

# An improved secured cloud data using dynamic rivest-shamir-adleman key

Ugbedejo Musa<sup>1</sup>, Marion O. Adebisi<sup>1</sup>, Francis Bukie Osang<sup>2</sup>, Abayomi Aduragba Adebisi<sup>3</sup>,  
Ayodele Ariyo Adebisi<sup>1,4</sup>

<sup>1</sup>Department of Computer Science, Landmark University, Omu-Aran, Nigeria

<sup>2</sup>Department of Computer Science, National Open University of Nigeria, Abuja, Nigeria

<sup>3</sup>Department of Electrical Power Engineering, Durban University of Technology, Durban, South Africa

<sup>4</sup>SDG 4, Landmark University, Omu-Aran, Nigeria

## Article Info

### Article history:

Received Nov 15, 2023

Revised Jul 19, 2023

Accepted Oct 19, 2023

### Keywords:

Data transmission

Encryption techniques

RSA algorithm

Secure cloud data

## ABSTRACT

Encryption methods had been widely used for secure data transmission and communication in both public and private organizations against intruders. Rivest-shamir-adleman (RSA) encryption algorithm is one of the most popular and efficient encryption schemes that has been in used for decades. Due to technological advancement and innovation, there is a threat to this algorithm. It is believed that introduction of quantum computer will break RSA algorithm easily. In view of this, it is pertinent to research into how RSA algorithm could be strengthened against all adversaries. This research aim at protecting client/server communication and file sharing by generating dynamic public and private keys. The proposed method was implemented in visual basic.net 2008. The result shows that dynamic keys do not affect the performance of the system and it is capable of protecting communication and file sharing between client/server. As the key generated keeps changing at an interval, it will difficult for most advance computer to factor any of the keys before another key is generated. This is the basis of the security of the proposed system.

This is an open access article under the [CC BY-SA](#) license.



## Corresponding Author:

Marion O. Adebisi

Department of Computer Science Landmark University

Omu-Aran, Kwara State, Nigeria

Email: marion.adebisi@lmu.edu.ng

## 1. INTRODUCTION

Data gathering, storage, and transmission are now made simple by technological innovation and advancement. Despite these technological advancements, it is becoming increasingly expensive for both individuals and organizations to store the huge amounts of data that are currently being gathered and analyzed. Recent technology advancement has provided solution to the problem of massive data storage and processing through cloud computing. The invention of cloud computing made it possible for individuals and organizations to store data utilizing the computational infrastructures of other people. Yuefa and Yaqiangm [1] in Ade and Mouratidis [2] defined cloud computing as a growing and well-liked approach of on-demand access to shared and dynamically customizable resources across a computer network. Cloud computing offers a platform where businesses and individuals may work without worrying about the price of computing resources or how to manage them. Dharani *et al.* [3] stated that numerous consumers are receiving new services using cloud computing based on their demand. These and many more features of cloud computing have made it well-liked and accepted by several businesses and people. Organizations such as Amazon, IBM, Microsoft, Alibaba and Heroku are well known providers of cloud computing services. These firms have invested a lot in cloud

infrastructures and are still doing so with consumers ranging from governmental entities that collect and distribute data globally [4]. Cloud service provider as mentioned above offer cloud services such as platform-as-a-service (PAAS), function-as-a-service (FAAS), software-as-a-service (SAAS) and infrastructure-as-a-service. PAAS is a cloud service that offers organizations and individuals to use their infrastructure instead of investing in its purchase. Also, IAAS is a service provided by cloud service provider where users can rent a software for use on demand without any investment in its purchase, development or management.

Cloud service has encouraged organizations, government of different nations and individual to embrace cloud computing due to its simplicity, cost effectiveness, high speed infrastructure and many other advantages it offers. But for consumers trying to acclimate to cloud computing platforms, privacy and security continue to be formidable obstacles. Businesses and individuals are concerned about the authenticity, security, and integrity of data kept in the cloud despite these fantastic qualities and the many benefits that come with its use. Over the time, researchers have offered numerous solutions, including encryption for data protection, digital signatures for authentication, and many other techniques to ameliorate these problems. Wu and Li [5] the fundamental principle of encryption is to convert the original message (plaintext) into an unintelligible format (ciphertext) using a specific method, preventing those who are unaware of the decryption technique from knowing the precise content of the data.

