DOI: 10.36108/ujees/2202.40.0140

UNIOSUN Journal of Engineering and Environmental Sciences. Vol. 4 No. 1. March. 2022

Comparative Analysis and Performance Evaluation of Cryptographic Algorithms

Ogundoyin, I.K., Ogunbiyi, D.T., Adebanji, S. and Okeyode, Y.O.

Abstract: Encryption, which is based on the science of cryptography, is required to protect data and information in computer networks. As computing overhead rises, available encryption techniques are put to the test. It is necessary to assess the encryption algorithms' performance in order to establish their applicability for various security scenarios and applications. The findings of available research on comparison studies and performance evaluation of encryption algorithms are inconsistent, therefore the authors could not draw any conclusions about the encryption algorithms' performance based on different metrics. Three of the most widely debated encryption algorithms, Rivest, Shamir, Adelman (RSA), Advanced Encryption Standard (AES), and Data Encryption Standard (DES) were implemented and simulated in the Java programming environment in this study. Some selected text data files were used to drive the simulation. Metrics such as encryption time, decryption time, memory utilization and data size were used to measure the performances of the selected algorithms. When the three selected algorithms were run on 50 bytes data size, AES had encryption time of 40ms, decryption time 37ms, and memory utilization of 12MB. DES had encryption time 38ms, decryption time of 35ms and memory utilization of 08MB. RSA had encryption time 36ms, decryption time of 34ms and memory utilization of 10MB. When the three selected algorithms were run on 500 bytes data size, AES had encryption time of 65ms, decryption time 61ms, and memory utilization of 25MB. DES had an encryption time of 60ms, decryption time of 58ms and memory utilization of 17MB. RSA had an encryption time of 69ms, decryption time of 65ms and memory utilization of 21MB. In conclusion, DES performed better than other algorithms in both small and large data sizes for all metrics considered.

Keywords: Algorithm, Cryptography, Encryption, Performance evaluation,. Simulation ,

I. Introduction

Data and information transmissions via the Internet face major security issues in this era of increasing interconnectedness of computer networks and sophistication of cyber-threats [1]. Cryptography is an important component of data security mechanisms that can be used to address security issues. Cryptography protects the confidentiality and integrity of genuine users by making information unreadable to unauthorized people [1, 2]. There are a variety of cryptographic algorithms available, each with different performance capabilities and trade-offs. However, each user needs a cryptographic method with a high level of security performance that can meet their security needs.

Cryptography which includes both encryption and decryption is used in a variety of fields, including military communications, video mail, visual telephones, and digital television [3, 4]. Asymmetric and symmetric cryptographic systems are the two types of cryptographic systems. In symmetric systems, the transmitter and receiver use the same secret key for encryption and decryption, whereas in asymmetric systems, users have two keys, one for encryption and the other for decryption.

Ogundoyin, I.K., Ogunbiyi, D.T, Adebanji, S. and Okeyode, Y.O. (Department of Information and Communication Technology, Osun State University, Osogbo, Osun State, Nigeria)
Corresponding author: ibraheem.ogundoyin@uniosun.edu.ng
Phone Number: :+2348032481441
Submitted: 01-01-2022
Accepted: 29-01-2022

Data encryption is used to prevent unauthorized access to sensitive information. This is because making intercepted data as difficult as possible is a key line of defence in cyber security design.

This idea can be used to safeguard many kinds of data, ranging from confidential government information to personal or individual information, to name a few [1,5]. Various cryptographic algorithms have been studied, including RSA, AES , DES, 3DES, Blowfish, and others [1,4,5]. There have been a lot of efforts put into evaluating the performances of the algorithms in order to identify their suitability for various types of information security needs. Despite the research efforts, the existing comparative studies reported inconsistent results, and choices of cryptographic algorithms selected for comparison skewed toward symmetric encryption algorithms in most cases. Therefore, an effort was made to strengthen some existing studies through the results gotten from the current study and, also the choices of encryption algorithms for comparison in this study were selected from both symmetric and asymmetric encryption/decryption algorithms as against choices made in most previous studies.

