in this paper are listed and described. ***DES:*** DES which stands for Digital Encryption standard, was developed in 1975 with contribution from the National Security Agency (NSA) and was the first algorithm to be recommended by the National institute of Standards and technology (NIST). DES is a block cipher of 64-bit key length (56-bit input key and 8 parity bits) and block size. The 56-bit key is used to generate a key table of 16 sub keys of 48-bit length used to cipher the plain text. The DES cipher runs 16 times with each of the keys encrypting the cipher text to totally encrypt the data and the decryption is the reverse of the encryption process using the same keys. One major advantage of the DES algorithm is its flexibility functioning in CBC, ECB, OFB AND CFB modes. DES was later found out to be susceptible to brute force attack by cryptanalysts and its short key length was a limitation for use in encrypting large amount of data. ***3DES*****:** After the weakness in the DES algorithm was exposed, the algorithm was revised by making the algorithm run 3 more times on the plain text hence the name triple DES. The new algorithm makes use of longer key lengths, 112-bits, 168-bits and 192-bits being the most commonly used. The triple DES algorithm improved upon the security offered by DES. The longer key length also improves its ability to cipher larger amounts of data. ***AES:*** AES which stands for Advanced Encryption Standard is the new standard of encryption declared by the NIST after the failure of the DES algorithm. This algorithm was developed in a competition organized by the NIST won by Vincent Rijmen and Joan Daemen with an algorithm called "Rijndael". It has varying key lengths of 128, 192 or 256-bits of which 256-bits is the default key length. AES achieves encryption on 128 bits data block which can be subdivided into 4 operational blocks by going through 10, 12 and 14 rounds for the 128, 192 and 256 bit keys respectively. The AES algorithm has been extensively tested and found to be perfect for both software and hardware implementations especially and smaller devices cutting across a number of powerful processors. ***Blowfish:*** Blowfish is a block cipher algorithm with 64 bit blocks and variable keys of 32-bit to 448-bit key length. Blowfish is the most commonly used domain name encryption algorithm as it is open source and free to use. No known attacks have been successful against blowfish though the keys might be considered weak due to the possible short key lengths.

Asymmetric algorithms were developed mainly to solve one of the major problems of symmetric algorithms which is the need to share the key as the same key used for encryption must be used for decryption. Asymmetric encryption aims to solve this problem. Asymmetric encryption involves the use of a key pair that are mathematically linked to each other known as the public key and the private key to encrypt and decrypt the data. The public key can be shared with anybody on the cryptosystem while the private key is kept private and used by the intended receiver to decrypt the data. Some popular asymmetric algorithms considered in this research are listed and described. ***RSA:*** RSA which stands for the last name of the inventors of the algorithm is an asymmetric encryption algorithm making use of a key pair created in 1978. RSA leverages off prime numbers to create this key pair but is impractical for use in encrypting large amounts of data due to the time it would take the computer to process. The algorithm makes use of variable key length and is a block cipher. The main advantage of the RSA algorithm is its convenience and enhanced security. ***Diffie-Hellman:*** This algorithm, introduced in 1976 commonly known as DH, stands for the last name of the two inventors Whit Diffie and Martin Hellman with contribution form Ralph Merkle. The DH algorithm is not really an encryption algorithm but a key exchange algorithm. The inventors wanted to create a secure medium of sharing the public keys over an insecure communication channel. The communicating parties share a secret number and a session key generated by the algorithm for decrypting and encrypting the message.

Some studies on algorithms had been conducted. Elminaam et al. (2008) conducted a research to test six encryption algorithms AES

