A Note on Time and Space Complexity of RSA and ElGamal Cryptographic Algorithms

Adeniyi Abidemi Emmanuel[1], Okeyinka Aderemi E[2], Adebiyi Marion O[3], Asani Emmanuel O[4]
Department of Computer Science, College of Pure and Applied Sciences
Landmark University, Omu-Aran
Kwara State, Nigeria

***Abstract*—The computational complexity study of algorithms is highly germane to the design and development of high-speed computing devices. The whole essence of computation is principally influenced by efficiency of algorithms; this is more so the case with the algorithms whose solution space explodes exponentially. Cryptographic algorithms are good examples of such algorithms. The goal of this study is to compare the computational speeds of RSA and ElGamal cryptographic algorithms by carrying out a survey of works done so far by researchers. This study has therefore examined some of the results of the studies already done and highlighted which of the RSA and ElGamal algorithms performed better under given parameters. It is expected that this study would spur further investigation of the behaviour of cryptographic structures in order to ascertain their complexity and impact on the field of theoretical computer science. The experimental results of many of the papers reviewed showed that RSA cryptographic algorithm performs better as regards to energy usage, time complexity and space complexity of text, image and audio data during encryption process while some studies showed that ElGamal performs better in terms of time complexity during decryption process.**

***Keywords*—RSA algorithm; ElGamal algorithm; time complexity; space complexity; data security**

I. Introduction

Cryptography is the scientific technique of converting plain texts to non-readable form and back to plain text again. Born out of wartime exigencies, the idea of cryptography is to make texts unreadable to unauthorized or unintended users. This is done by deploying various cryptographic algorithms [1],[2]. Data sent in an encrypted state can only be decrypted by the targeted party using cypher keys. Thus, it hard or in many cases impossible to decode by an intruder who intercepts the encrypted files. Contemporary cryptosystem security is not focused on data secrecy but on the secrecy of a relatively small amount of knowledge, called a cypher key [2],[3]. There are five major functions of cryptography, viz. authentication, privacy, integrity, non-repudiation and service reliability. Authentication encompasses the processes of verification. Cryptography helps to verify authenticity of the data source, as well as that of the data, that is to ensure the data has not been modified. This is achieved through public key infrastructure (PKI), digital certificate and digital signature. Cryptography also ensures the privacy of the data being transferred. In order words, the data is protected against unauthorized user and attacks. Cryptography also ensures the integrity of the data that is to ensure the message has not been modified. Non-repudiation means that the sender and recipient cannot dispute they've had the message sent. Finally, cryptography ensures Reliability in service; this is to ensure that the users are provided with quality service since systems are prone to attack.

Cryptographic algorithms are generally classified either as private key cryptography or public key cryptography (see Fig. 1 for the cryptography classification tree).

Private key cryptography, also referred to as Secret key cryptography (SKC) algorithms or Symmetric cryptography are set of one key techniques in which the encryption and decryption process require one and the same key [4]. As shown in Fig. 2, the sender uses the key to scramble the plaintext (or any set of values) and send the cipher text to the receiver. The recipient uses the same key to decrypt the information and retrieve the plaintext. Since both functions have a single key, secret key cryptography is often called symmetric encryption.

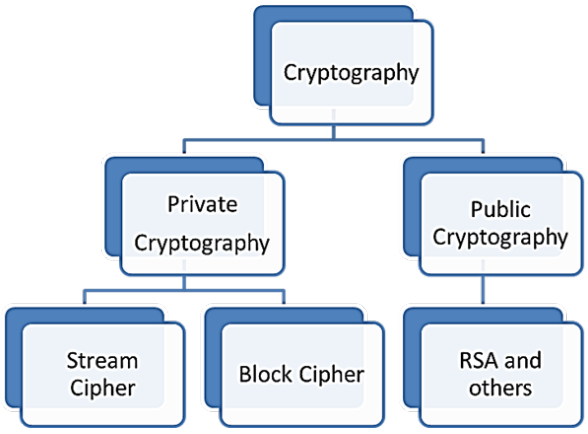

Fig. 1. Classification Tree of Cryptographic Algorithms.