Numerous researchers have developed both symmetric (such as data encryption standard, advance encryption standard, blowfish, and triple data encryption standard (DES)) and asymmetric (such as rivest-shamir-adleman (RSA), elliptic curve, and elgamal key exchange) methods of data protection. In asymmetric data encryption, a single key is used by sender and receiver to encode and decode the message. The issue with symmetric encryption systems is that there is no safe way to exchange keys prior to communication, leaving the process open to hijacking. The answer to symmetric encryption was asymmetric encryption, sometimes known as public key cryptography. It enables communication process participants to exchange encryption keys in a secure manner without prior negotiation. RSA encryption algorithm is one of the various asymmetric encryption systems that aids in protecting cloud data and offering user authentication. Several researchers have reported successful break of low key and poorly implemented RSA algorithm [6], [7]. Aside this, it has been reported that the introduction of quantum computer [8], [9] may render RSA algorithm inefficient. A quantum algorithm and related circuit was introduced in [10]. The authors claimed that, when used with a quantum computer, will be able to break the RSA encryption scheme. In view of this, the goal of this research was to use a dynamic public key that changes periodically to offer a solution to these issues. With this, another key would have been generated before an eavesdropper could effectively break a key using a quantum computer, factorization algorithm, or any other method of speculating or anticipating the private key.

## 2. LITERATURE REVIEW

Several cryptographic algorithms have been proposed and implemented to secure data on the cloud and to control access to the data. Hash function, symmetric, and asymmetric are the three distinct types of cryptography algorithms [11]-[13]. Symmetric key encryption is a type of encryption in which the secret key used for encryption and decryption by the sender and the recipient is the same [14], [15]. There are good number of symmetric algorithm in existence today which include DES, advance encryption standard (AES), blowfish, triple DES, international encryption algorithm (IDEA), rivest series (RC4, RC5, RC6) are only a few of the symmetric algorithms in use today [16].

Asymmetric encryption encrypts and decrypts data using two different keys that are mathematically connected to one another [17]. The public key which is one of the keys, is made public and is used to encrypt data while the second key, the private key is made private and used to decrypt the encrypted data. In this method, the data is encrypted using the public key, which is made available to the public, and is then protected from assaults by being decrypted using a private key. The RSA and elliptic curve cryptography (ECC) algorithms are the two primary varieties of asymmetric encryption. RSA is one of the oldest and among the most used encryption algorithm. The algorithm was developed in 1977 by three Massachusetts Institute of Technology researchers, Okediran *et al.* [18].

RSA is effective because it chooses two distinct random primes of a specified size and multiplies them to produce another enormous number. For one to be able to break this algorithm, the task of finding the original primes in the multiplied behemoth seems very difficult even with the most advanced supercomputers. The original algorithm has been modified severally to strengthen its security [19]–[24].

Thangavel *et al.* [20] proposed a modification to the original RSA algorithm. The author used four prime numbers as against two proposed originally. The system is highly secure, according to the authors, because the public exponent  $n$  is created by multiplying two large prime numbers, and the values of the encryption (E) and decryption (D) keys are based on the product of four large prime numbers. The efficiency

of the novel method was shown by comparisons between it and the conventional RSA technique, the authors failed to adequately highlight the computing impact of the four prime numbers on the machine's resources.

An algorithm to enhance the original RSA algorithm was presented by [15]. The authors' strategy involved creation of more complicated key pair-a public and private key-so that an opponent could never deduce the private key from the public key. In order to achieve higher algorithm complexity, the proposed scheme used four random large prime numbers to generate public-private key pairs. It also applies XOR operation along with the more completed intermediate process in key-generation encryption and decryption phases, making it extremely difficult for third parties to attack thereby increasing security. The authors asserted that the new method improved the security of the conventional RSA techniques, but they also guaranteed that it would take some time before the algorithm could be cracked. Given that the algorithm might eventually malfunction, this remark suggests that additional research in this area is necessary.

Mohamed *et al.* [21] suggested a model that takes four prime numbers in an effort to make the RSA algorithm safer. Two public keys are supplied to the recipient in place of one public key. However, there is a speed issue, hence the Chinese remainder theorem is utilized in RSA decryption to increase speed. In the same year [22], proposed modified RSA algorithm using Mersenne numbers. The improved method was developed with medical ultrasound imaging equipment in mind, according to the authors. They demonstrated the improvement in security performances by comparing the encryption capabilities between two prime numbers and three prime numbers with a Mersenne number. The author established that the proposed algorithm's encryption capabilities are improved using the fixed Mersenne prime numbers when compared to the algorithm's use of factoring two prime numbers, which offers almost the same amount of time. Consequently, this newly created modified RSA technique may be useful for securely accessing patient ultrasound images both on and off site.

A modified RSA cryptosystem algorithm known as "Asymmetric key based cryptographic algorithm using four prime numbers to secure message communication (ACAFP)" was presented by [7]. The authors proposed generation of four small prime numbers. This makes the algorithm to use less memory space and consume little resources. This feature makes the algorithm perform in a memory and resource constraint resources. The performance evaluation conducted indicated that the new algorithm performs better than the classical algorithm. The algorithm was not subjected to various attack to establish its resistance to various known attacks.

