Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

SECURE RSA CRYPTOSYSTEM BASED ON MULTIPLE KEYS

Ali Najam Mahawash Al-Jubouri [1], Dr. Rana Jumaa Surayh Al-Janabi [2],

[1] University of Al-Qadisiyah, Department of Computer Science, Qadisiyah, Iraq

Com.post13@qu.edu.iq, alianjeem@gmail.com

[2] University of Al-Qadisiyah, Department of Computer Science, Qadisiyah, Iraq

rana.aljanaby@qu.edu.iq

ARTICLEINFO

Article history:
Received: 06 /06/2021
Rrevised form: 25 /06/2021
Accepted : 30 /06/2021
Available online: 04 /08/2021

Keywords:

Block cipher, Security,

Multiple key RSA,

Random number Generator,

Segments key, seed key.

ABSTRACT

Information and communication technology are spreading very rapidly in terms of information exchange over the Internet, and this information is vulnerable to threats by hackers. Information security is mainly achieved by using encryption techniques to protect it when it is transmitted over an unsecured channel. In this paper, a modified encryption system for the RSA algorithm is presented using a fixed encryption key size and divide that key into specific sections, to encrypt and decrypt blocks using multiple public and private keys. The encryption process can be done for each block by choosing different keys according to the random generator key (seed key) and encrypt each block with these different keys. Through the random arrangement of blocks and the properties of a modified cipher block in the RSA algorithm within the proposed model, to increase security at the expense of time, the use of large keys in the RSA algorithm is very slow since small RSA keys are vulnerable to factorization attacks. To overcome that problem, we increase complexity and use larger block sizes without sacrificing speed, and compare them with the original RSA algorithm. As a result, this method is more efficient, secured, and not easily breakable.

MSC. 41A25; 41A35; 41A36

DOI : https://doi.org/10.29304/jqcm.2021.13.3.824

∗Corresponding author: *Ali Najam Mahawash Al-Jubouri.*

Email addresses: *Com.post13@qu.edu.iq.*

Communicated by*: Dr. Rana Jumaa Surayh aljanabi.*

1. INTRODUCTION

Encryption science is the technique of confidential writing and also to ensure the exchange of information over the Internet network between two parties remains secure, and it is the best technique for obtaining confidentiality and security of information that includes concealing and protecting data by way of translating a file into a type that is entirely different from the original version, and it and hard for a hacker to decipher it This helps individuals to protect the transfer of confidential information and has a high level of integrity[1]. Its main goal is to keep the data safe from unauthorized access, an encryption system usually includes operations translating the plain text into ciphertext, and the number of hidden keys used.

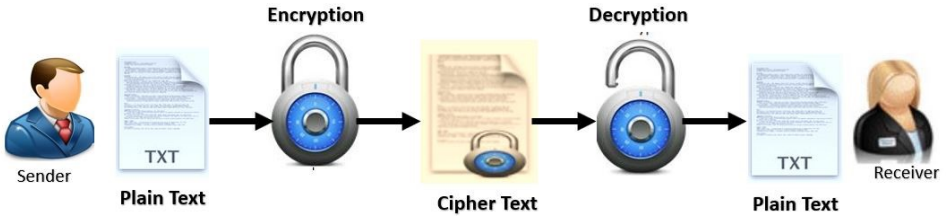

Fig 1: Encryption and Decryption

The cypher is divided into two broad classes[2], (i) Symmetric cryptography is one and (ii) asymmetric cryptography is the other. In symmetric encryption (Shared Key Ciphering System) one key is used in both the encryption process and the decryption process[1][3]. more ever this approach to encryption includes many well-known algorithms such as data encryption standard (DES) and Advanced Encryption Standard (AES).

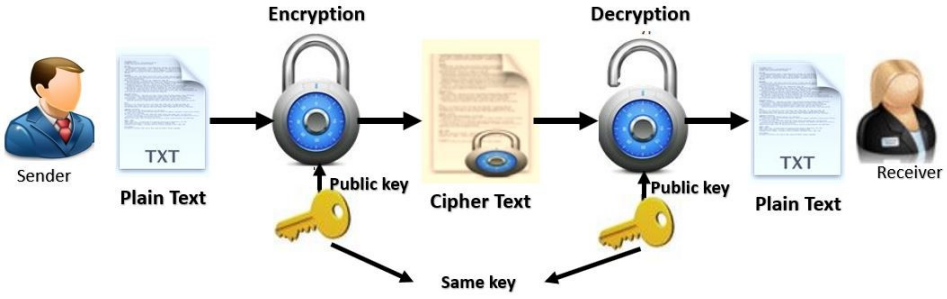

Fig 2: Private Key Encryption

While the other type of asymmetric encryption (Public Key Ciphering System) applies two different approaches to the keys used, such as a public key to encrypt the original text and a private key to decrypt the encrypted text. Many algorithms use asymmetric encryption, such as the RSA (Rivest-Shamir-Adleman) algorithm and algorithm (Al Gamal).