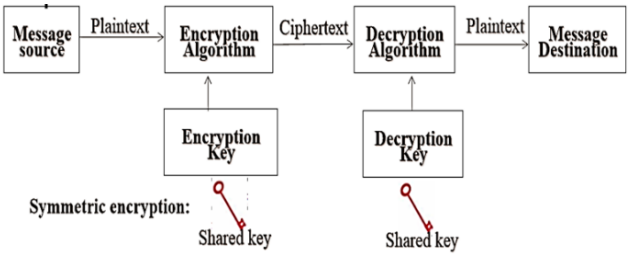

Fig. 2. Private Key Cryptographic Algorithm [5].

SKCs are further classified into Stream Buffers and Block Buffers. The Stream buffer encodes a message's characters a little at a time. Stream Cipher is used on a data stream by working on it in bits at one time. Block cipher accepts bits and then encrypts them as one entity [6]. If data is in blocked state, the data is encrypted / decrypted. Most commonly employed cryptographic secret key techniques include: Data Encryption Standard (DES), Triple Data Encryption Algorithm (3-DES), Advanced Encryption Standard (AES), BLOWFISH, and Rivest Cipher 4 among many others. The main hindrance to the practical application of a symmetric-key method is the need to share secret keys properly [7]. This has to be executed in a way to limit vulnerability or interception of the message. In the past, this will always have to be achieved by some sort of face-to - face interaction, which in certain situations appears very unrealistic when considering space and speed. If one believes that privacy is a concern, first of all the sharing of keys becomes more troublesome because of the need for a secure transferring of information. Some other concerns about the symmetric technique relates to secret key mismatch. All individual does have an equivalent secret key in the symmetric encryption. If the number of transaction participants rise, the probability of conflict or key mismatch increase drastically. Increasing new user requires more possible vulnerability point which an intruder might exploit. If such an attacker succeeds in obtaining control of just one of the secret keys, others will be totally compromised [8].

Public keys on the other hand are referred to as Asymmetric cryptography because the encoding key varies from the decoding key, which is private to the user only [9]. Asymmetric cryptography employs the use of mathematical techniques in the generation of keys without the possibility of being able to generate one key from the other. Anyone who may want to submit a message gets a public key readily accessible. A second key is kept private, so only the recipient is aware of it. Files that are encoded using the public key (text, binary files or documents) could not be decoded using the same method, but using the corresponding private key. Public key cryptographic algorithms are known to be slower during encryption process due to number of files generated. Nevertheless, they are more secure when it comes to security measures. Fig. 3 depicts the encryption and decryption processes of the Asymmetric cryptography.

Some of the commonly used public key cryptographic algorithms are: RSA, DIFFIE-HELLMAN, PAILLIER and Elgamal.

These cryptographic algorithms are combinatorial in nature; hence an evaluation of their complexities is sacrosanct to their design, development and deployment in high-speed computing devices. Therefore, the goal of this study is to review existing experimental works in literature with special focus on the complexities of the RSA and ElGamal cryptographic algorithms.

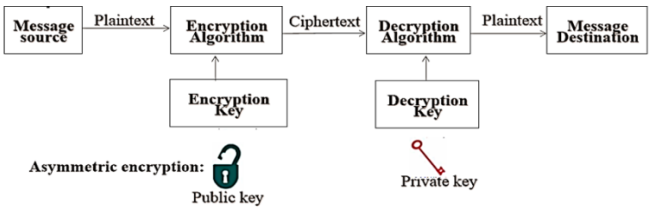

Fig. 3. Public Key Cryptography Algorithm [5].

II. The RSA Algorithm

RSA is a public-key cryptography technique which is centered mostly on purported complexity of factoring large prime numbers. An RSA user initiates and then presents, as their public key, the combination of two large prime numbers together with an auxiliary number. It is important to keep the prime numbers hidden. Anyone may use the public key to encode the data, conversely with current established approaches, if the public key is big enough, the message can only be decoded by someone who knows the prime numbers [10],[11]. The encryption and decryption procedure is described using the following pseudocode ().

