IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 12 Issue 05, May 2025
ISSN (Online) 2348 – 7968 | Impact Factor – 6.72
www.ijiset.com

Performance Analysis of Symmetric Cryptographic Techniques

Kwame Owusu Bempah[1] and Isaac Owusu-Mensah[2]

[1]Department of Computer Science and Information Technology, Christian Service University, Kumasi, Ghana

[2] Department of Integrated Science Education, Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development, Kumasi, Ghana

Abstract
There are numerous cryptography techniques, and most users of these algorithms require algorithms that come with minimum cost and greater performance. Example, in banking transactions, preference is taken for security at a higher cost, unlike gaming applications that prioritize speed in their operation, but there is no algorithm that exhibits such characteristics and therefore requires analysis of these algorithms on platforms such as speed, strength, and security to give an intuition in choosing a suitable cryptography technique. In this paper, we have implemented and analyzed the performance of three symmetric algorithms: DES, AES-128, and the AES-512 bit algorithm on grounds of security and speed using the cryptography parameters encryption time, decryption time, and key generation time to detect which one is efficient.
Keywords: *DES, Decryption, Encryption, Symmetric Algorithm.*

1. Introduction and Preliminaries

It is always crucial to secure confidential data or documents from being invaded by a third party. This is accomplished through cryptography techniques called algorithms. Cryptography entails studying and applying mathematical methods known as algorithms to convert secret messages, data, or documents into illegible format [1] and includes the processes of encryption as well as decryption. In practicing cryptography in the areas of encryption and decryption, a secret key is always needed when converting plaintext into an unreadable format, which must be known by both the sender and the receiver. The key knowledge is used in classifying cryptography into two aspects, namely, symmetric cryptography and asymmetric cryptography [7], from which symmetric algorithms and asymmetric algorithms related to privacy [8] can also be coined. John Daemen and Vincent Rijmen initiated a symmetric algorithm called the Rijndael algorithm, which is popularly known as the Advanced Encryption Standard (AES), which has been widely accepted by the National Institute of Standards and Technology (NIST) [13]. DES is another symmetric technique known as Data encryption standard (DES), which was adopted by the US government as a standard encryption algorithm [13]. There are enormous cryptography algorithms or techniques that have been developed and used practically, which include DES, AES, Blowfish, ElGamal, and Paillier, that protect confidential data [2], and the analysis of their performances is a great field of research that needs to be analyzed always. Security, speed, key generation, encryption time and decryption time, and memory are very crucial cryptography parameters that are given top priority as far as cryptography algorithms regarding performance analysis are concerned. Seth and Mishra [4] performed analysis on three cryptography algorithms: AES, DES, and RSA, on the basis of their computational time, usage of memory, and byte usage, and the results showed that DES uses a small encryption time and the memory usage of AES is very minimal, but there is a slight difference in encryption time between the AES and DES algorithms. Both the encryption time and the memory usage for the RSA algorithm were found to be very high, while the output byte is very small. Song et al. [5] also did a comparative analysis of the performances of two cryptography algorithms, AES and DES, which were evaluated and analyzed based on their avalanche effect, simulation, and memory. It was discovered that the avalanche effect of AES was high as compared to DES, and AES requires less memory for implementation and also uses a smaller amount of time for encryption as compared to DES. More work can be done on encryption pertaining to images to enhance the security aspect of the system. Verma et al. [9] conducted an analysis of two cryptography algorithms, namely DES and blowfish. The focus of their comparison was to assess these algorithm's performance in area of secrecy, speed and power consumption. In the analysis, it was found that blowfish performs very fast as compared to DES and even AES, but in the research work of [12], the result also proved otherwise, where AES performs better than blowfish. The results shown above are performances that have been analyzed theoretically and

are not backed by implementation results. Motivated by the results above, in this paper, we extend by including a proposed symmetric AES algorithm in addition to DES and AES-128 algorithms and then implement the algorithms on an Eclipse platform, Java version 4.22 on an HP intel-core i3 processor, and then analyze their performance on grounds of their encryption time, decryption time, and key generation time to determine which one is efficient on the various platforms.