The key length is one of the drawbacks with the RSA algorithm. Encryption and decryption take longer time with long key length. Meanwhile, the security of the RSA algorithm depends on the key size, to ameliorate this problem [23], presented an approach that algorithm reduces the time of encryption and decryption processes by dividing the file into blocks and enhances the strength of the algorithm by increasing the key size. This capability paves the way for consumers to conveniently store data in the cloud.

A hybrid data security technique that combines the classical RSA and Gaussian interpolation algorithms was introduced by Dawson *et al.* [24]. The integration boosts RSA's security to a fifth-degree level. The message's ASCII values are encrypted using the Gaussian first forward interpolation, while the second and third levels are encrypted and decrypted using the conventional RSA. Gaussian backward interpolation is used to decrypt the data once more in the last step. The factorization issue with the conventional RSA is addressed with the integration. Four different algorithms RSA, short range natural number (SRNN), two-key pair methods, and the suggested algorithm were used in the comparative analysis. According to the authors, they found that the suggested algorithm takes less time to encrypt and decrypt small amounts of data than it does to encode and decrypt large amounts of data.

The Euclidean approach was also used by [25] to modify the RSA algorithm and enhance its performance. The suggested algorithm outperforms the RSA algorithm using the greatest common divisor (GCD) technique in terms of speed, throughput, power consumption, and use of euclid-RSA. In order to achieve good performance and security, it will be necessary to speed up RSA in the future using decreased exponents and the new variant's most advantageous properties.

### 3. METHOD

#### 3.1. Rivest-shamir-adleman algorithm

Assuming two parties A and B want to exchange files or communicate securely, both parties will generate private key and public key. The procedure for generating public and private key is detailed in Algorithm 1. Each of the party will publish her key to the public folder. The sender will copy the public key of the receiver from the public folder and use it to encrypt the file. See Algorithm 2 for detailed procedure for encryption. Once the file is encrypted, the sender send will then send the encrypted file to the receiver. At the receiving end, the receiver use her private key to decrypt the file as described in Algorithm 3. The above steps describe the procedure for generating public and private key for RSA encryption algorithm. Once the keys are properly generated, the next step is to encrypt the message using the public key of the receiver.

### Algorithm 1. RSA key generation

The procedure for encryption and decryption using RSA algorithm is as stated.

- Step 1: generate two distinct prime numbers, p and q
- Step 2: compute the product of the two prime numbers i.e.  $n=pq$
- Step 3: compute the totient of n and q i.e. Let  $\phi(n) = (p-1)(q-1)$
- Step 4: pick an integer e such that  $1 < e < \phi(n)$ , and  $\text{gcd}(\phi(n), e) = 1$
- Step 5: compute the private exponent d such that  $d = e^{-1} \text{ mod } \phi(n)$
- Step 6: publish the public key as [e,n]
- Step 7: keep the secret key securely as [d,n]

### Algorithm 2. RSA encryption

The sender performs the steps to encrypt the data and send to the receiver:

- Step 1: the sender obtains the public key of the receiver
- Step 2: the message (plaintext) is converted to a positive integer value m,  $1 < m < n$ .
- Step 3: the sender encrypt message m as  $c = m^e \text{ mod } n$
- Step 4: in order to ascertain the originality of the message, the sender signed the document with his private public key.
- Step 5: the sender send the encrypted message to the receiver

### Algorithm 3. RSA decryption

The recipient receives an encrypted message, he thereafter apply his private key to decrypt the message back to the original state. The steps are the steps taken to decrypt the cipher text.

- Step 1: the recipient apply his private key [d,n] to the cipher text i.e  $M = C^d \text{ Mod } n$
- Step 2: convert the integer value obtained from step 1 to ASCII character
- Step 3: verify the integrity of the message by applying sender's public key
- Step 4: the converted value in step 2 is the original message

## 3.2. Proposed system architecture

The fundamental concept of this research is to continuously generate encryption and decryption keys at predetermined intervals. This implies that the public key folder will be updated continuously at a set interval. This is explained in the schematic diagram depicted in Figure 1.