There are quite a good number of works regarding the state- of- the-art, cryptography, which includes: encryption, decryption and steganography. The cryptography systems have been used in many data and information security instances.

In [2], Onyesolu and Ogwara developed a hybrid encryption system that includes two symmetric encryption methods, 3DES and AES. In terms of preventing attacks on database information, the new hybrid method developed proved to be more reliable. However, the reason for choice chosen algorithms hybridized not stated.

Pavithra and Ramadevi [4] compared four of the most extensively used symmetric key algorithms, including DES, 3DES, AES, and Blowfish. Rounds block size, key size, encryption/decryption time, and CPU process time were utilized as comparison metrics in terms of throughput and power consumption. According to the findings, blowfish is a better alternative than AES. This study did not include RSA.

Rani and Kaur [5] studied several cryptographic methods and thereafter proposed a hybrid encryption technique that combined the AES and Elgamal algorithms.

The novel hybrid cryptographic technology was designed to improve throughput by reducing encryption and decryption times. The proposed scheme improved network security. The study did not consider basic and widely mentioned cryptographic algorithms.

Sandip and Vijay [10], in a recent research of several encryption and decryption strategies, examined some of the available algorithms based on certain criteria in order to fulfil users' needs and to recognize the algorithm cost-performance trade-off. The findings allow users to select encryption algorithms that meet their performance needs.

Hamouda in [11] presented a performance comparison of encryption methods. The study examined the three of the most useful algorithms: DES, 3DES, and AES. The algorithms were evaluated for data security, encryption time, and throughput. The results of the experiments revealed that the algorithms performed differently depending on the inputs. RSA was not considered in this study.

Awotunde et. al. [12], presented a comparison of many algorithms (Blowfish, AES, 3DES, and DES). The review investigated the memory building rate, different key sizes, CPU use time period, and encryption speed of the four methods to see how much computer resource was consumed and how long it took each algorithm to perform its task RSA not included in the comparison.

Onder and Ender [13] presented an encryption technique that is strong enough to protect data from hostile attackers. The method ensures that the data sent between the sender and receiver is kept private. The authors employed a blockchain architecture that could be implemented rapidly and with minimal resources, and the blockchain algorithm was immune to message change attacks. It was also particularly efficient in terms of digital design, as it had a high volume of encrypted information flowing through it and required quick decryption. Other relevant work includes the following [1, 6-9].

Available comparative studies and performance evaluation of basic cryptographic algorithms reported conflicting results and, choices of cryptographic algorithms selected for comparison skewed toward symmetric encryption algorithms in most cases. Therefore, effort was made to strengthen some existing studies through the results gotten from the current study and, also the choices of encryption algorithms for comparison in this study were selected from both symmetric and asymmetric encryption/decryption algorithms as against choices made in most previous studies. The algorithms of choices were three most mentioned of the cryptographic algorithms in literature. The adopted algorithms were simulated and evaluated to have insight into their efficiencies in term of some performance metrics. This would guide users to suitably adopt the encryption algorithms to different information security situations.

II. Materials and Methods

The methodology of this research includes various methods proposed to achieve the goal of the paper. The methods consist of: data description, adoption of encryption and decryption algorithms, simulation and evaluation of the performances of the selected algorithms. The three popular encryption algorithms adopted in the study are AES, DES and RSA. The algorithms were simulated in a Java programming environment, and some selected text data files were used to drive the simulation. Metrics such as encryption time, decryption time, memory utilization and data size were used to measure the performances of the selected algorithms.

A. Data Description

The simulation experiments were conducted using five text files of various sizes and formats, as well as a comparison of the three selected algorithms: AES, DES, and RSA.

B. Description of the Adopted Cryptographic Algorithms

The selected file encryption algorithms (RSA, AES and DES) were implemented using Java programming language on IntelliJ IDEA.

i. RSA

The RSA algorithm is a widely used public key cryptographic algorithm. Rivest, Shamir, and Adleman were the three mathematicians who created RSA. RSA is an asymmetric encryption algorithm, it is found in many applications and can be used for key exchange, digital signatures, and data encryption. RSA employs a variable-size encryption block and a variable-size key.