Start
Generate two large and prime numbers $\boldsymbol{p}$ and $\boldsymbol{q}$, where $\boldsymbol{p} \cong \boldsymbol{q}$
Compute $\boldsymbol{n} = \boldsymbol{p} * \boldsymbol{q}$
Compute $\boldsymbol{\varphi(n)} = (\boldsymbol{p}-\mathbf{1}) * (\boldsymbol{q}-\mathbf{1})$
Select $\boldsymbol{e}$ such that for $\mathbf{1} < \boldsymbol{e} < \boldsymbol{\varphi(n)}$, $\boldsymbol{e}$ and $\boldsymbol{n}$ are coprime.
Compute $\boldsymbol{d}$ such that $(\boldsymbol{d} * \boldsymbol{e})\ \boldsymbol{mod}\ \boldsymbol{\varphi(n)} = \mathbf{1}$
stop

Key Generation
Public Key is $(\boldsymbol{e}, \boldsymbol{n})$
Private Key is $(\boldsymbol{d}, \boldsymbol{n})$
Encryption= $\boldsymbol{C = M^e mod\ n}$
Decryption= $\boldsymbol{M = C^d mod\ n}$

Users have to produce the pair of keys before encryption and decryption is completed and then those keys are used for encryption scheme.

The RSA offers a high security; thus, given the complexity and huge keys, an attacker should not be able to crack RSA by factoring. RSA is used for data encryption / decryption, which has the power to sign and/or validate digital data. RSA doesn't really need the use of a specific hash function, so the protection of the signatures and encryption depends in part on the preference of hashing algorithm used to measure the signatures [12]. The public key cryptography RSA is by far the most commonly used asymmetric cryptographic algorithm. It may be used in order to some anonymity and digital certificates and its protection is centered on the challenge of integral factorization being intractable.

III. The ElGamal Algorithm

The Elgamal algorithm was created by Taher Elgamal in the year 1984. It is an asymmetric cryptographic technique that is centered on key exchange. Elgamal encryption / decryption technique is based on the complexity of discrete logarithm where it is simple to lift digits to large powers however the inverse calculation of the discrete logarithm is far more complex to do. The dual important benefits of this approach are fast standardized security for long messages and data growth rate. ElGamal's biggest downside is its need for randomness and its slow process [13]. The encryption and decryption procedure of the ElGamal algorithm is described using the following pseudocode ().

Key Generation
Step 1: Start
Step 2: Generate big random prime p and θ of Z^*_p
Step 3: Select a random integer a, $1 \leq a \leq p-2$ and compute $\theta^a \pmod p$
Step 4: Compute the public key as (p, θ, θ^a) and the private key as a.

Encryption
Step 5: Obtain the Receiver public key (p, θ, θ^a).
Step 6: Denote the message as an integer m in the range $\{0,1\ldots\ldots, p-1\}$
Step 7: Select a random integer k, $2 \leq k \leq p-2$
Step 8: Compute $\gamma = \theta^k \pmod p$ and $\delta = m\,(\theta^a)k \pmod p$
Step 9: Encrypt the message $c = (\gamma, \delta)$

Decryption
Step 10: use private key a to compute $\gamma^{p-1-a} \pmod p$
Step 11: Decrypt the message m with $\gamma^a.\delta \pmod p$
Step 12: Stop

IV. Complexity of Cryptographic Algorithms

The study of the complexity of cryptographic algorithms is an important area of theoretical computing that helps to find or select the most efficient and effective algorithm to solve a combinatorial problem. The study on complexity of cryptographic algorithms improves on data security and privacy and also mode of data communication with the aim to understand the intrinsic difficulty of computational problems. The complexity of any cryptographic algorithm can be measured in terms of time, space or energy needed for it to encrypt and decrypt in a worst-case scenario. Thus, complexity describes the computational efforts needed for a crypto-system to encrypt and decrypt data. The algorithm's time complexity measures the amount of time the algorithm takes to execute as a function of the input length. The time complexity of cryptographic relies on factors such as hardware, operating system, processor among others. An algorithm's time complexity is commonly expressed through asymptotic notations: Big O which is denoted as O(n), Big Theta denoted as Θ(n) and Big Omega demoted as Ω(n). The time complexity analysis of an algorithm begins by trying to count the range of simple tasks to be performed while running the algorithm. Addition, subtraction, multiplication, division and compare are simple operations. The iteration below gives the description of operation of computing the time complexity of an algorithm.

```
for i : 1 to length of N
  if N[i] is equal to x
    return TRUE
  return FALSE
```

The total time of the algorithm depends on the length of the array N; if the length of the array increase the time of execution will also increase.