2. Main Results

In this section, we provide a brief summary of the algorithms and then analyze their performance using the metrics of encryption time, decryption time and key generation time. 2.1. AES-512 Bits Algorithmic Encryption. This algorithm is a modified AES512 Bits Data Encryption Algorithm[11] that uses plaintext characters of sixty-four bytes arranged in a (8×8) block matrix. 512 bits keys are randomly selected from the Galois Field, $GF(2^9)$ to encrypt the plaintext through five stages of transformations using the algorithm, $A_i = V + M_i$, where V represents the plaintext, M represents the keys, and A denotes the ciphertext. The stages of the algorithm are explained briefly below. 2.1.1. STATE. This stage is where the plaintext characters is arranged into (8×8) block matrix, and an initial key, $i = 1$ which contains sixty-four keys from $GF(2^9)$ is employed to start the process to produce $A_1 = V + M_1$. 2.1.2. $SKGF$. This step is where the keys left in the field are chosen randomly to encrypt the plaintext, V once more to create the sub algorithm, $A_i = \sum_{i=2}^{8}(V + M_i)[11]$. 2.1.3. SRL. In this stage, the subscript of A from the earlier two stages serves as the type of row and the number of times it must be moved to the left outside for creation of another matrix [11].

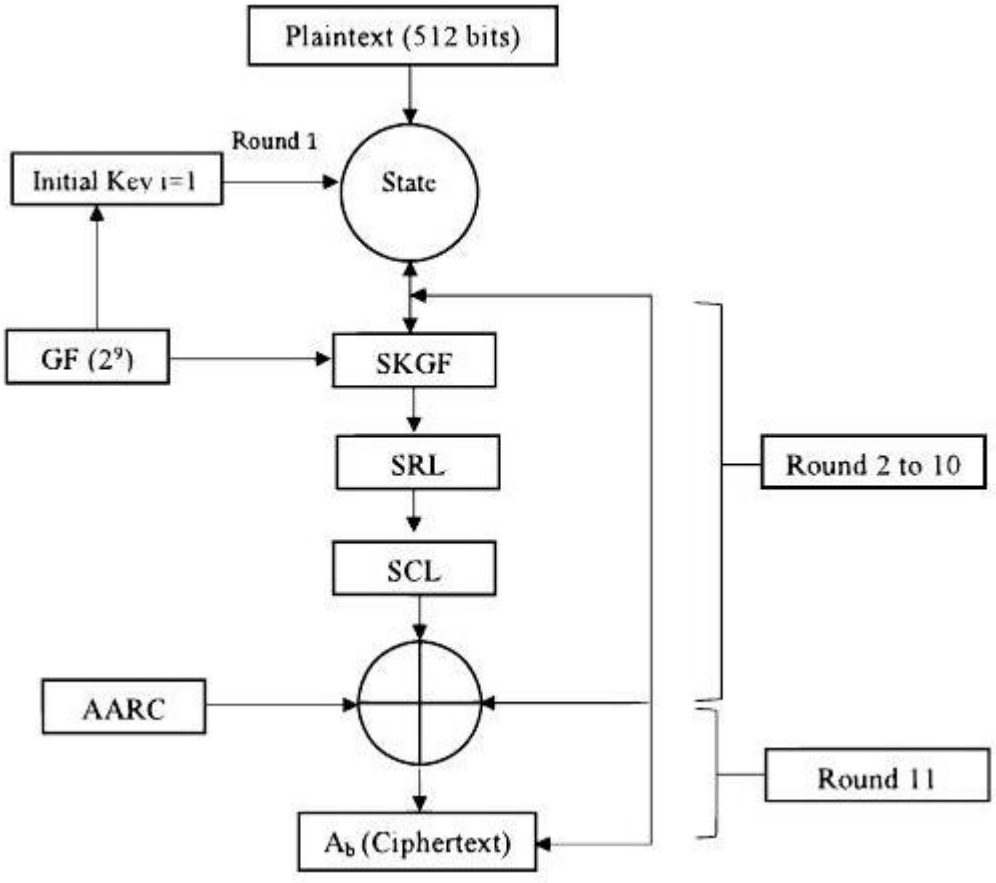

Figure 1. Modified AES-512 Bits Encryption Diagram 2.1.4. SCL. This stage is where we embark on column shifting. The output from the earlier stage is adopted where The subscript of A represents the type of column and how many times the entire column is moved to the left outside the matrix to generate a separate matrix [11]. we refer to this as Shift Column Left (SCL). 2.1.5. $AARC$. This represents the final step of the process where all the ciphertext in the SCL stage are