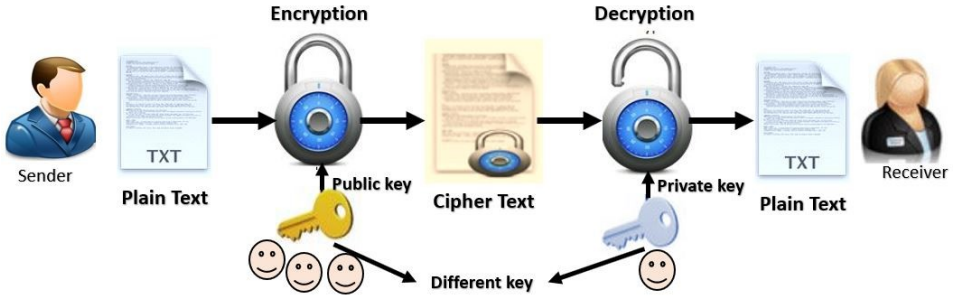

Fig 3: Public Key Encryption

The final approach is the asymmetric type of cryptography which is a revolution in cryptography[4]. One of the most potent algorithms that can be used for both encryption and digital signatures is considered to be RSA [5]. Currently, encryption algorithms provide a high degree of confidentiality by including all important data for any company or group of individuals until the encryption is implemented to access the information in a restricted and reliable manner and to ensure its correctness [6]. In this paper, we present a method for encrypting messages using the original RSA cryptosystem, as well as encrypting the same messages using the New RSA algorithm and comparing the results, design an application to encrypt and decrypt message using the C#.Net programming language.

2. Overview

In this research, we investigate the impact of utilizing large integers in the RSA method to improve security but at the expense of time and complexity. Although employing larger than 512-bit keys in RSA improves security, it also lengthens the time it takes to encrypt and decrypt data. We also offer a methodology that employs several keys public and private to provide stronger encryption without lengthening the time it takes to perform the transformation.

3. Methodology RSA _Algorithm

The best known and most commonly used public-key scheme at the present is RSA. It was invented by three cryptologists in 1978 by Ronald Rivest, Adi Shamir, and Leonard Adleman, of MIT, RSA is used for privacy, for the authentication of digital records, for payment systems for electronic credit and debit cards, and business systems such as web servers and browsers Web Traffic Safe [7]. It consists of three stages: primary key generation, encryption, and decryption stage, there are two different keys to the RSA cryptosystem, public and private key. It is possible to officially disclose the public key and is used for encrypting plaintext or color images, But the hidden key is used to decode the cipher code [8]. An algorithm RSA is a public key cipher system that uses number theory, so its security depends on the difficulty of the analysis for large prime numbers. which is a mathematically known problem for which there is no solution[9]. The private key is linked mathematically to the public key in public encryption. To create a private key, you can attack a public-key scheme. Usually, data encryption and decryption are complicated problems with mathematical equations due to the constraints needed on computer resources in particular, processor and memory, When the value of the key is big enough that it becomes much harder to know the common factors of the main number, RSA comes to be more protected [10]. The RSA algorithm is a cryptographic technique that Today's most secure algorithm for communicating between an issuer and a recipient[11][12].

3.1 Standard RSA algorithm

Refer to the algorithm's original search address for more detailed information on the basic RSA algorithm, including how it operates, the way to construct it, and deal with data [13].