The space complexity of cryptographic algorithm is the measure of space (memory) it takes the algorithm to run as a function of its input-length. The complexity of space is determined by the size of any input. If for a given input size the complexity is taken as the maximum complexity over all inputs of that size, then the complexity is called the worst-case complexity. And if the complexity is taken as the average complexity over all inputs of a given size, then the complexity is called the expected complexity.

V. A Review of Existing Related Studies on the Complexity of RSA and ElGamal

Kayalvizhi et al [14] studied the energy complexity of RSA and ElGamal Algorithms for Wireless Sensor Networks. The study compared the performance of the RSA cryptographic algorithm with the ElGamal cryptographic algorithm by evaluating their energy efficiency and network lifetime. The study implemented both algorithms on a cluster network topology environment and compared the performance of the different network cluster. The computational experiments evaluated the complexity of RSA and ElGamal cryptographic algorithms in term of energy required and it was observed that the energy complexity of RSA was minimal when compared to ElGamal. Hence, RSA cryptographic algorithm requires less energy when it comes to protection of wireless communication. Therefore, RSA is said to be computationally efficient in terms of complexity of energy usage.

Qing and Yunfei [15] designed and implemented an efficient RSA variant with the aim to speed up RSA decryption process. In order to increase computational complexity, EAPRSA (Encrypt Assistant Multi-Power RSA) was introduced by moving some decryption arithmetic operations to encryption. Multi-Power RSA and RSA-S2 systems are integrated in the current new RSA variant. The multi-Power RSA used the N = p^2 q(b=3) formula modulus where p and q are n/3 bits respectively. The proposed technique reported that the space complexity of the decryption process of the proposed EAPRSA was a substantial improvement over the generic RSA.

Afolabi [16] performed a comparative performance assessment study on RSA cryptographic Algorithms. The study determined the complexity of the cryptographic algorithm in terms of Time, memory and output bytes. The study was conducted using a cryptographic technique on text file of different sizes. It was reported that RSA utilizes more time to encrypt and decrypt data and also uses more space while generating low output byte.

Chia et al., [17], studied time and space complexity of RSA and ElGamal cryptography algorithms. Encryption and decryption operations were achieved by modular exponentiation in RSA cryptographic algorithm. Also, fast modular exponentiation in RSA algorithms was considered of practical importance. By using the fast modular exponentiation of documenting the typical components in the folded sub strings, the performance of the binary algorithm could be improved and thus, reducing the computational complexity of modular exponentiation effectively.

In addition, Farah et al. [18], presented the implementation and comparative evaluation of techniques for variable text files: RSA, ElGamal, and Paillier; the encoding time, decoding time, encoded data rate and decoded data rate for each algorithm. The paper also identified and determined which algorithms perform better in terms of the time complexity. The experimental result showed that the time complexity of RSA during encryption process is computationally good and ElGamal did better with regard to decryption time complexity. Indeed, the overall complexity of RSA cryptographic algorithms in measure of the throughput was higher in the encryption process while ElGamal throughout complexity was higher during decryption process.

Vijayalakshmi and Bommanna [19] undertook a comparative analysis of RSA and ECC in Identity-Based Authenticated Modern Multiparty key Agreement scheme. The study implements two popular public key cryptographic algorithms and compared there performance by computing the processing time and memory size for the method of encryption and decryption. The study used different key sizes and variable text lengths to analyze the performance of both algorithms. The identity-based authenticated key agreement protocol showed that the protocol using ECC block cipher for user authentication offered significantly better performance in terms of memory complexity requirements and processing time complexity. Thus, their findings demonstrated ECC dominance over RSA in terms of time and memory complexities allocation for execution.