The key-pair consists of a very high integer, n, which is the product of two prime numbers calculated using special processes [1, 4]. The RSA algorithm is represented in Figure 1:

ii. AES

The substitution-permutation network design paradigm underpins AES. The block size in AES is 128 bits, and the key size is 128,192, or 256 bits. AES uses the state, which is a 4 x 4 column-major order matrix of bytes. In a finite field, the majority of AES calculations are done. The AES cipher is defined as a set of transformation rounds that turn plaintext input into ciphertext output [1, 4]. The number of repeat cycles is as follows:

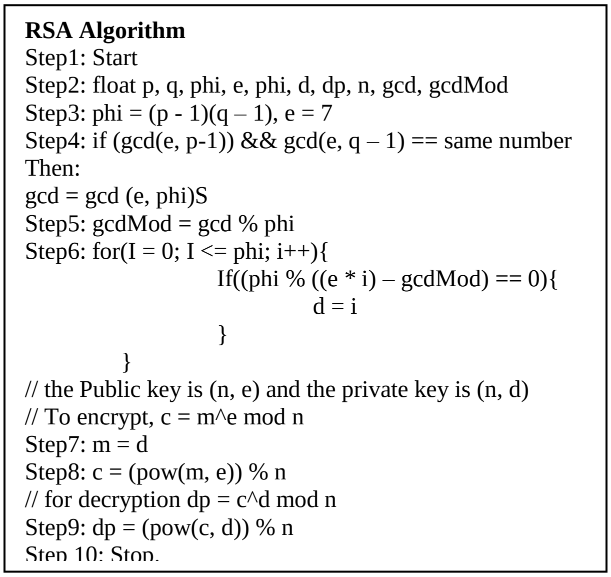
- For 128 bit keys, there are ten cycles of repetition.
- For 192 bit keys, there are 12 cycles of repetition.
- For 256 bit keys, there are 14 cycles of repetition.

The algorithm for AES is represented in Figure 2

iii. DES

Data is encrypted in 64-bit chunks using the DES method. DES accepts 64 bits of plain text as input and generates 64 bits of cipher text as output.

The key has a length of 64 bits. DES creates a permutation of all 2^{64} possible 64-bit combinations, each of which can be 0 or 1. Each 64-bit block is divided into two 32-bit blocks, one on the left and one on the right. A 64-bit cipher block C is converted to a 64-bit message block M using the DES algorithm. When each 64-bit block is encrypted separately, the Electronic Code Book (ECB) form of encryption is utilized. Chain Block Coding, CBC and DES are the two kinds of DES encryption (DES). The algorithm for DES is represented in Figure 3.



```
RSA Algorithm
Step1: Start
Step2: float p, q, phi, e, phi, d, dp, n, gcd, gcdMod
Step3: phi = (p - 1)(q – 1), e = 7
Step4: if (gcd(e, p-1)) && gcd(e, q – 1) == same number
Then:
gcd = gcd (e, phi)S
Step5: gcdMod = gcd % phi
Step6: for(I = 0; I <= phi; i++){
                If((phi % ((e * i) – gcdMod) == 0){
                        d = i
                }
        }
// the Public key is (n, e) and the private key is (n, d)
// To encrypt, c = m^e mod n
Step7: m = d
Step8: c = (pow(m, e)) % n
// for decryption dp = c^d mod n
Step9: dp = (pow(c, d)) % n
Step 10: Stop.
```

Figure 1: RSA Algorithm

```
AES Algorithm
Step1:  Start
Step2:From the cipher key, create a series of round keys.
Step 3: Use the block data to initialise the state array (plaintext).
Step 4: To the starting state array, append the initial round key.
Step 5: Calculate nine rounds of state manipulation.
Step 6: Calculate the tenth and final round of state manipulation.
Step 7: Stop
```

Figure 2: AES Algorithm

C. Experimental Design

Simulation was setup for the comparative study of the selected cryptographic algorithms (AES, DES, and RSA). Figure 4 shows a flowchart of simulation flow of the selected cryptographic algorithms. From Figure 4, the start symbol shows commencement of the simulation.