4. Related works

Many researchers have been introduced several ideas to improve the efficiency of the RSA algorithm, this section contains the results of RSA cryptosystem researchers' efforts, the debate is based on the RSA algorithm modification. **Narander et al**., (2016), proposed a method modified RSA algorithm, that is based on n-prime numbers. this strategy employs prime numbers Because big primes numbers are difficult to factorize, they have issues with security, performance, and efficiency[14]. **The study**[15] used a general 3-key RSA technique, but to tackle the problem of server upload data the study still has to verify the safety level. **The MA Islam**[16] proposed in 2018, Instead of two prime numbers, this study uses "n" distinct prime numbers, which increases the attacking time to find the big prime number, Modified RSA (MRSA) needs a longer key generation period since it is based on a large factor value "N" The longer it takes to generate a key, the longer it takes to break the system, The disadvantage of this implementation is that takes time longer. **A research** paper has been published in 2015 for (I Jahan, M Asif, LJ Rozario) introducing a new algorithm This study focuses on number theory and public-key cryptosystems, to make the RSA cryptosystem more secure. To encrypt the message, the RSA cryptosystem generates a single public key. While it is difficult to find the factors of n and obtain p and q, two large prime numbers, our proposed algorithm encryption keys are sent separately rather than all at once [17]. The proposed RSA algorithm is used in a context that demands high security while still being slow. **The study** [18] R. Felista Sugirtha (2021) proposed the RSA method is changed to increase its performance using the Euclidean approach. the suggested algorithm exhibits its higher performance. In comparison with the RSA algorithm with the GCD approach, this article clearly reveals that Euclid-RSA is more effective than the RSA algorithm using the GCD technology in terms of Avalanche effects, encryption, decryption, throughput, and power consumption. The RSA with lowered exponents will be accelerated in the future and the most favorable parameters for good performance. They all have a problem with security, complexity, and time. As a result, we will attempt to tackle this problem using the proposed approach.

5. Proposed model

In previous research, there are problems with complexity and time. when complexity grows, the amount of time also grows. The proposed algorithm would not change the basic structure of the RSA algorithm but would lead to higher complexity by using many public and private keys without time-consuming. Where the encryption and decryption block is set to a fixed length of 2048 bits, the 2048-bit block is then segmented into 128, 256, 512, or 1024-bit. segments then the RSA key pairs (public and private) are generated to the segment numbers. Each block is encrypted from the message with a randomly generated different key; we use a random generator (seed key) for the order random sequence for the keys and maintain the order during decryption. Later during decryption, the seed is used to recover the same random sequence that was used in encryption. Using large keys leads to better encryption but at the cost of time; with the proposed problem is solved, we will increase the complexity, making it more challenging for the attacker to break the encryption for each segment. And if the attacker tries to factorize (n) using an integer factorization algorithm. He cannot determine the order in which they are used without obtaining the Seed Key. According to this model, it has additional advantages, but the protection remains highly related to the RSA problem's hardness.

Algorithm For key generation:

Input: Segmentation Sizes, Seed key for the sequence generator(seed).

Output: Private Keys, Public Keys.

1- The size of the block is (2048).
2- compute the total number of segments (N = size of a block/segment size).
3- Generate the size of the RSA key pair for each (segment Size).
4- The public key ([seg 1 key, seg 2 key, ... seg N key, N]).
5- The private key ([seg 1 key, seg 2 key, ... seg N key, N]).
6- The seed key is pre-shared by safe means between the sender and the recipient.

Algorithm of Encryption:

Input: Seed Key, Public Keys, bits to encrypt.

1- we extract the Public key of RSA Keys and (number of segments).
2- Generator = Start the sequence generator with (Seed Key)
3- While the Bits To Encrypt field is not empty:
   a- Block = cut a block from 2048 bits To Encrypt.
   b- Add padding if necessary.
   c- Segment List = Split Block to the desired number of segments (N).
   d- For any segment in the list of segments:
      i. **Sequence** = **Generator**. (0, **N**-1).
      ii. Encrypt **segment** with RSA using List of Public RSA Keys.
      iii. **Next** segment.
   e- Encrypted **Block** Append to Encrypted Bits.
   f- **Go** to step2.
4- **Output: Encrypted Bits.**

Algorithm of Decryption:

Input: Seed Key, **Private key**, **Bits to Decrypt**.

1- we extract the Private key of RSA Keys and (number of segments).
2- **Generator** = Start the sequence generator with (Seed Key).
3- While the Bits To Encrypt field is not empty:
   a- Block = cut a block from 2048 bits To Decrypt.
   b- Decrypted Block = Create a 2048-bit empty block.
   c- Segment List = Split Block to the desired number of segments (N).
   d- For any segment in the list of segments:
      i. **Sequence** = **Generator**. (0, **N**-1).
      ii. Encrypt **segment** with RSA using List of Private RSA Keys.
      iii. **Next** segment.
   e- If **padding** is detected, **remove** it.
   f- **Go** to step2.
4- Output: Decrypted block**.**