Furthermore, Annapoorna et al., [20] compared two asymmetric algorithms RSA and ElGamal for secure file transmission. The paper gave tabular reports of Key length value, algorithm sort, security threats, pace usability of the model, key usage, energy consumption and hardware / software implementation discrepancy between complexity of RSA and ElGamal cryptographic algorithms. The study analyzed the complexity of the cryptographic algorithm in terms of their security level. The study stated that ElGamal algorithm is more secure as compare to RSA cryptographic algorithm. Additionally, RSA algorithm was reported to perform poorly in terms of time complexity because it generates more than one public keys when encrypting and decrypting data.

Tin and Su [21], carried out a comparative Study of RSA and ElGamal on audio protection systems, based on the period of execution. The implemented encryption algorithms provide a secure communication over the internet and play a crucial role in efficient information security systems. The proposed system utilized two public key algorithms RSA and ElGamal algorithms to analyze their complexity of execution time on audio file. The study used audio (.mp3) file type with various file sizes to analyze the encryption and decryption of complexity time using C# programming language. The experimental results showed that RSA algorithm is faster than ElGamal algorithm in encrypting and decrypting audio file.

Andysah et al., [22], studied the Performance comparison of the public-key cryptographic algorithms RSA and ElGamal. RSA occured in the factorization of large primes whereas ElGamal occured in the computation of discrete logarithms. The study utilized public-key RSA and Elgamal encryption techniques for the encryption and decryption of a text file. The results of the experiments revealed that RSA algorithm outperformed the Elgamal algorithm in terms of time complexity.

More so, Kyaw, Kyaw and Nay [23], studied the time complexity of RSA public key encryption method and ElGamal public key encryption method. The study was concerned with the encryption process on text, image and audio data to obtain the encryption and decryption time of RSA and ElGamal cryptographic algorithms. The results of the experiment showed that RSA encryption and decryption time complexity was significantly better than those of the ElGamal cryptography algorithm.

Haval et al., [24], proposed and implemented a modified ElGamal cryptographic algorithm to increase the complexity of the algorithm in term of time, speed and reducing the expansion rate in the file size after encryption process. Some modification was performed on the traditional ElGamal cryptographic algorithm such as using addition operation instead of multiplication in the encryption process to decrease the file size. A comparative evaluation of the modified and traditional ElGamal was carried out on text data and the result showed that the time complexity performance of modified Elgamal was better than the traditional Elgamal.

VI. Conclusion

In this paper, survey of performance analysis of asymmetric cryptographic algorithms in particular RSA and ElGamal was carried out and analyzed. The experimental results of many of the papers reviewed showed that RSA cryptographic algorithm performs better as regards to energy usage, time complexity and space complexity of text, image and audio data during encryption process while some studies showed that ElGamal performs better in terms of time complexity during decryption process. This survey was limited to text, image and audio data. However, more performance metrics like video file and mixed data can be considered in further studies in order to widen the scope of this survey.

References

[1] M. Marwaha, R. Bedi, A. Singh, and T. Singh, “Comparative analysis of cryptographic algorithms”. Int J Adv Engg Tech/IV/III/July-Sept, 16, 18. 2013.

[2] S. Rani, and H. Kaur. “Technical review on symmetric and asymmetric cryptography algorithms”. International Journal of Advanced Research in Computer Science, 8(4). 2017.

[3] A. G. Walia. “Cryptography Algorithms: A Review.” International Journal of Engineering Development and Research. 2014.

[4] R. Sonia and K. Harpreet "Technical Review on symmetric and Asymmetric Cryptography Algorithms". International Journal of Advanced Research in Computer Science. www.ijarcs.info. 2017.

[5] F. Muhammad, J. Sapiee and H. Abdulkadir et. al., "A survey on Cryptographic Encryption Algorithms". International Journal of Advanced Computer Science and Applications, ol. 8, No. 11. pp 333-344. 2017.

[6] K. Howard. U.S. Patent No. 10,009,168. Washington, DC: U.S. Patent and Trademark Office. 2018.

[7] K. Sajay, S. Babu, and Y. Vijayalakshmi. "Enhancing the security of cloud data using hybrid encryption algorithm". Journal of Ambient Intelligence and Humanized Computing, 1-10. 2019.