Step1: Start

Step2: The IP-initial permutation function in the DES algorithm accepts a 64-bit plain text block and performs permutation on it.

Step3: It divides the data into two halves, RTP (Right Plain Text) and LTP (Left Plain Text).

Step4: Each of these blocks (RPT and LPT) goes through a 16-round encryption process.

Step5: The 16-round encryption described above, followed the following steps: The key transformation, expansion permutation, S-Box and P-Box permutations, and ultimately the swap or XOR are all covered.

Step6: To obtain a DES ciphertext in 64 bits, the block is recombined from RPT and LPT in an FP-Final Permutation on the just combined block.

Step7:The process for decryption uses the reverse order of the same key since DES is a symmetrical algorithm

Step8: Stop

Figure 3: DES Algorithm

Following the commencement of the simulation process, is to choose the cryptographic algorithm of choice (AES, DES, RSA). It is to be noted that all the algorithms have been implemented as a module in the Java programming environment. The selection of algorithm of choice for the simulation is to call the module or function implementing any of the cryptographic algorithms. The selection of algorithm of choice will be followed by inputting the text to be encrypted. This will be followed by inputting the encryption key. Both texts to be encrypted and the encryption key will be validated. This will be followed by running the selected encryption module of the cryptographic algorithm to produce the cipher text. The corresponding decryption module will also be run to produce the original text. Figure 5 is an interface showing a simulation environment designed for the selected cryptographic algorithms.

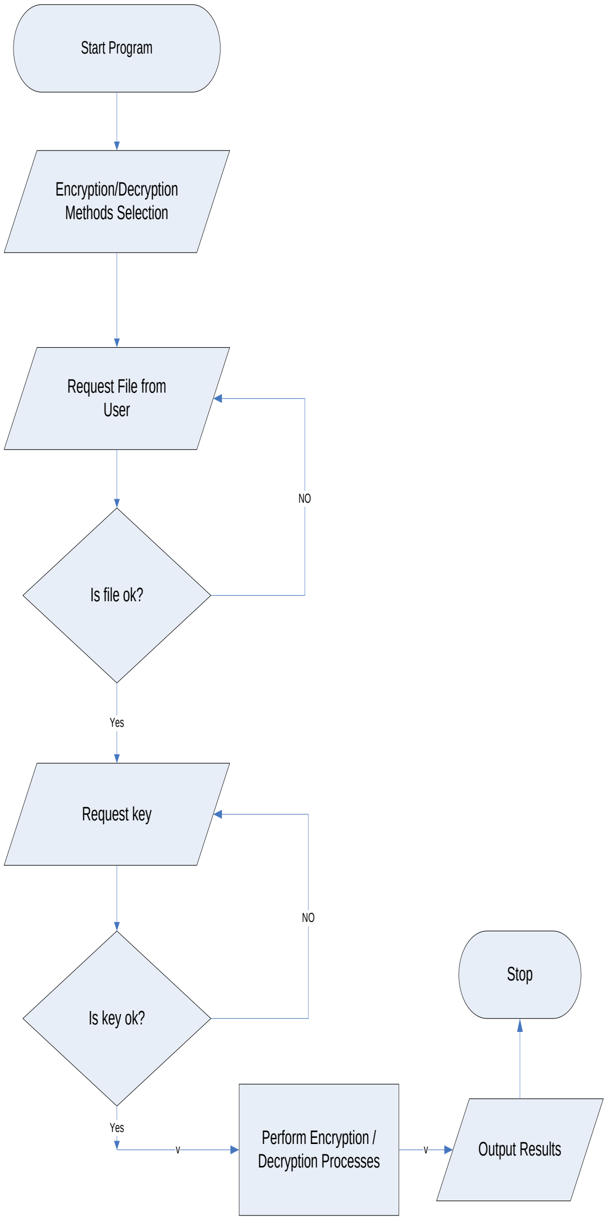
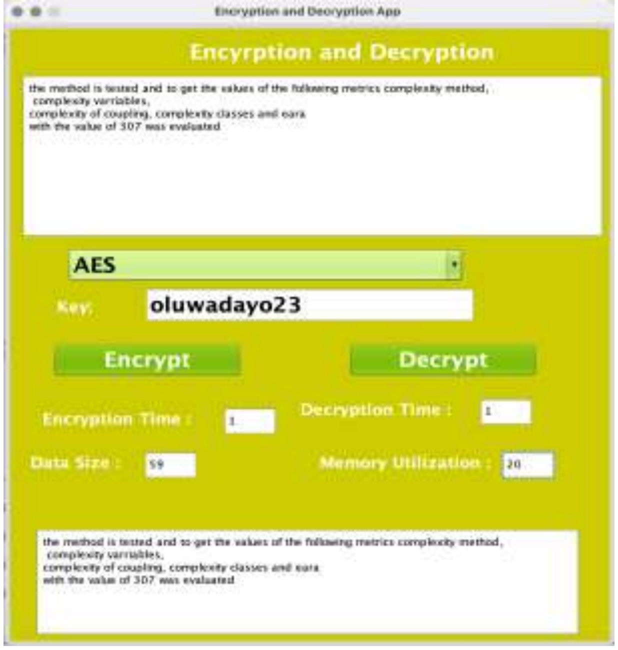