summed up to have a bulky cipher text, A_b. From the process of the algorithm, it can be vividly seen that the modified algorithm uses basic arithmetic operations to initiate the processes of the algorithm, which makes the algorithm very fast when undergoing encryption. The process for the modified AES-512 bits [11] encryption diagram is shown in Fig. 1.2.2 DES. DES comprises a key length of 56 bits and a block size of 64 bits. It began initially with a 64-bit key, but there was a restriction by the National Standard Authority (NSA) on DES to use a 56-bit key [3]. DES is able to perform in different modes called CBC, ECB, CFB, and OFB, making it adjustable [3]. 2.3. AES-128 Algorithm. In Advanced Encryption Standard, a plaintext is chosen for 128 bits, 192 bits, or 256 bits with either 128, 192, or 256 bits of keys, which undergoes ten rounds, twelve rounds, and fourteen rounds of encryption process respectively [10]. The AES-128-bit algorithm technique process of encryption of the plaintext undergoes five operational transformations (state, sub-byte, shiftrows, mixcolumn, and addroundkey), which are performed using a square block array of 4×4 matrix.

3. Performance Analysis

This section brings out the set up used for the experiment and their outcome from the experiment for symmetric algorithms. 3.1. Experimental set-up. We conducted the implementation and comparison of the DES, AES-128, and proposed AES-512 bit algorithms using the Java programming language within the Eclipse IDE. Our implementation utilized the Java security and crypto packages, which offer essential security functionalities such as encryption, decryption, key generation, and authentication. For our experimentation, we utilized solely text files of varying sizes ($42\ \text{KB}, 64\ \text{KB}, 150\ \text{KB}, 220\ \text{KB}$, and 286 KB) as input for encryption. The resulting encrypted output of each file was saved for subsequent decryption. To ensure fair analysis, we consistently utilized the same input files for all algorithms throughout the experiment. Additionally, we maintained uniform system conditions for all implementations and analyses to ensure consistent memory and processor settings for algorithm comparisons. The Java crypto and Java security packages use interfaces that are responsible for implementing the Java security architecture. These components are broadly categorized into two groups. To begin with, cryptography classes are available to perform information transmission operations. Additionally, there are authentication and access control classes that facilitate the management of message breakdown and digital signatures for user verification purposes. By utilizing these resources properly, we can customize cryptographic algorithms easily by tweaking certain functions within them. The process involved in implementing such customized algorithms using java. security and java.crypto packages involves several essential steps: generation of a key through the use of a generator class; creation of a cipher object specified with algorithm name and mode parameters; initiation of encryption or decryption modes on the cipher object; completion via execution employing the doFinal () procedure. 3.2. Parameters For Analysis. Each encryption algorithm possesses its own unique advantages and disadvantages, and it is therefore essential to understand their strengths and drawbacks when opting for an appropriate cryptography algorithm. In this paper, the following metrics are used for the analysis of the algorithms: 3.2.1. Encryption time. This refers to the time taken for the plaintext to be converted into ciphertext. The encryption time in our experiment is measured in seconds and is dependent on key size and mode of operation. The encryption time must be very minimal to make the system fast 3.2.2. Decryption time. This involves the time required to convert the ciphertext into plaintext and is measured in millisecond in our experiment. The decryption time must also be very small to make the system very fast.

Table 1. File size With Encryption/Decryption Time

Algorithms	File Size (KB) KB	Encryption time (s)	Decryption time (s)
DES	42	0.35	0.50
	64	0.90	0.61
	150	1.25	0.92
	220	1.50	1.50
	286	1.80	1.65

AES-128	42	0.30	0.30
	64	0.60	0.52
	150	0.85	0.72
	220	1.00	0.95
	286	1.14	1.12
AES-512	42	0.25	0.25
	64	0.55	0.50
	150	0.82	0.70
	220	0.98	0.81
	286	1.10	1.10