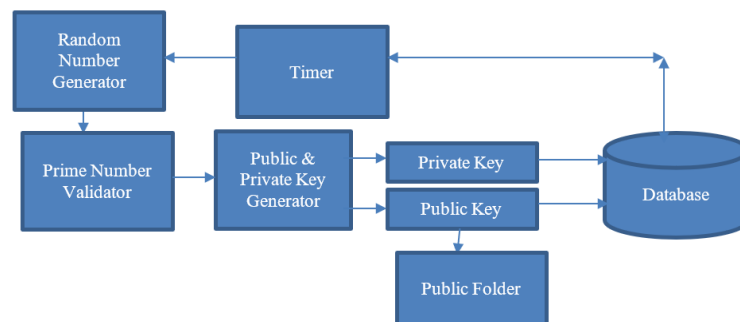


Figure 1. Proposed architecture for key generation

### 3.3. Random number generator and prime number validator

The random number generator module generates two large random numbers (p, q) as specified by RSA algorithm. The output of the random number generator is validated by the validator module. The validator module ensure that the number generated is a prime number. To validate any number generated, it must pass the primality test. The procedure for primality test is outlined in Algorithm 4. Once the two numbers p, q is validated, they are sent to key generator module where the public and private keys will be generated. In this case Fermat's test algorithm is used.

### 3.4. Key generator

Once the two prime number has been generated and validated, the next thing is to generate both the public and private key. Algorithm 1 is used to generate these keys. Once the keys has been generated, the public key is published at the public folder, a copy of it is kept in a database. Once the shelf life of the key expires, a new key will be generated and publish at the public folder. The public key is used for encryption while the private key is used for decryption. The private key is kept secretly. The purpose of the timer is to keep track of the shelf life of the keys. Once the shelf-life elapse, the timer triggers the process of generating another key.

**Algorithm 4. The procedure for primality test**

Input: number p, q  
 Output: Prime Number  
 Step 1: check if p,q generated is equal to 2 or 3  
 Step 2: if step 1 is TRUE return the number to key generator  
 Step 3: if step 1 is False, check if the number generated is  $\leq 1$  OR number (mod) 2=0 OR number (mod) 3= 0  
 Step 4: if step 3 condition in step 3 is false, go back to algorithm number generator to generate another number  
 Step 5: set up a loop: start=5, incremental=6, condition = counter \* counter  $\leq$  (p,q)  
 Step 6: if (p,q) % =0 or (p,q) % (counter+2) =0 generate another number  
 Step 7: if condition in step 6= true, pass the value to key generator module

**4. RESULTS AND DISCUSSION**

The proposed system was implemented on core i7 computer with Window 7 operating system. Visual basic 2008 was used to implement the algorithm proposed in this paper. Screenshot of the system implementation is depicted in Figure 2.

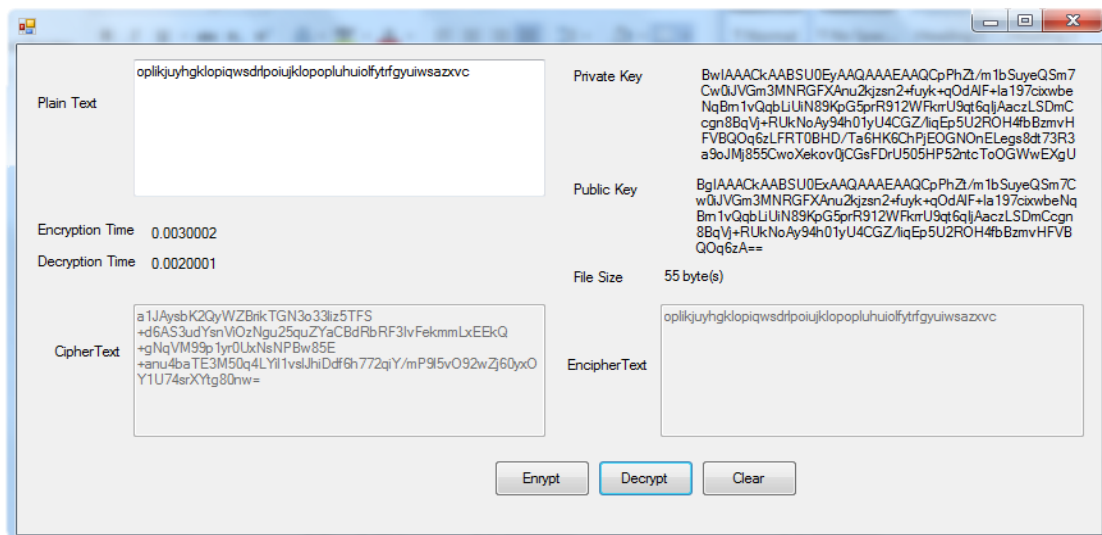


Figure 2. Screenshot of the system implementation

Table 1 shows the time for encryption and decryption in seconds with different file size using dynamic encryption and decryption keys. Using dynamic keys means different encryption and decryption keys are used to encrypt and decrypt each file. From the Table 1, it is observed that the decryption time is constant except the first decryption, while the encryption time decreases as the file size increases. The implication of this is that dynamic key generation does not have impact on the decryption time even when the file size is increased. Dynamic key generation will help in protecting cloud file even with the modern computer with high speed.