(Rijndael), DES, 3DES, RC2, Blowfish, and RC6 against each other which different settings such as varying data block size, different data type(text or audio), power consumption, key size and encryption and decryption speed. These factors are collated by collecting various measurement metrics on the implementation device such as CPU process time and CPU clock cycles and battery power. After experimentation and simulation, they found out that Blowfish had the highest throughput as per encryption and decryption time followed by the RC6 algorithm. It was also found out that the key size has an effect of increasing the encryption and decryption time for the AES and RC6 algorithms supporting variable key lengths. The DES algorithm was also found to perform lower than the 3DES algorithm. Ramdeo (2012) undertook an evaluation of four data encryption algorithms, DES, 3DES, Blowfish and AES. The experimentation was based on the standard implementation provided by the .NET environment. The results showed that Blowfish had the highest throughput in encryption and decryption amongst all algorithms tested with 3DES showing the least performance. In ECB mode of analysis, Blowfish also had the least processing time and AES consumed more resources when the block size is relatively large. 3DES consuming more processing time than DES due to its triple rounds. Finally, Blowfish and AES were said to have no wormholes in security but DES and 3DES had. Ramesh and Umarani (2012) undertook a study to compare six symmetric encryption algorithms RC6, UMARAM, DES, 3DES, RC2 AND UR5 in terms of encryption speed depending on file size, operating system and data type. These algorithms were implemented on Microsoft Visual basic 6.0 in their standard format which they used to build the front end tool. They found out that all the algorithms run faster on Windows XP operating system compared to Windows Vista and Windows 7 but UR5 was the most efficient and UMARAM runs slower than DES and 3DES for text data. They also concluded that UR5 encrypts image data best across all three platforms. Ebrahim et al. (2013) carried out a research aimed at comparing different symmetric encryption algorithms such as AES, DES, 3DES, MARS, IDEA, Serpent, Blowfish and Twofish based on their architecture, scalability, reliability, flexibility, security and limitations. After an exhaustive research on these algorithms based on the aforementioned factors, they found AES to be the fastest and most secure among symmetric encryption algorithms with no alarming weaknesses but small flaws such as weak keys and insecure medium for transmission of keys. They also found DES to be really slow for software implementation compared to implementation in hardware. IDEA has a problem of large key classes enabling cryptanalysts to be able to reverse engineer the key thereby cracking the algorithm. Triple DES was found to be the slowest but safest compared to DES and IDEA but still posing the same problem in software implementation as DES. SERPENT was slow to analyze and complex as were Twofish and MARS.

Research into cryptography algorithms was also carried out by Gupta and Walia (2014) who proposed a review on algorithms such as DES, 3DES, Blowfish, AES, IDEA and RSA. They carried out an analysis on their ability to secure data, key size, and block size. After extensive research, they concluded that the blowfish algorithm was superior to other symmetric encryption algorithms in terms of power consumption, better performance and efficiency. They also found that RSA was the most widely used asymmetric algorithm and has easy implementation with other algorithms especially for key encryption to improve security of symmetric algorithms. In their work in 2015, Ayyappadas et al. carried out a research on six symmetric encryption algorithms DES, 3DES, AES, RC2, Blowfish and RC6 to check their effect of the power consumption for wireless devices that run on battery power. They effected a comparative analysis based on data type, data size and decryption speed. They found out that Blowfish was the most power efficient, least processing time and maximum throughput compared to the other algorithms with RC2 the least performer. Koko and Mustapha (2015) carried out a comparison on various encryption algorithms to improve secure data communication. They considered AES with varying keys, DES, 3DES, RC4 and Blowfish. They found the RC4 algorithm to be fastest amongst the tested algorithms but the least secure as it had various weaknesses recorded. They also labelled DES and Triple DES as slow and very slow respectively with both being unsafe and crackable. Bisht and Singh (2015) in their research paper, carried out an analysis between AES, DES, RSA and the DIFFIE-HELLMAN algorithms for cryptography based on key size, encryption and decryption times, cost, ease of implementation, power consumption and security. They found out that the symmetric encryption techniques were better in terms of speed and power consumption but the asymmetric ones were better in terms of