Figure 4: Simulation Flow of the Selected Cryptographic Algorithms



the method is tested and to get the values of the following metrics complexity method,
 complexity varriables,
complexity of coupling, complexity classes and eara
with the value of 307 was evaluated

AES

Key: oluwadayo23

Encryption Time : 1

Decryption Time : 1

Data Size : 59

Memory Utilization : 20

the method is tested and to get the values of the following metrics complexity method,
 complexity varriables,
complexity of coupling, complexity classes and eara
with the value of 307 was evaluated

Figure 5: Interface Showing Simulation Environment Designed for the Selected Cryptographic Algorithms

D. Performance Metrics

The following metrics were used to evaluate the performance of the selected encryption and decryption algorithms:

i. **Encryption time:** The time taken to generate a cipher text from plain text is known as encryption time. It was calculated as follow:

$$T = \frac{DS}{S} \qquad (1)$$

Where, T = Time, DS = Data size, S = Speed.

ii. Decryption time: Decryption time is the time it takes to convert cipher text to plain text. Equation 1 is also used to calculate decryption time

iii. Memory utilization: Memory utilization is calculated using the formula:

$$Mu = TM - (FM + BM + CM) \qquad (2)$$

Where, MU = Memory Utilization, TM = Total Memory, FM = Free Memory, BM = Buffers Memory, and CM = Cached Memory.

III. Results and Discussion

The selected file encryption algorithms (RSA, AES and DES) were implemented and simulated using Java programming language on IntelliJ IDEA. Table 1, Table 2 and Table 3 show the results of the simulation carried out using data of different file sizes. Based on the given metrics, Table 1 shows the Encryption time of the selected algorithms (AES, DES, RSA). When the three selected algorithms were run on 50 bytes data size, AES had an encryption time of 40ms, DES had 38ms and RSA had 36ms. When the three selected algorithms were run on 500 bytes data size, AES had an encryption time of 65ms, DES had 60ms and RSA had 69ms. Figure 6 shows a graph comparing the encryption time of the selected algorithms.

Table 2 shows the decryption time of the selected algorithms (AES, DES, RSA). When the three selected algorithms were run on 50 bytes data size, AES had a decryption time of 37ms, DES had 35ms and RSA had 34ms. When the three selected algorithms were run on 500 bytes data size, AES had a decryption time of 61ms, DES had 58ms and RSA had 65ms. Figure 7 shows a graph comparing the decryption time of the selected algorithms.

Table 3 shows the memory utilization of the selected algorithms (AES, DES, RSA). When the three selected algorithms were run on 50 bytes data size, AES utilized 12MB, DES utilized 08MB, and RSA used 10MB. When the three selected algorithms were run on 500 bytes data size, AES utilized 25MB, DES used 17MB and RSA utilized 21MB. Figure 8 shows a graph comparing the memory utilization of the selected algorithms.