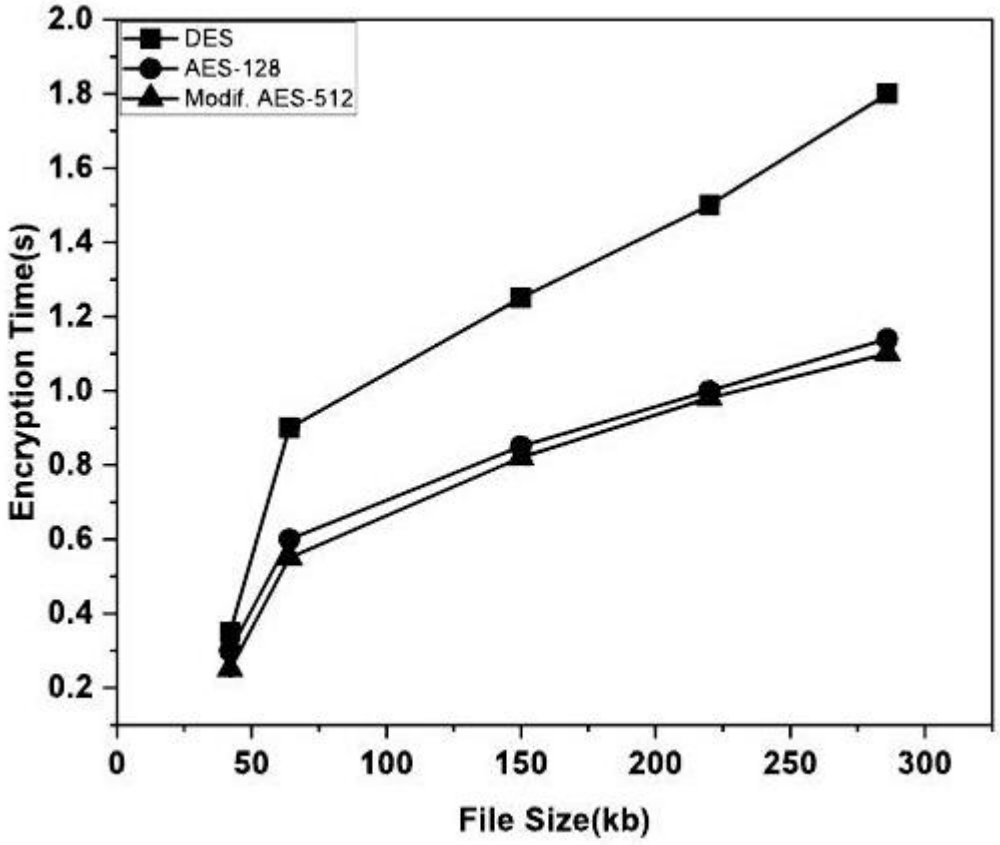

Figure 2. Graph showing Encryption time

3.2.3. Key generation time. Key generation time is the time within which an encryption key can be generated for the encryption of plaintext and is dependent solely on the length of the key. A longer time for key generation makes it harder for intruders to break the encryption through brute-force or other means and therefore ensures high security.

Table 2: Key Generation Time

Algorithms	Size of Key (Bits)	Key Generation Time(milliseconds)
DES	56	37 ms
AES-128	128	68 ms
AES-512	512	145 ms

4. Results and Analysis

Table 1 reveals the encryption time and decryption time, while Table 2 shows the key generation time of the algorithms, and the results of Table 1 depict that as the file size gets larger, the encryption time and decryption time also get inflated for each algorithm. Fig 2. and Fig 3. depicts the encryption times and decryption times of DES, AES128, and the modified AES-512 bit algorithms using text files of different sizes. It can be seen that AES-128 has mean encryption time and decryption time to be 0.778s and 0.722s respectively; DES has mean encryption time and decryption time to be 1.16 s and 1.036 s respectively and lastly AES-512 has mean encryption time and decryption time to be respectively 0.74 s and 0.672 s. AES- 512 has slightly faster encryption and decryption times compared to AES-128 and DES. In Conclusion, the modified AES-512 bit algorithm has slightly faster encryption and decryption times compared to AES-128. DES algorithm is the slowest among the three algorithms in both encryption and decryption processes. Fig 4. shows the key generation time against the length of the bits of the algorithms, and it can be seen that AES-128 with a key size of 128 bits has mean Key Generation Time of 68.0 ms; the modified AES- 512 with a key size of 512 bits has mean Key Generation Time of 145.0 ms and lastly; DES with a key size of 56 bits has mean Key Generation Time of 37.0 ms. The modified AES-512 has the highest key generation time, which is expected due to the