tunability. They also found AES to be better in cost and ease of implementation among the symmetric key algorithms while RSA best in security and speed amongst the asymmetric algorithms. Awotunde et al. (2016) studied four encryption algorithms Blowfish, DES, 3DES and AES for viability, reliability and performance levels. They implemented these algorithms on the JAVA programming language which already has built in encryption APIs such as java crypto and java safety keeping factors such as operation environment and file size constant across the algorithms. They leveraged on factors such as encryption speed, memory consumption and CPU utilization to determine the viability and reliability of each of the algorithms. They found out that 3DES had the least CPU utilization time, with DES having the fastest encryption speed and memory utilization amongst all algorithms tested. In the aspect of performance, Blowfish was found to outperform others as they had wormholes in their security systems. In 2016, Koukou et al. carried out a comparative analysis on AES, Blowfish, CAST-128 and the DES encryption algorithms. They were most interested on the effect of different load sizes on the algorithms using parameters such as speed, block size and key size. They also tested the avalanche effect of these algorithms in ECB and CBC mode. After experimentation, they found out that AES offers the best security with Blowfish also having a good security level higher than both CAST-128 and DES with DES being the least secure. DES showed the highest avalanche effect in both modes tested. Maqsood et al. (2017) carried out a comparative analysis on modern cryptography techniques such as DES, AES, RSA and ELGamal depending on their encryption and decryption times, file size and key generation time. They implemented the algorithms on the Java platform on an Intel Pentium processor of 2.34GHz and 1gigabyte Ram size. After carrying out the experiment, they found out that DES had the smallest amount of time for key generation, followed by ElGamal, AES and finally RSA. In the aspect of encryption time, AES has the least encryption and decryption time compared to the other algorithms tested alongside for all the file sizes.

In this paper, a simulation-based comparative analysis was carried out on symmetric algorithms (AES, 3DES and Blowfish) and asymmetric algorithms (RSA and Diffie-Hellman) in terms of encryption and decryption times to enhance better system performance. Also, a graphical user interface application was developed using C# programming Language.

2. Materials and methods

The implementation of this research was done with the C# programming language using Microsoft Visual Studio 2010 on a computer system with a core i3, 64 bit processor and 6 gigabytes of RAM. The Application Programming Interface for implementing each of the algorithms was developed using C# programming Language. This simulation software is built to be able to simulate encryption and decryption of symmetric cryptographic algorithms such as the AES algorithm, 3DES and the blowfish algorithm as well as asymmetric algorithms such as the Diffie-Hellman key exchange algorithm and the Rivest-Shamir-Adleman algorithm (RSA). The software measures the encryption and decryption times of each of these algorithms hence is an efficient tool for carrying out the proposed comparative analysis. The software is validated as all the libraries used are already present in the C# programming language which is a standard and accepted programming language worldwide. In the aspect of testing, the simulation software is only constructively tested to make sure it satisfies the requirements for which it is built. Destructive testing will however be ignored as it is not meant for commercial use yet. For the steganography, an external open source software is redesigned and used to achieve the aims of this research and used in the analysis.

The parameters considered in this design are:

Encryption Speed: As earlier defined, encryption speed of an algorithm is the time taken by the algorithm to convert plaintext to cipher text. The decryption speed is the time taken to convert the cipher text back to plaintext. This parameter is essential to any encryption algorithm due to time being an important factor in the applications for which they are deployed. This time is gotten by initiating an instance of the stopwatch class just before the encryption or decryption and stopping it right after. The result from this is displayed in milliseconds.

Key Size: For algorithms with variable key sizes, the size of the key affects the encryption time as the key is directly used to process each of the blocks converting plaintext to cipher text in both symmetric and asymmetric algorithms. Therefore varying key sizes will also produce varying encryption and decryption times. The Key sizing for the algorithms was left as default as most applications employ them as default except otherwise selected.

Block Modes: The cipher block chaining mode was used for the encryption algorithms as it provides

better security by using initialization vectors for encryption of each block thereby hiding patterns in large data sets that may be containing similar blocks. The simulator Flow Chart is shown in Fig.1 and the GUI for the AES Encryption is shown in Fig. 2.