Table 1: Encryption Time of the Selected Algorithms (AES, DES, RSA)

Data Size(Byte)	Encryption time(s)		
	AES	DES	RSA
50	40	38	36
100	50	46	47
150	60	53	61
200	62	56	65
500	65	60	69

Table 2: Decryption Time of the Selected Algorithms (AES, DES, RSA)

Data Size(Byte)	Decryption time(s)		
	AES	DES	RSA
50	37	35	34
100	46	43	44
150	57	50	58
200	60	52	62
500	61	58	65

Table 3: Memory Utilization of the Selected Algorithms (AES, DES, RSA)

Data Size(Byte)	Memory Utilization (Byte)		
	AES	DES	RSA
50	12	08	10
100	17	12	15
150	18	14	14
200	23	15	20
500	25	17	21

Discussing the simulation results, it was observed that on a small data size, RSA had the best performance in terms of encryption and decryption time.

However, as the data size progressively increased, there was a decline in the performances of RSA. When a data size of 500Byte was used, RSA had the poorest performance; While DES had the best performance.

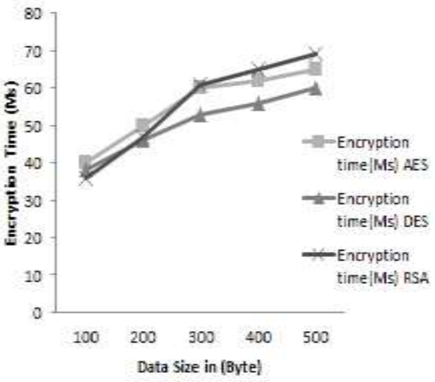


Figure 6: Graph showing comparative analysis of Encryption time of the selected algorithms (AES, DES, RSA)

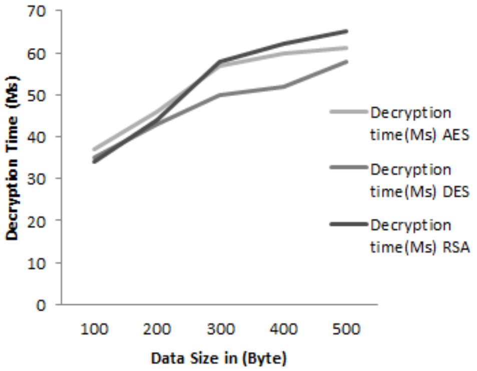


Figure 7: Graph showing comparative analysis of deccryption time of the selected algorithms (AES, DES, RSA)

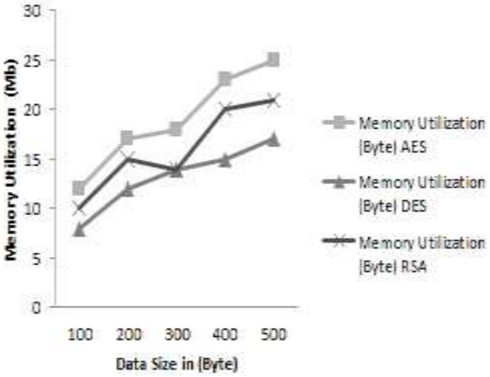


Figure 8: Graph showing comparative analysis of memory utilization of the selected algorithms (AES, DES, RSA)

A study of the result also revealed that AES only had a good performance at large data sizes, but the performance was not better at small data sizes. In overall performance, DES seemed to be better than every other algorithm based on the metrics used for evaluation as shown in Tables 1 – 3. Figures 6 - 8 also show different characteristics of the selected algorithms in terms of performance metrics used. The results further showed that AES encryption algorithm had the least performance in terms of metrics used. From the performance results of the selected algorithms, DES performed better than RSA mostly on large data sizes, this is mainly due to the computational complexity of RSA program implementation and, also RSA is a public cryptographic algorithm having both secret and private key. DES is symmetric and has only one key, which is secret. The results of the study gave impetus on the applicability of the studied algorithms. Since RSA had good performance on small data sizes and DES had good performances on large size data, the two algorithms could be hybrid to design a very strong encryption system for data protection. This also explains the reliability of RSA algorithm on data security in terms of encryption time, decryption time and memory utilization, and how it is better than AES. In overall performance, DES had the best performance compared to RSA and AES. Finally, it can be seen that increasing the key size results in a significant shift in memory utilization as well as encryption and decryption times.