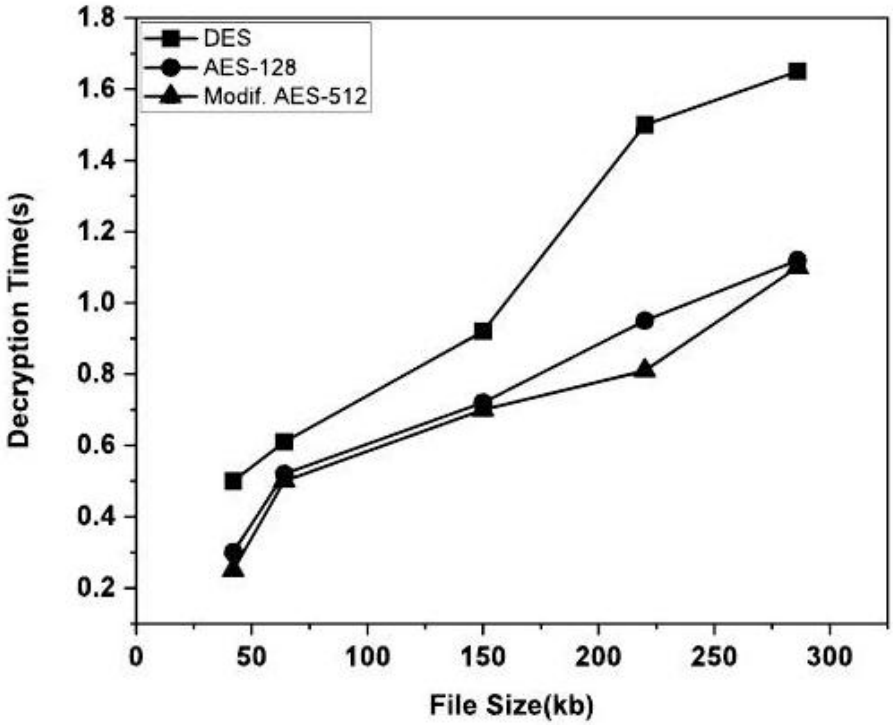

Figure 3. Graph showing Decryption Time

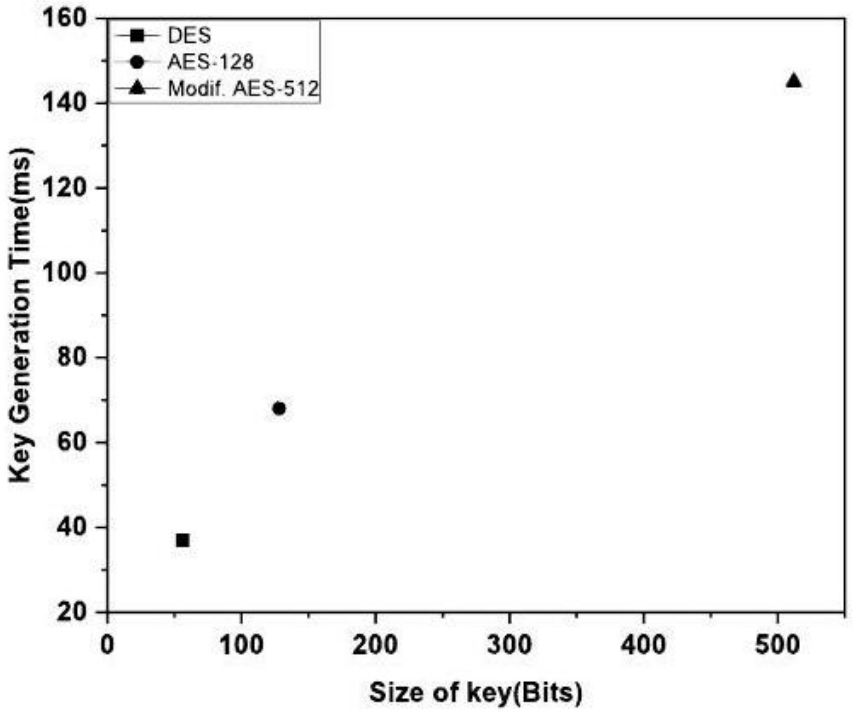

Figure 4. Graph showing Key Generation Time

larger key size. DES has the lowest key generation time, which is also expected due to the smaller key size. AES-128 falls somewhere in between, with a moderate key generation time. The longer key generation time for the modified AES-512 is justified by the higher security provided by the 512 -bit key, which makes it more resistant to brute-force attacks compared to shorter keys. Although the AES-128 algorithm has a shorter key generation time than AES-512, it still provides a high level of security. The 128-bit key is considered secure for most applications. Despite the shortest key generation time for the DES, the 56-bit key size is considered insecure by modern standards. It is vulnerable to brute-force attacks due to the relatively small key space.