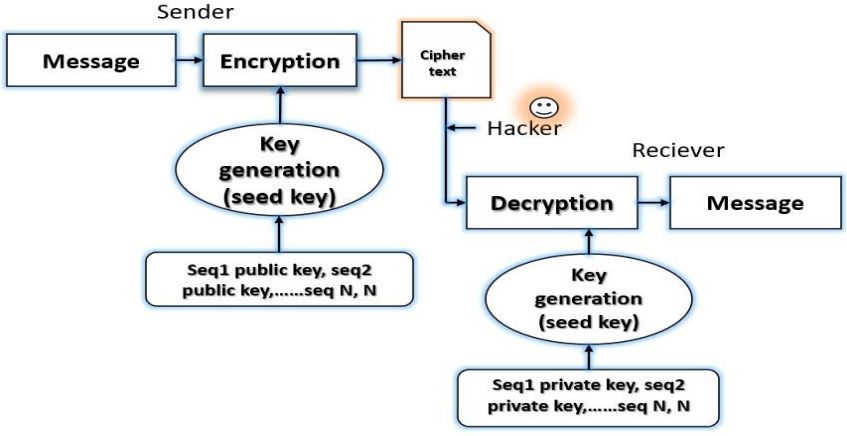

Figure 4. Flow chart of Modified RSA (New-RSA) Algorithm.

5.1 Implementation and Result

The RSA algorithm is cryptographic technique security is due to the fact that it is a cryptographic system that uses two keys, one public and one private, both of which employ large prime numbers (usually 2048 bits as suggested). demands a high level of computing efforts to decode it via factorization. In this section the experiment is carried out in the testing is done in a system with the following specifications: CPU: 4700HQ, RAM 16GB DDR3, OS Windows 10 Home v2004, we're using the C#.Net programming language. All of the work is done in a single thread, no parallel processing is used to speed up the encryption or decryption. The modified RSA algorithm has different multiple keys and important levels of security and speed. Decomposing the prime numbers into factors becomes more difficult as the length of the keys increases. As a result, the longer the private key increased making it more difficult to factors the key.

5.2 Performance Analysis

The proposed algorithm is examined in different input bit sizes. Table 1 shows the performance of the original RSA algorithm. While table 2 shows the performance of the Modified RSA method in terms of encryption time and decryption. Comparing the table1 and table2, it can be concluded that the time of encryption and decryption of Modified RSA is lower than that of the original RSA algorithm.

sequence	Algorithm	Encryption Time	Decryption Time
1-	**RSA 128**	**4419 ms**	**41752 ms**
2-	**RSA 256**	**5392 ms**	**116213 ms**
3-	**RSA 512**	**8397 ms**	**365841 ms**
4-	**RSA 1024**	**14302 ms**	**1283359 ms**
5-	**RSA 2048**	**26635 ms**	**4749015 ms**

Table 1. Analyzing time for original RSA algorithm.

sequence	Algorithm	Encryption Time	Decryption Time
1-	**Proposed (16 Segments)**	**4258 ms**	**38145 ms**
2-	**Proposed (8 Segments)**	**5368 ms**	**109678 ms**
3-	**Proposed (4 Segments)**	**8367 ms**	**362710 ms**
4-	**Proposed (2 Segments**	**14340 ms**	**1280451 ms**

Table 2. Analyzing time for RSA NEW algorithm.

There is an almost different amount of time-consuming two algorithms. However, as bit length increases, the discrepancy between curves grows rapidly.