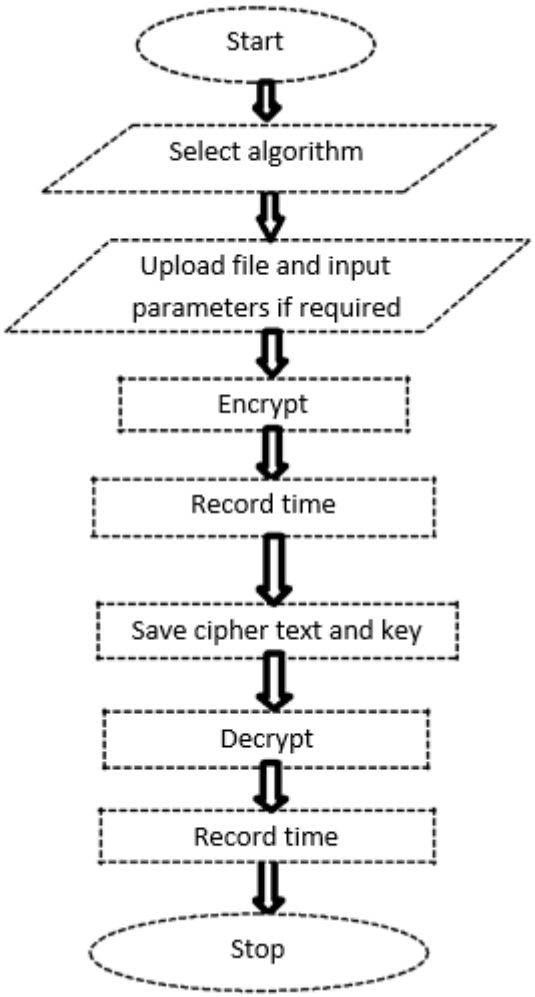

Fig. 1: Simulator flow chart

The files used to test these algorithms were divided into three. Smaller sized files depicting voice packets, data packets, machine to machine commands, ranging from 1 kilobyte to 5 kilobytes, larger file sizes ranging from 5 megabytes to 25 megabytes simulating bigger file transfers such as document transfer applications over the web and key files used to test the asymmetric algorithms. The file sizes were spread this way to imitate real practices and to compare the effect of the encryption algorithm on smaller file sizes and the effect on larger file sizes in terms of encryption and decryption time. The simulation is conducted repeatedly to make sure results conform for valid comparative analysis. The files are uploaded one after the other for each encryption algorithm, encrypted and the time taken for encryption recorded, further decrypted and time for decryption is also recorded. For the symmetric encryption algorithms, this is done 5 times for each file size and algorithm and the average of the results taken. Care is also taken to reset the algorithm after every use so as to start the encryption process from the very beginning to avoid using already made keys and tampering with expected results. The simulation here is done in two parts, first the smaller file sizes and the larger file sizes. For the smaller file sizes, all 3 symmetric encryption algorithms are tested while for the larger file sizes, only the AES and the Triple DES algorithms are tested. The blowfish algorithm is in practice, not normally used for encryption of large data due to its large key sizes and computational burden. As this research aims to answer questions involving real practices, only the AES and the triple DES algorithms are tested for larger file sizes. For the asymmetric encryption algorithms, the files with the keys are tested as they are not used to encrypt data but to encrypt symmetric keys used in the symmetric encryption.