Table 1. Encryption and decryption time using dynamic keys

SN	File size	Encryption time (s)	Decryption time (s)
1	10 bytes	0.1840105	0.0030001
2	15 bytes	0.1630094	0.0020001
3	20 bytes	0.1930111	0.0020001
4	25 bytes	0.1940111	0.0020001
5	30 bytes	0.210012	0.0020001
6	35 bytes	0.1210069	0.0020001
7	40 bytes	0.1660095	0.0020001
8	45 bytes	0.1480085	0.0020001
9	50 bytes	0.1460084	0.0020001
10	55 bytes	0.141008	0.0020001

Figure 3 shows the time taken between the encryption and decryption time using dynamic keys. From the graph, it was observed that it takes a longer time to encrypt while decryption takes a shorter time. In Table 2, the encryption and decryption of time of various files with different file size are presented. All the files are encrypted and decrypted with static encryption and decryption keys. As observed in Table 1, the encryption time decreases as the file size decreases. The time difference is minimal. This means that the time taken to generate dynamic keys is negligible when compared to time taken when using static key. With this, there is no overhead cost generating dynamic keys.

Figure 4 shows the time taken to encrypt and decrypt file using static keys. It takes longer time to encrypt the first file while subsequent encryption takes less time. From the graph, the gap between the first encryption and subsequent encryption is high. The implication of this gap is the time taken to generate the keys before encryption.

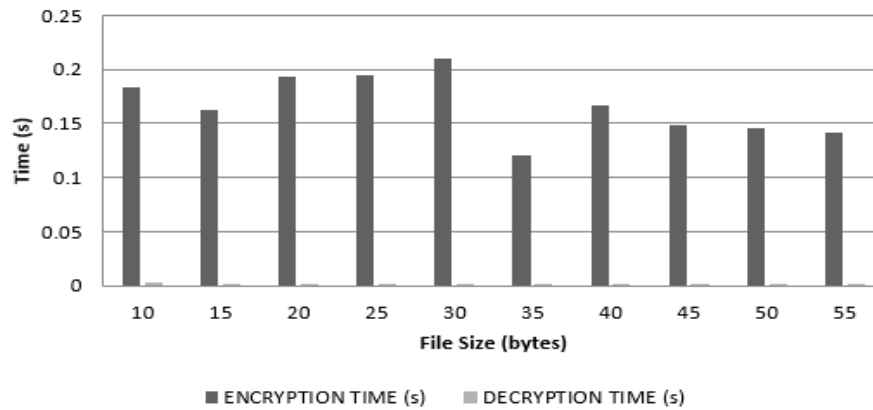


Figure 3. Encryption/decryption time (using dynamic keys)

Table 2. Encryption/decryption time using static keys

SN	File size	Encryption time (s)	Decryption time (s)
1	10 bytes	0.1180068	0.0020001
2	15 bytes	0.0030002	0.0030001
3	20 bytes	0.0030002	0.0020001
4	25 bytes	0.0030001	0.0020001
5	30 bytes	0.0020001	0.0020001
6	35 bytes	0.0030002	0.0020002
7	40 bytes	0.0030002	0.0020001
8	45 bytes	0.0030002	0.0020001
9	50 bytes	0.0030001	0.0020001
10	55 bytes	0.0030002	0.0020001

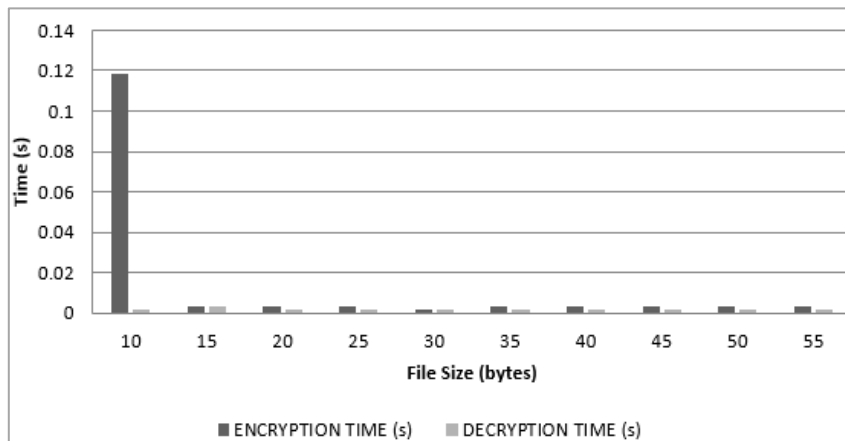


Figure 4. Encryption/decryption with static keys

**4.1. Dynamic keys vs static keys**

In Figure 5, the time taken to encrypt files using dynamic and static keys is presented. From the graph, the time taken for decryption is constant irrespective of the file size. Although, there is gap between encryption and decryption time. The gap is as a result of generating keys before encryption for each file. The graph representing the comparison between time taken to decrypt file using dynamic and static keys is presented in Figure 6. From the chart, it was observed that time taken to decrypt using dynamic and static keys is at par.