[8] S. Ankush, A. Jyoti, D. Aarti and S. Pratibha. "Implementation & Analysis of RSA and ElGamal Algorithm". Asian J. of Adv. Basic Sci.: 2(3), 125-129. 2014. ISSN (Online): 2347 – 4114. www.ajabs.org.

[9] R. Sonia and K. Harpreet. "Technical Review on symmetric and Asymmetric Cryptography Algorithms". International Journal of Advanced Research in Computer Science. www.ijarcs.info. 2017.

[10] E. Arboleda, J. Balaba, and J. Espineli. "Chaotic rivest-shamir-adlerman algorithm with data encryption standard scheduling". Bulletin of Electrical Engineering and Informatics, 6(3), 219-227. 2017.

[11] N. Jayapandian, and A. Rahman. "Secure and efficient online data storage and sharing over cloud environment using probabilistic with homomorphic encryption". Cluster Computing, 20(2), 1561-1573. 2017.

[12] M. Baba-Ahmed, F. Benmansour, and A. Sedjelmaci. "A Cryptosystem Architecture and Design for Encrypted Data Transmissions". Electrotehnica, Electronica, Automatica, 67(2). 2019.

[13] M. Marwaha, R. Bedi, A. Singh, and T. Singh. "Comparative analysis of cryptographic algorithms". Int J Adv Engg Tech/IV/III/July-Sept, 16, 18. 2013.

[14] R. Kayalvizhi, M. Vijayalakshmi, and V. Vaidehi. "Energy Analysis of RSA and Elgamal Algorithms for Wireless sensor Networks". CNSA 2010. CCIS 89, pp. 172-180. 2010. Springer-Verlag Berlin Heidelberg.

[15] L. Qing, L. Yunfei, L. Hao. "On the Design and Implementation of an Efficient RSA Variant", Advanced Computer Theory and Engineering (ICACTE), 2010, pp.533-536.

[16] A. Afolabi, and O. Atanda. "Comparative analysis of some selected cryptographic algorithms". Computing Information Systems, Development Informatics and Allied Research Journal. Vol, 7, 41-52. 2016.

[17] L. Chia, and H. Chen. "Computational Complexity Theoretical Analyses on Cryptographic Algorithms for Computer Security Application", Innovations in Bio-Inspired computing and Applications(IBICA), 2012, pp. 307 – 311.

[18] S. Farah, Y. Javed, A. Shamim, and T. Nawaz. "An experimental study on performance evaluation of asymmetric encryption algorithms". In Recent Advances in Information Science, Proceeding of the 3rd European Conf. of Computer Science,(EECS-12). 2012. (pp. 121-124).

[19] P. Vijayalakshmi, and K. Bommanna. "Performance Analysis of RSA and ECC in Identity-Based Authenticated New Multiparty Key Agreement Protocol". International Conference on Computing, Communication and Applications (ICCCA), 22-24 Feb. 2012, pp 1-5.

[20] S. Annapoorna, S. Shravya, and K. Krithika. "A review on Asymmetric cryptography RSA and Elgamal Algorithms". International Journal of Innovation Research in Computer and Communication Engineering. Vol. 2, 2014. Special Issue 5.

[21] Z. Tin, and W. Su. "Performance Analysis of RSA and ElGamal for Audio Security". International Journal of Scientific Engineering and Technology Research. Volume.03, issue.11, June-2014, Pages: 2494-2498.

[22] P. Andysah, K. Elviwani and O. Boni. "Comparative Analysis of RSA and Elgamal Cryptographic Public-key Algorithms". 2018.

[23] M. Kyaw, S. Kyaw, and A. Nay. "Time Performance Analysis of RSA and Elgamal Public-Key Cryptosystems". International Journal of Trend in Scientific Research and Development (IJTSRD). Volume 3 Issue 6, October 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470.

[24] I. Haval, W. Hussein and M. Abduallah. "An efficient ElGamal cryptosystem scheme". International Journal of Computers and Applications, 2019. DOI: 10.1080/1206212X.2019.1678799.