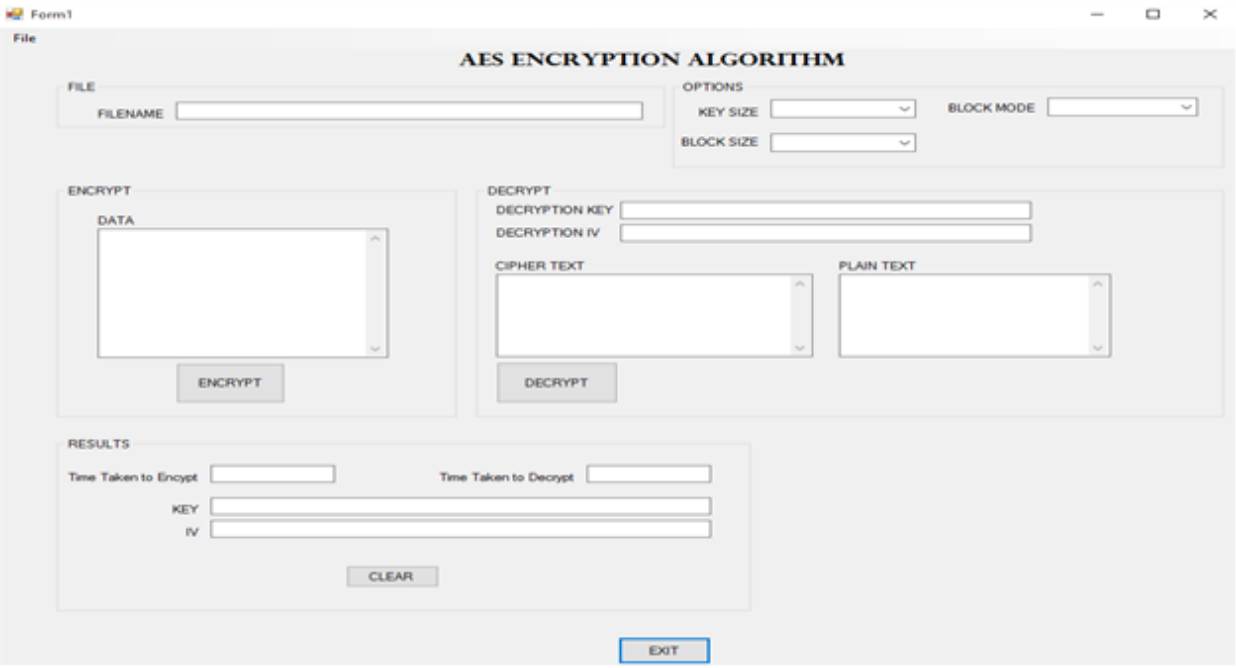

Fig. 2: Simulator GUI for AES encryption

3. Results and discussion

3.1 Symmetric algorithms

After the simulation was carefully carried out as described, the results were tabulated as follows. Table 1 shows the results for the simulation for the 3 symmetric algorithms tested and the smaller file sizes where the time is in milliseconds. Fig .3 shows the encryption times of 3 symmetric algorithms for small file sizes while Fig. 4 shows the decryption times of 3 symmetric algorithms for small file sizes with time measured in milliseconds. From Figures 3 and 4, for smaller file sizes, the 3DES algorithm averaged faster encryption and decryption time than the AES algorithm. The Blowfish algorithm had the slowest encryption and decryption time across all 5 file sizes tested. This shows that even at smaller file sizes, the encryption technique matters especially for time dependent applications such as voice applications where any form of delay cannot be tolerated even though security has to be put in place to stop packets from being hacked. This can be achieved by implementing the algorithm with the fastest encryption for the application.

Table 1: Simulation results for symmetric algorithms with smaller file sizes

Algorithm	File size (kg)	Encrypt speed (ms)					Average encrypt speed (m/s)	Decrypt speed (ms)					Average decrypt speed (m/s)
		1st	2nd	3rd	4th	5th		1st	2nd	3rd	4th	5th	
AES	1	2	2	2	2	2	2	1	1	1	1	1	1
	2	2	2	2	2	2	2	1	1	1	1	1	1
	3	2	2	2	2	2	2	1	1	1	2	1	1.2
	4	2	2	2	2	2	2	1	1	1	2	1	1.2
	5	2	2	2	2	2	2	2	2	1	1	2	1.6
3DES	1	2	2	2	1	1	1.6	1	1	1	1	1	1
	2	2	2	1	2	2	1.8	1	1	1	1	1	1
	3	2	2	2	1	2	1.8	1	1	1	1	1	1
	4	2	2	2	2	1	1.8	2	1	1	1	1	1
	5	2	2	2	2	2	2	1	1	1	2	1	1.2
Blowfish	1	3	3	3	3	4	3.2	3	3	3	3	4	3.2
	2	4	3	3	4	3	3.4	4	3	4	4	3	3.6
	3	4	4	4	3	3	3.6	4	4	4	4	4	4
	4	4	4	4	4	5	4.2	4	4	4	4	4	4
	5	4	4	5	4	5	4.4	4	4	4	5	4	4.2