IV. Conclusion

This research presented performance evaluations of most mentioned cryptographic algorithms in literature and simulated them in Java programming language environment. The results of the research were in line with the results of some previous studies, to give the previous works better footing and strengthened their claims. Furthermore, the results demonstrated the characteristics of the selected algorithms (RSA, AES and DES) in terms of metrics considered in the study, thereby giving better insight into the applicability of the various cryptographic algorithms. The results of this study are relevant and will enable users and designers to choose appropriate cryptographic algorithms for specific applications security performances.

References

[1] Lalit, M.G., Abdus, S. and Hitendra, G. "TBHM: A Secure Threshold-Based Encryption Combined With Homomorphic Properties for Communicating Health Records", *International Journal of Information Technology and Web Engineering (IJITWE), IGI Global*, vol. 15, no. 3, 2020, pp. 1-17.

[2] Onyesolu, M.O. and Ogwara, N.O. "On Information Security using a Hybrid Cryptographic Model", *International Research Journal of Computer Science (IRJCS)*, vol. 4, no. 11, 2016, pp. 15- 22.

[3] Irawan, A., Andri, H., Alif, F. and Sufa, A. "E-Document Authentication with Digital Signature Model for Smart City in Indonesia", *Journal of Engineering Science and Technology Special Issue on INCITEST2019*, 2020, pp. 28 – 35. Pavithra and Ramadevi "Performance Evaluation of the Symmetric Key Algorithms", *Journal of Global Research in Computer Science*, vol. 3, no. 8, 2012, pp. 243.

[4] Rani, S. and Kaur, H. "Implementation and Comparison of Hybrid Encryption Model for Secure Network using AES and Elgamal", *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, 2017, pp. 45-46.

[5] Surbhi, B, Rajdeep, H, Rahul "Design and Analysis of Secure One-way Function for the

Protection of Symmetric Key Cryptosystems". *International Journal for Research in Applied Science & Engineering Technology*, vol. 9, no. 12, 2020, pp. 245-250.

[6] Zefreh, E.Z. "An Image Encryption Scheme Based on a Hybrid Model of DNA Computing Chaotic Systems and Hash Functions", *Multimed Tools Appl,* 79, 2020**,** pp**.** 24993–25022.

[7] Dwiki, and Ida, "Data Securing of Patients in Cloud Computing using a Combination of Sha256 and MD5", *International Journal of Advanced Engineering Technology, vol.* 6, no. 4, 2020, pp. 234-240.

[8] Pravin, S. and Rahul, M. "Performance Analysis of Cascaded Hybrid Symmetric Encryption Models", *Turkish Journal of Computer and Mathematics Education*, vol. 12, no.2, 2021, pp. 1699-1708.

[9] Sandip, T. and Vijay, K.V. "A Recent Study of Various Encryption and Decryption Techniques", *International Research Journal of Advanced Engineering and Science*, vol. 1, no. 3, 2016, pp. 92-94.

[10] Hamouda, B.E.H.H. "Comparative Study of Different Cryptographic Algorithms", *Journal of Information Security*, vol. 11, 2020, pp. 138-148. https://doi.org/10.4236/jis.2020.113009

[11] Awotunde, J.B., Ameen, A.O. Oladipo, I.D., Tomori, A.R. and Abdulraheem, M. "Evaluation of Four Encryption Algorithms for Viability, Reliability and Performance Estimation", *Nigerian Journal of Technological Development*, vol. 13, no. 2, 2016, pp. 74 -84.

[12] Onder, S. and Ender, S. "Cross-object Information Security: A Study on New Generation Encryption", AIP Conference Proceedings, 2019, pp. 2086, 030034 https://doi.org/10.1063/1.5095119