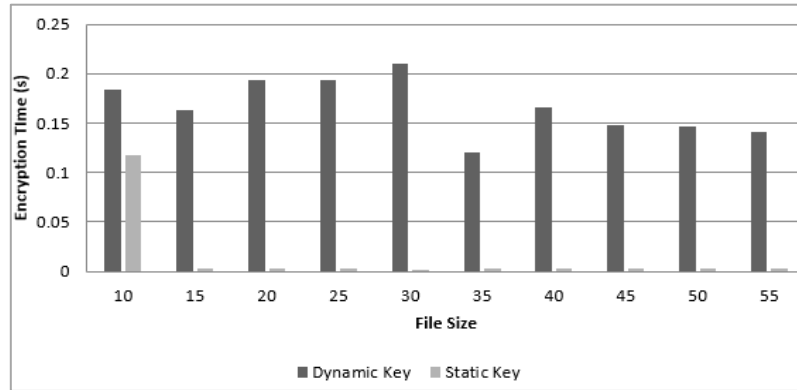


Figure 5. Encryption time (dynamic vs static key)

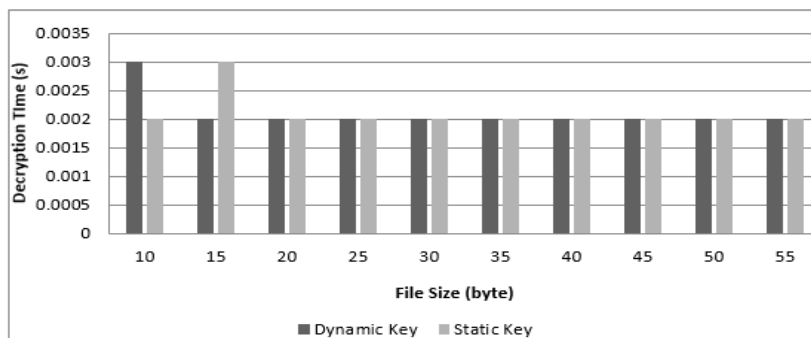


Figure 6. Decryption time (dynamic vs static key)

**5. CONCLUSION**

This paper present RSA algorithm where dynamic keys is being used to encrypt and decrypt files on the cloud server. This is one of the methods in which files can be protected from eavesdropper especially those who want to take advantage of super computer to break RSA algorithm. From our observation, there is no significant difference between time taken to generate static keys and dynamics keys. This means that there is no huge overhead cost on the resource of the machine. The performance of the proposed algorithm is the better than existing methods mentioned in the literature in terms of speed and resources usage.




**REFERENCES**

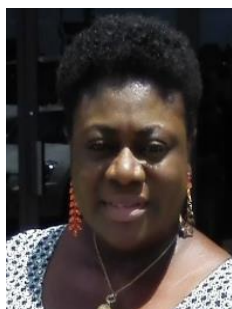
- [1] W. D. Yuefa and G. Yaqiangm, "Data security model for cloud computing", *In Proceedings of the 2009 International Workshop on Information Security and Application (IWISA 2009)*, Qingdao, China, 21–22 pp. 141–144, November 2009.
- [2] R. Adee and H. Mouratidis, "A dynamic four-step data security model for data in cloud computing based on cryptography and steganography," *Sensors*, vol. 22, no. 3, p. 1109, Feb. 2022, doi: 10.3390/s22031109.
- [3] A. P. S, N. Dharani, S. Pavithra, and R. Aiswarya, "Cloud data security using elliptic curve cryptography," *International Research Journal of Engineering and Technology(IRJET)*, vol. 4, no. 9, pp. 32–36, 2017.
- [4] A. Orobosade, T. Aderonke, A. Boniface, and A. J., "Cloud application security using hybrid encryption," *Communications on Applied Electronics*, vol. 7, no. 33, pp. 25–31, May 2020, doi: 10.5120/cae2020652866.
- [5] K. Wu and C. Li, "Application of symmetric encryption algorithm sensor in the research of college student security management system," *Journal of Sensors*, vol. 2022, pp. 1–7, Jul. 2022, doi: 10.1155/2022/3323547.
- [6] V.S. Rajput, J.M. Keller and P. Mor, "Secure cryptography with ngDH protocol along with RSA & AES algorithm," *International journal of scientific research in engineering and management*, vol. 06, no. 02, Feb. 2022, doi: 10.55041/IJSREM11503.