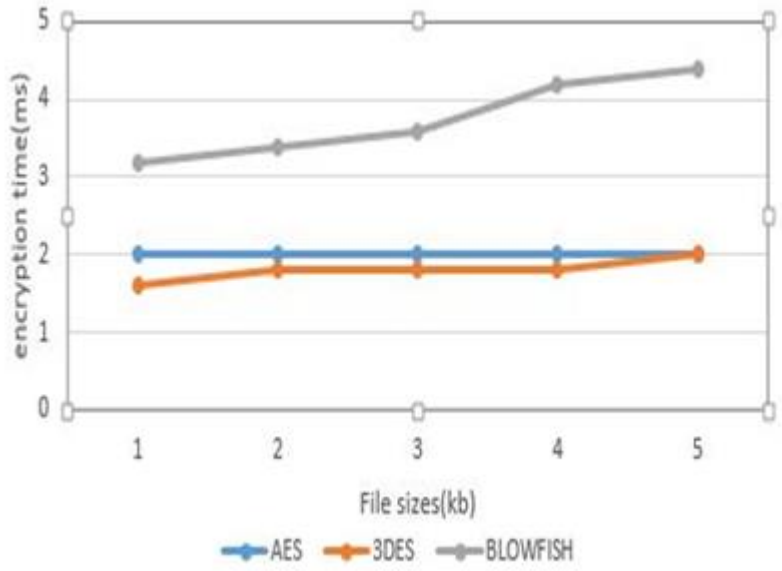

Fig. 3: Encryption times of 3 symmetric algorithms for small file sizes

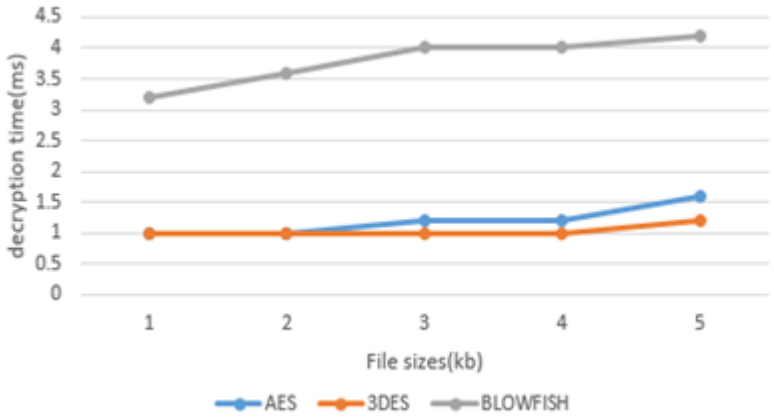

Fig. 4: Decryption times of 3 symmetric algorithms for small file sizes

For the larger file sizes, the following results were obtained. Table 2 shows the result of the simulation between two symmetric encryption algorithms, AES and 3DES. Fig. 5 shows the encryption times for the symmetric algorithms for larger file sizes while Fig. 6 shows the decryption times of the symmetric algorithms for larger file sizes where the time was measured in milliseconds. From the results plotted in Figures 5 and 6, it is clear that the AES algorithm has faster encryption and decryption times compared to the 3DES algorithm. This makes sense as the 3DES algorithm has way more computational processes involved in carrying out its encryption and decryption process. As the name suggests, it carries out the steps of the DES algorithm 3 times.

Table 2: Simulation results for symmetric algorithms with larger file sizes

Algorithm	File size (mb)	Encrypt speed (ms)					Average decrypt speed (m/s)	Decrypt speed (ms)					Average decrypt speed (m/s)
		1st	2nd	3rd	4th	5th		1st	2nd	3rd	4th	5th	
AES	5	63	57	62	65	66	62.6	75	68	74	80	98	79
	10	113	121	127	120	123	120.8	183	190	209	156	144	176.4
	15	167	167	198	162	174	173.6	262	227	280	226	263	251.6
	20	289	247	274	277	264	270.2	317	306	352	305	335	323
	25	268	281	285	273	276	276.6	347	395	374	369	351	367.2
3DES	5	419	429	429	434	427	427.6	474	482	449	447	519	474.2

10	910	1077	931	907	914	947.8	918	1004	989	878	905	938.8
15	1274	1254	1224	1255	1231	1247.6	1369	1327	1604	1371	1327	1399.6
20	1633	1704	1887	1719	1647	1718	1760	1740	1744	1783	1737	1752.8
25	1996	2034	2041	2021	2008	2020	2154	2308	2143	2325	2148	2215.6