4. Conclusion

From the analysis shown in this paper, it is seen that when opting for a particular cryptography algorithm, there should be an overview of the techniques of the algorithms with regard to their performance since each algorithm has its own strengths and drawbacks. The results of the experiment show that algorithms with shorter encryption times are generally more efficient and faster, which can be beneficial for applications requiring quick data processing. Similarly, to encryption, shorter decryption times indicate efficiency and speed. However, faster encryption times might sometimes indicate weaker security if the algorithm sacrifices complexity for speed. Larger keys are generally more secure because they provide a larger number of possible combinations, making it more difficult for an attacker to guess or brute-force the key. The choice of algorithm should balance security needs with performance requirements. From the analysis, the modified AES-512 Bit algorithm with moderate encryption and decryption times might offer a good trade-off between security and efficiency followed by AES-128 and lastly DES algorithm. However, the security of an algorithm should not be solely dependent on the key size but the underlying mathematical principles and implementation of the algorithm must play a crucial role. Example, some algorithms with smaller keys may be more secure than others with larger keys due to their design.

DECLARATION OF COMPETING INTEREST

The authors declare that there are no competing interests.

FUNDING SOURCES

This research did not receive specific grants from any funding agencies.

References

[1] Albrecht Beutelspacher. Cryptology. MAA, 1994

[2] 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(6):442-44

[3] Patil, Priyadarshini / Narayankar, Prashant / Narayan, D. G. / Meena, S. Md A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish 2016 Procedia Computer Science, Vol. 78 Elsevier p. 617-624

[4] Shashi Mehrotra Seth and Rajan Mishra. Comparative analysis of encryption algorithms for data communication 1. 2011.

[5] Y Song, N Kim, and D Kim. 2010 international conference on information and communication technology convergence. ICTC 2010, pages 579-589, 2010.

[6] Al Hasib, Abdullah / Haque, Abul Ahsan Md Mahmudul A comparative study of the performance and security issues of AES and RSA cryptography 20082008 third international conference on convergence and hybrid information technology, Vol. 2 p. 505-510.

[7] Kumar, Yogesh / Munjal, Rajiv / Sharma, Harsh Comparison of symmetric and asymmetric cryptography with existing vulnerabilities and countermeasures 2011 International Journal of Computer Science and Management Studies, Vol. 11, No. 03 p. 60-63

[8] Thirupalu, U. / Reddy, E. Kesavulu Performance analysis of cryptographic algorithms in the information security 2019 IJERT. NCISIOT-2019 Conference Proceedings, Vol. 8, No. 2 p. 64-9

[9] Om Prakash Verma, Ritu Agarwal, Dhiraj Dafouti, and Shobha Tyagi.Notice of violation of ieee publication principles: Peformance analysis of data encryption algorithms. In 2011 3rd International Conference on Electronics Computer Technology, volume 5, pages 399-403. IEEE, 2011.

[10] Rihan, Shaza D. / Khalid, Ahmed / Osman, Saife Eldin F. A performance comparison of encryption algorithms AES and DES, 2015 International Journal of Engineering Research and Technology (IJERT), Vol. 4, No. 12 p. 151-154.

[11] Bempah, Kwame Owusu / Gyamfi, Kwasi Baah / Boateng, Francis Ohene / Owusu-Mensah, Isaac A Modified AES-512 Bits Algorithm for Data Encryption 2024 European Journal of Pure and Applied Mathematics, Vol. 17, No. 2 p. 979-995.

[12] AL Jeeva, Dr V Palanisamy, and K Kanagaram. Comparative analysis of performance efficiency and security measures of some encryption algorithms. International Journal of Engineering Research and Applications (IJERA), 2(3):3033-3037, 2012.

[13] Jain, Rishabh / Jejurkar, Rahul / Chopade, Shrikrishna / Vaidya, Someshwar / Sanap, Mahesh AES algorithm using 512 bit key implementation for secure communication 2014 International Journal of Advanced Research in Computer Science, Vol. 1, No. 3 Citeseer

[14] Singh, Gurpreet A study of encryption algorithms (RSA, DES, 3DES and AES) for information security 2013 International Journal of Computer Applications, Vol. 67, No. 19 Citeseer

[15] Inproceedings Saqib, Nazar Abbas Rodriguez-Henriquez, Francisco and Diaz-Perez, Arturo AES algorithm implementation-an efficient approach for sequential and pipeline architectures 20