- [7] Ü. Çavuşoğlu, A. Akgül, A. Zengin, and I. Pehlivan, "The design and implementation of hybrid RSA algorithm using a novel chaos based RNG," *Chaos, Solitons & Fractals*, vol. 104, pp. 655–667, Nov. 2017, doi: 10.1016/j.chaos.2017.09.025.
- [8] C. M. Kota and C. Aissi, "Implementation of the RSA algorithm and its cryptanalysis," in *2002 GSW Proceedings, ASEE Conferences*, 2022, doi: 10.18260/1-2-620-38785.
- [9] Y. Dong, H. Liu, Y. Fu, and X. Che, "Improving the success rate of quantum algorithm attacking RSA encryption system," *Journal of Applied Physics*, vol. 134, no. 2, Jul. 2023, doi: 10.1063/5.0153709.
- [10] A. Joshi, R. Kumbhar, A. Mehta, V. Kosamkar and H. Shetty, "Breaking RSA encryption using quantum computer", *International Journal of Research and Analytical Reviews*, vol.9, no. 2, pp. 885-887, 2022.
- [11] S. Arora, "A review on various methods of cryptography for cyber security," *Journal of Algebraic Statistics*, vol. 13, no. 3, pp. 5016–5024, 2022.
- [12] V.L. B and M. De, "Cryptography Techniques for Software Security," *Software Security*, p. 6, 2022.
- [13] B. S. Rawal, G. Manogaran, and A. Peter, *Cybersecurity and Identity Access Management*. Springer Singapore, 2023, pp. 129-139, doi: 10.1007/978-981-19-2658-7.
- [14] S. Padhiar and K. H. Mori, "A Comparative study on symmetric and asymmetric key encryption techniques," 2022, pp. 132–144. doi: 10.4018/978-1-7998-6988-7.ch008.
- [15] R. Imam, F. Anwer, and M. Nadeem, "An effective and enhanced RSA based public key encryption scheme (XRSA)," *International Journal of Information Technology*, vol. 14, no. 5, pp. 2645–2656, Aug. 2022, doi: 10.1007/s41870-022-00993-y.
- [16] B. Abdurakhimov, I. Boykuziyev, and J. Abdurazzokov, "Encryption systems and the history of their development," *InterConf*, vol. 18, no. 95, pp. 768–776, Jan. 2022, doi: 10.51582/interconf.19-20.01.2022.085.
- [17] H. W. Dhany, F. Izhari, H. Fahmi, M. Tulus, and M. Sutarmar, "Encryption and decryption using password based encryption, MD5, and DES," in *Proceedings of the International Conference on Public Policy, Social Computing and Development 2017 (ICOPOSDev 2017)*, Paris, France: Atlantis Press, 2018, pp. 278-283, doi: 10.2991/icosposdev-17.2018.57.
- [18] O. O. Okediran, A. A. Sijuade, and W. B. Wahab, "Secure electronic voting using a hybrid cryptosystem and steganography," *Journal of Advances in Mathematics and Computer Science*, pp. 1–26, 2019.
- [19] A. H. Lone and A. Khalique, "Generalized RSA using 2 k prime numbers with secure key generation," *Security and Communication Networks*, vol. 9, no. 17, pp. 4443–4450, Nov. 2016, doi: 10.1002/sec.1619.
- [20] M. Thangavel, P. Varalakshmi, M. Murralli, and K. Nithya, "An enhanced and secured RSA Key generation scheme (ESRKGs)," *Journal of Information Security and Applications*, vol. 20, pp. 3–10, Feb. 2015, doi: 10.1016/j.jisa.2014.10.004.
- [21] H. Mohamed, H. A. Abd, R. M. Hussien, R. S. Abdeldaym, A. Elkader, and R. Hussein, "Modified RSA algorithm using two public key and chinese remainder theorem," *International journal of Electronics and Information Engineering*, vol. 10, no. 1, pp. 51–64, 2019, [Online]. Available: <https://www.researchgate.net/publication/334603666>
- [22] S.-H. Shin, W. S. Yoo, and H. Choi, "Development of modified RSA algorithm using fixed mersenne prime numbers for medical ultrasound imaging instrumentation," *Computer Assisted Surgery*, vol. 24, no. sup2, pp. 73–78, Oct. 2019, doi: 10.1080/24699322.2019.1649070.
- [23] I. G. Amalarethinam and H. M. Leena, "Enhanced RSA algorithm with varying key sizes for data security in cloud," in *2017 World Congress on Computing and Communication Technologies (WCCCT)*, IEEE, Feb. 2017, pp. 172–175. doi: 10.1109/WCCCT.2016.50.
- [24] J. K. Dawson, F. Twum, J. B. H. Acquah, Y. M. Missah, and B. B. K. Ayawli, "An enhanced RSA algorithm using Gaussian interpolation formula," *International Journal of Computer Aided Engineering and Technology*, vol. 16, no. 4, p. 534, 2022, doi: 10.1504/IJCAET.2022.123996.
- [25] R. F. S. L. Et.al, "Improvement of RSA algorithm using euclidean technique," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 3, pp. 4694–4700, Apr. 2021, doi: 10.17762/turcomat.v12i3.1889.