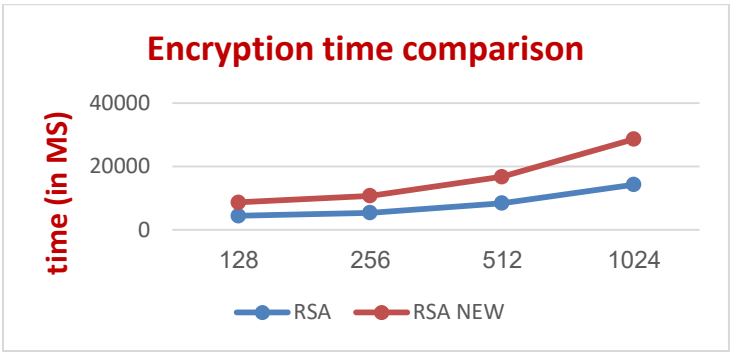

Figure 5. Encryption time comparison

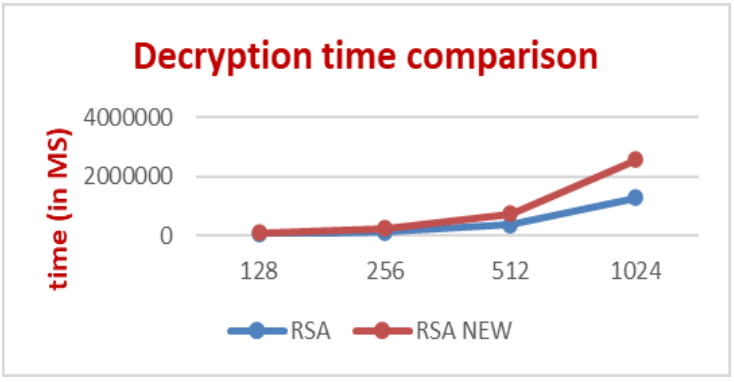

Figure 6. Decryption time comparison

We used the standard RSA algorithm as a guide, with main sizes of 128, 256, 512, 1024, and 2048 bits, and we validated the proposed structure with two segments of 1024 bits each, four segments of 512 bits each, eight segments of 256 bits each, and finally 16 segments of 128 bits each. With 512-bit input primes, modified RSA takes 8367 milliseconds to encrypt, while RSA takes 8397 milliseconds. A comparison between encryption and decryption time between RSA and the proposed Modified RSA system is presented in Figure 5 and Figure 6. It shows that RSA and Modified RSA take approximately different times for the small bits of prime values. the difference between curves increases significantly as bit length increases.

6. *Randomness Testing to assess security*

The National Institute of Standards and Technologies (NIST) randomity checks are a necessary way of randomity checking. Enhancement of the performance of RSA and increase the security and according to Randomness testing (using NIST statistical tests) has developed a package of 15 statistical tests to assure the randomness of a cryptography algorithm, NIST randomness tests prerequisite for the secure use of keys[19]. These tests focus on a range of non-randomness kinds which can exist in a sequence, the fifteen tests:

1) The Frequency (Mon obit) test;
2) Frequency test within a block;
3) The Run test;
4) Tests for the longest-Run-of-ones in a block;
5) The Binary matrix rank test;
6) The Discrete Fourier transform test;
7) The Non-overlapping template matching test;
8) The Overlapping template matching test;
9) Maurer's "Universal statistical" test;
10) The Linear complexity test;
11) The Serial test;
12) The Approximate entropy test;
13) The cumulative sums test;
14) The Random excursions test;
15) The Random excursions variant test.

After NIST tests have been done there are certain tests with values to P-value, as shown in Table3

RSA NEW			
Approximate Entropy P-Value	0.460603163146689	Random	PASS
Block Frequency P-Value	0.43521410993043	Random	PASS
Frequency P-Value	0.926881758866827	Random	PASS
Longest Run P-Value	0.793856434858983	Random	PASS
NonOver lapping Template P-Value	0.496420336555827	Random	PASS
Overlapping Template P-Value	0.438011375461119	Random	PASS

Since the P-value is ≥ 0.01, accept the sequence as random[19].

7. CONCLUSION AND FUTURE WORK

Proposed RSA cryptosystem based on several public and private keys for encryption and decryption processes to make it more complex. with increase encrypt and decrypt speed, to improve security, the proposed structure employs a fixed block size of 2048 bits and encrypts each segment with a randomly generated RSA key. Each part of the message is encrypted with different multiple keys. the structure proposed combines some properties of block cipher techniques and random block ordering with the traditional RSA algorithm to overcome the problem of increasing security at the expense of the time that is present, It is difficult to crack the modified algorithm since it requires more than one key to encrypt the message randomly, the key sequence is one of the keys that must be sent to the recipient secretly which is necessary for decryption. Compared to the regular RSA method, this model gives more security and increases message complexity, and gives us the confidence to send messages across non-secure channel**. In our future work**, more analysis can be done on the multiple keys (public and private) method for higher key sizes. and working on the various RSA attacks in order to make the RSA cryptosystem more secure and increase its speed.

References

[1] M. Barakat, C. Eder, and T. Hanke, “An introduction to cryptography,” *Timo Hanke RWTH Aachen Univ.*, pp. 1–145, 2018.

[2] A. Kahate, *Cryptography and network security*. Tata McGraw-Hill Education, 2013.

[3] G. J. Simmons, “Symmetric and asymmetric encryption,” *ACM Comput. Surv.*, vol. 11, no. 4, pp. 305–330, 1979.