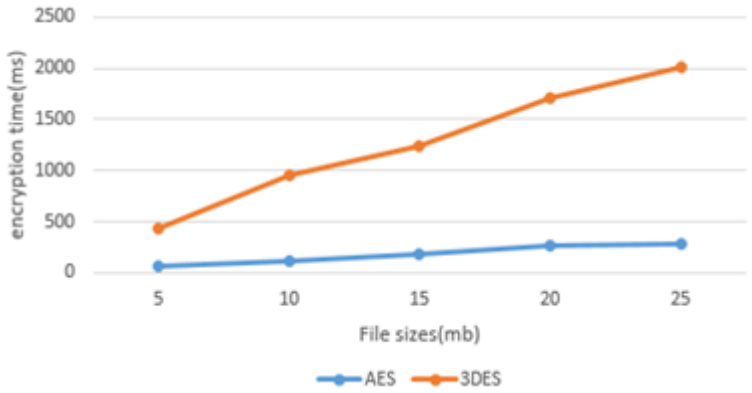

Fig. 5: Encryption times for the symmetric algorithms for larger file sizes

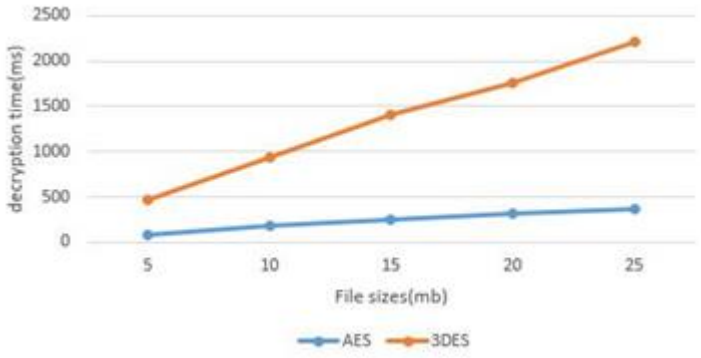

Fig. 6: Decryption times of the symmetric algorithms for larger file sizes

Generally, for smaller file sizes, it is seen that the 3DES algorithm is slightly faster than the AES algorithm but for larger file sizes, the AES algorithm is considerably faster than the 3DES algorithm.

3.2 Asymmetric algorithms

For the asymmetric algorithms, the following results were obtained. Table 3 shows the results of the simulation for the asymmetric encryption algorithms used to encrypt the keys used by the symmetric algorithms for their encryption processes. Fig. 7 shows the different encryption times of the asymmetric algorithms for different key sizes while Fig. 8 shows the different decryption times of the asymmetric algorithms for different key sizes. From the Figures 7 and 8, it can be seen that the diffie-hellman algorithm is much more efficient in the aspect of encryption time but not so much in decryption. The RSA algorithm decrypts faster but encrypts a lot slower. From the tests, it was seen that the RSA algorithm couldn't encrypt larger sized files as it threw an error of bad length. This further goes to show why these asymmetric algorithms are not used to encrypt data and just used only to encrypt the keys as it provides a safe medium for key exchange.

Table 3: Simulation results for asymmetric algorithms

Algorithm	Key size (bits)	Encrypt speed (m/s) 1st	2nd	3rd	4th	5th	Average decrypt speed (m/s)	Decrypt speed (m/s) 1st	2nd	3rd	4th	5th	Average decrypt speed (m/s)
RSA	112	91	84	83	101	137	99.2	3	3	3	4	5	3.6
	128	97	95	91	93	132	101.6	4	4	3	4	3	3.6
	192	91	105	133	156	101	117.2	3	3	4	4	3	3.4
	256	125	118	116	115	120	118.8	3	3	4	3	3	3.2
Diffie-Hellman	112	8	10	10	11	12	10.2	8	9	8	9	8	8.4
	128	8	9	8	11	10	9.2	11	9	8	7	8	8.6
	192	10	9	8	10	9	9.2	9	9	8	8	8	8.4
	256	9	9	11	10	10	9.8	8	9	8	8	8	8.2