## BIOGRAPHIES OF AUTHORS






**Ugbedejo Musa**    received the M.Sc., degree in Management Information System (MIS) from Covenant University, Ota, Ogun State, Nigeria. Her research areas are data science, data analysis and project management. She is a lecturer with The Federal Polytechnic, Idah, Kogi State. She is currently a Ph.D., student in the Department of Computer Science, Colledge of Pure and Applied Science, Landmark University, Omu-Aran, Kwara State. She can be contact at email: [musa.ugbedejo@yahoo.com](mailto:musa.ugbedejo@yahoo.com).






**Marion O. Adebisi**    received a B.Sc. degree in Computer Science from University of Ilorin, Kwara State, Nigeria in 2000. Her MSc. and Ph.D., degree also in Computer Science, Bioinformatics Option from Covenant University, Ota, Nigeria in 2008 and 2014 respectively. She is associate professor in Computer Science Department of Landmark University and Covenant University. Her research interests include bioinformatics, genomics, proteomics, and Organism's inter-pathway analysis. She has published widely in local and international reputable journals. He is a member of Nigerian Computer Society (NCS), the Computer Registration Council of Nigeria (CPN) and IEEE member. She can be contact at email: [marion.adebisi@lmu.edu.ng](mailto:marion.adebisi@lmu.edu.ng).








**Francis Bukie Osang**    had a B.Sc., degree in Computer Science from the University of Calabar in 2002, M.Sc., in Information Technology from the National Open University of Nigeria in 2010 and Doctor of Philosophy Ph.D., in ICT from the ICT University in Baton Rouge, Louisiana, USA in 2014. He is a member, International Association of Computer Science and Information Technology, Member, Computer Registration Council of Nigeria. He is currently the Head of the Department of Computer Science, National Open University of Nigeria. He is a research scholar with over 16 publications published in international Journals and conference proceedings. His current research interest is in behavioral information systems. He can be contact at email: fosang@noun.edu.ng.



**Abayomi Aduragba Adebisi**    is a lecturer at the Durban University of Technology, South Africa. He received his Higher National Diploma in Electrical and Electronics Engineering Technology at Kwara State Polytechnic, Ilorin, Nigeria, in 2006, and his Master of Engineering and Doctor of Engineering degrees in Electrical Engineering from the Durban University of Technology, South Africa in 2017 and 2021 respectively. He worked in the Department of Electrical and Information Engineering (EIE) at Covenant University, Nigeria, and currently lectures at Durban University of Technology, South Africa. He has published Scopus-indexed journal articles and conference papers. His areas of research interest are power systems, renewable energy, optimization of renewable energy systems and engineering education. He supervises postgraduate degree students. He can be contact at email: abayomia@dut.ac.za.



**Professor Ayodele Ariyo Adebisi**    is a Faculty and Former Head of Department of Computer and Information Sciences, Covenant University, Ota Nigeria. He is currently the Dean, College of Pure and Applied Sciences at Landmark University, Omu-Aran, Nigeria, a sister University to Covenant University. He holds a B.Sc. degree in Computer Science and MBA degree from University of Ilorin, Ilorin, Nigeria in 1996 and 2000 respectively. He had his M.Sc. and Ph.D. degree in Management Information System (MIS) from Covenant University, Nigeria in 2006 and 2012. His research interests include, application of soft computing techniques in solving real life problems, software engineering and information system research. He has successfully mentored and supervised several postgraduate students at Masters and Ph.D. level. He has published widely in local and international reputable journals. He is a member of Nigerian Computer Society (NCS), the Computer Registration Council of Nigeria (CPN) and IEEE member. He can be contacted at email: ayo.adebisi@lmu.edu.ng.