[4] U. Rathod, S. Sreenivas, and B. R. Chandavarkar, “Comparative Study Between RSA Algorithm and Its Variants: Inception to Date,” in *ICCCE 2020*, Springer, 2021, pp. 139–149.

[5] P. Singh and R. K. Chauhan, “A Survey on Comparisons of Cryptographic Algorithms Using Certain Parameters in WSN.,” *Int. J. Electr. Comput. Eng.*, vol. 7, no. 4, 2017.

[6] R. S. Jamgekar and G. S. Joshi, “File encryption and decryption using secure RSA,” *Int. J. Emerg. Sci. Eng.*, vol. 1, no. 4, pp. 11–14, 2013.

[7] D. Boneh, “Twenty years of attacks on the RSA cryptosystem,” *Not. AMS*, vol. 46, no. 2, pp. 203–213, 1999.

[8] J. Hoffstein, J. Pipher, J. H. Silverman, and J. H. Silverman, *An introduction to mathematical cryptography*, vol. 1. Springer, 2008.

[9] G. Ye, K. Jiao, and X. Huang, "Quantum logistic image encryption algorithm based on SHA-3 and RSA," *Nonlinear Dyn.*, vol. 104, no. 3, pp. 2807–2827, 2021.

[10] T. S. Obaid, "Study A Public Key in RSA Algorithm," *Eur. J. Eng. Technol. Res.*, vol. 5, no. 4, pp. 395–398, 2020.

[11] N. D. Pantoja, A. F. Jiménez, S. A. Donado, and K. Márceles, "Cryptanalysis of the RSA Algorithm Under a System Distributed Using SBC Devices," in *International Congress of Telematics and Computing*, 2018, pp. 3–12.

[12] W. Susilo, J. Tonien, and G. Yang, "Divide and capture: An improved cryptanalysis of the encryption standard algorithm RSA," *Comput. Stand. Interfaces*, vol. 74, p. 103470, 2021.

[13] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[14] N. Kumar and P. Chaudhary, "Implementation of modified RSA cryptosystem for data encryption and decryption based on n prime number and bit stuffing," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, 2016, pp. 1–6.

[15] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of IoT and cloud computing," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 964–975, 2018.

[16] M. A. Islam, M. A. Islam, N. Islam, and B. Shabnam, "A modified and secured RSA public key cryptosystem based on 'n' prime numbers," *J. Comput. Commun.*, vol. 6, no. 03, p. 78, 2018.

[17] I. Jahan, M. Asif, and L. J. Rozario, "Improved RSA cryptosystem based on the study of number theory and public key cryptosystems," *Am. J. Eng. Res.*, vol. 4, no. 1, pp. 143–149, 2015.

[18] R. F. S. Lizy, "Improvement of RSA Algorithm Using Euclidean Technique," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 3, pp. 4694–4700, 2021.

[19] H. Shi, T. Pu, W. Mou, and Y. Chen, "NIST Randomness Tests on the Extended Key of Quantum Noise Random Stream Cipher," in *2019 18th International Conference on Optical Communications and Networks (ICOCN)*, 2019, pp. 1–3.

[20] A. N. Mazher, J. Waleed, and A. T. MaoLood, "The Security Threats and Solutions of Network Functions Virtualization: A Review," J. Al-Qadisiyah Comput. Sci. Math., vol. 12, no. 4, p. Page-38, 2020.

[21] A. M. Abbas, A. M. S. Rahma, and N. F. Hassan, "Comparative study on encrypted database techniques," J. Al-Qadisiyah Comput. Sci. Math., vol. 12, no. 3, p. Page-28, 2020.

[22] N. S. Ahmed and S. H. Ahmed, "Enhancement RC4 Algorithm Based on Logistic Maps with Multi-Parameters," J. Al-Qadisiyah Comput. Sci. Math., vol. 11, no. 4, p. Page-58, 2019.

[23] A. S. Ahmed, H. A. Salah, and J. Q. Jameel, "Multikey image encryption algorithm based on a high-complexity hyperchaotic system," J. Al-Qadisiyah Comput. Sci. Math., vol. 11, no. 3, p. Page-69, 2019.

[24] S. H. Shaker, E. Ali, and I. A. Abdullah, "Security Systems Based On Eye Movement Tracking Methods," J. AL-Qadisiyah Comput. Sci. Math., vol. 10, no. 3, p. Page-70, 2018