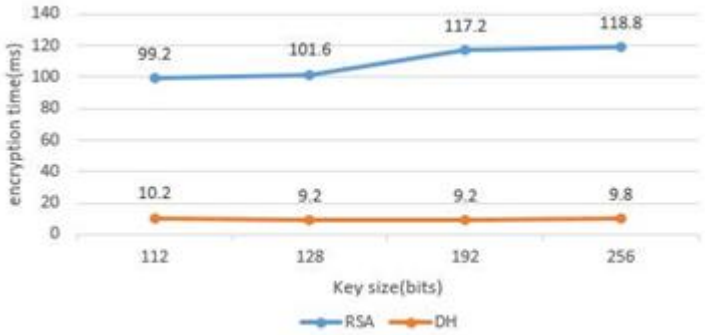

Fig. 7: Different encryption times of the asymmetric algorithms

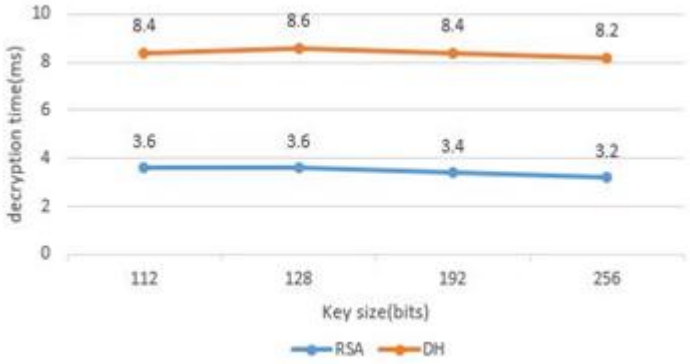

Fig. 8: Different decryption times of the asymmetric algorithms

4. Conclusions

This research carried out a simulation based analysis on cryptographic algorithms. The simulation environment included implementing the undertaken algorithms on the C# programming language which already has APIs for cryptography and steganography. The speed of encryption for each of these algorithms was the criteria for the comparative analysis between them. The symmetric algorithms discussed were the Advanced Encryption Standard (AES), Triple Data Encryption Standard (3DES) and the Blowfish algorithm. For the asymmetric algorithms, the Rivest-Shamir-Adleman (RSA) algorithm and the Diffie-Hellman Key exchange algorithms were considered. The simulation results showed that for smaller file sizes, the 3DES algorithm was faster compared to both AES and Blowfish. For larger file sizes, the AES algorithm was considerably faster than the 3DES algorithm in both encryption and decryption. The asymmetric algorithms, RSA and Diffie-Hellman, were used to encrypt sample keys and the Diffie-Hellman algorithm showed most efficiency in terms of encryption speed while the RSA algorithm decrypted faster. Previously from other researches, it has always been generalized that the AES algorithm is much faster

than other symmetric encryption algorithms discussed. Though this is somewhat true, this research has provided evidence that this is not true for all data sizes. It shows that at smaller data sizes, the 3DES algorithm is as fast as the AES algorithm and even faster. As the data size increases, due to the computational burden of the 3DES algorithm, it gets slower compared to the AES algorithm.

References

Awotunde, J. B, Ameen, A.O., Oladipo, I.D., Tomori, A.R. and Abdulraheem, M. (2016) Evaluation of four encryption algorithms for viability, Reliability and performance estimation. Nigerian Journal of Technological Development, 13: 74-82.

Ayyappadas, P., Devassy, A, George, S.C. and Devassy, A. (2015) Survey of Symmetric Cryptography Algorithms. IOSR Journal of Electronics and Communication Engineering, 5: 65-75.

Bisht, N. and Singh, S. (2015) A comparative study of some symmetric and asymmetric key cryptography algorithms, International Journal of Innovative Research in Science, Engineering and Technology. 4: 1028-1031.

Ebrahim, M., Khan, S. and Khalid, U.B. (2013) Symmetric algorithm Survey: A Comparative analysis. International Journal of Computer Applications, 61: 12-19.

Elminaam, D.S., Kader, H.M. and Hahoud M.M. (2008) Performance Evaluation of symmetric encryption algorithms. International Journal of Computer Science and Network Security, 8: 280-286.

Gupta, A. and Walia, N. (2014) Cryptography Algorithms: A review. IJEDR, 2: 1667-1672.

Koko, S.M. and Mustapha, A.B. (2015) Comparison of various Encryption Algorithms and Techniques for improving secure data Communication. IOSR Journal of Computer Engineering, 17: 62-69.

Koukou, Y.M., Othman, S.H., Siraj, M.M. and Nkiama, H. (2016) Comparative Study of AES, Blowfish, CAST-128 and DES Encryption Algorithm. IOSR Journal of Engineering, 6: 01-07.

Maqsood, F., Ahmed, M., Ali, M.M. and Shah, M.A. (2017) Cryptography: A Comparative Analysis for Modern Techniques. International Journal of Advanced Computer Science and Applications, 8: 442-448.

Ramdeo N. (2012) Evaluation of Data Encryption Algorithms. International Journal of Scientific and Engineering Research, 3: 1-3.

Ramesh, G. and Umarani, R. (2012) A comparative study of six most common symmetric encryption algorithms across different platforms. International Journal of Computer Applications, 46: 6-9.