Mesopotamian journal of Cybersecurity
Vol.5,No.2, **pp**. 424–435
DOI: https://doi.org/10.58496/MJCS/2025/026; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

A Dynamic DNA Cryptosystem for Secure File Sharing

Kanaan J. Brakas[1,*], Mafaz Alanezi[2]

[1] *Department of Computer Science, University of Mosul, Mosul, Iraq.*

[2] *ICT Research Unit, Computer Center, University of Mosul, Mosul, Iraq.*

ARTICLE INFO

Article History

Received 18 Aug 2024
Revised 18 Oct 2024
Accepted 03 Mar 2025
Published 06 Jun 2025

Keywords

Symmetric Encryption

RNA Encoding

Elliptic Curve Cryptography

Digital Signatures

Brute Force Attack

ABSTRACT

The digital age relies heavily on file sharing, which has become more important with the increasing use of digital data and the Internet. Along with this increasing importance come major and widespread security issues, especially for files containing sensitive or vital data, such as those related to commercial, military, or healthcare sectors. The most common and effective way to protect the security and privacy of shared files containing sensitive information is still encryption. With the development of high-efficiency devices and massive processing capabilities, traditional encryption methods are increasingly under pressure from modern computer security threats, particularly quantum computing.

This study aims to raise resistance against these security concerns by proposing a dynamic encryption system based on DNA (Deoxyribonucleic Acid) encryption. DNA bases are selected to encrypt data dynamically and based on the key in the proposed system. Then, another level is added using (Ribonucleic Acid) RNA, which includes 256 * 256 different bases that are also selected dynamically. This makes this algorithm more effective in dealing with modern security attacks, including those from quantum computers. The results demonstrate the system's ability to encrypt and decrypt various types of files while providing a strong defense against conventional attacks.

Furthermore, because the system is dynamic, it provides enhanced protection against attacks by contemporary quantum computers due to the wide range of encryption rules. This makes it almost impossible to guess or test every potential vulnerability. The suggested cryptosystem also produces reduced file sizes and is more time-efficient. The performance findings show that processing times for various file types closely converge, with an average throughput between 38,000 and 47,000 bytes per second.

1. INTRODUCTION

File sharing is crucial for collaboration in various fields, such as science, industry, medicine, the military, and other areas [1]. Due to the low cost and high performance of networking and Internet technologies, digital file sharing is becoming more popular, efficient, and cost-effective. However, this rise in digital file sharing brings challenges related to security, particularly in handling private information and maintaining privacy to prevent unauthorized access. Encryption algorithms ensure that digital files remain private and secure [2]. These algorithms vary in terms of key types, data size, the number of keys employed, and performance speed [3].

One of the main challenges of digital file-sharing is preventing unauthorized access and protecting the privacy of sensitive data. Potential threats, including those posed by quantum computing, are evolving, and they could pose a real threat to traditional encryption algorithms. In addition, the complexity and sheer volume of data being shared require the implementation of effective and adaptable encryption algorithms to address these threats without compromising security [4] [5].

The proposed algorithm contributes to providing a multi-layer dynamic encryption system based on DNA encryption, as the use of DNA encryption contributes to increasing the level of dynamism by providing a choice of one out of eight different rules for the first level and then one out of 256 * 256 rules for the second level to transform and encode the data. The selection process depends on a secret key.

The goal of this proposal is to create a highly secure, multilayer dynamic DNA encryption system for file sharing that is resistant to both conventional and quantum attacks. The system uses DNA- and RNA-based cryptography, with a particular

*Corresponding author. Email: kjbbaby@gmail.com

focus on RNA coding rules, along with the elliptic curve Diffie–Hellman (ECDH) algorithm and the elliptic curve digital signature algorithm (ECDSA) for key management and exchange. This ensures the secure sharing of a wide range of files between communicating parties.

DNA cryptography is a method for encoding data in the form of DNA sequences. With this technology, each character of the alphabet can be transformed into a unique set of four basic units that form the genetic code. DNA barcoding, an emerging technology, is based on the principles of DNA information technology [6]. Each method depends on the specific requirements of the task. Recently, DNA barcoding has been widely adopted to increase the security and confidentiality of transmitted data because of its many advantages, including parallel molecular computing, data storage, transmission, and computational capabilities [7] [8].

The genetic code is a system of rules used by living organisms to convert the information stored in genetic material (genes) into proteins. There are 64 possible codons in total, each consisting of a combination of the four bases uracil (U), cytosine (C), adenine (A), and guanine (G) [9]. In the standard genetic code, 61 of these codons are sense codons, which are translated into specific amino acids by the ribosome's biochemical machinery. The remaining three codons are nonsense codons, which function as termination signals or stop codons [10]. The genetic code is considered degenerate, meaning that some codons are associated with the same amino acid, referred to as "multiplets." The sextets are encoded by six codons, the quartets by four codons, the triplets by three codons, the doublets by two codons, and the singlets by just one codon [11].

The proposed model has several limitations. Scalability concerns the suggested methodology, which demonstrates efficiency with modest data quantities; nevertheless, its effectiveness diminishes as data sizes increase. This may impact its practical applicability in the context of ever-expanding data. The model's computational complexity may necessitate substantial resources, posing a constraint in contexts with restricted hardware capabilities or for real-time applications. The model may exhibit outstanding performance in its intended domain, but its applicability to other domains may be constrained. Extensive cross-domain testing may be required to confirm its wider applicability.

The current proposal aims to build a high-security dynamic file cryptosystem that is resistant to quantum attacks, as well as other traditional attacks, to secure the sharing of various types of files between two communicating parties via DNA and (RNA) amino acid technologies, along with (ECDH) and (ECDSA) for key exchange and management.

The novelty of this proposed system lies in its use of two levels of dynamic encryption. The first level dynamically selects a DNA base from a set of available base pairs. The second level dynamically selects an RNA base from a list of 256 × 256 randomly generated base pairs, converting eight binary bits into an RNA code and vice versa. This approach makes brute-force attacks impossible, even with a quantum computer.

The rest of the paper is organized as follows: Section 2 reviews previous studies and the literature. Section 3 introduces our proposed dynamic DNA cryptography algorithm. Section 4 presents the system's performance, results, and discussion. Finally, Section 5 concludes the study.

2. LITERATURE REVIEW

Digital image encryption for the healthcare industry (Demirkol et al., 2024) proposes another chaotic equation-based DNA encryption method in which memristors are applied to the image and the circuit level is secured. The privacy and security aspects of releasing personal information from remote sensing photos should be considered[12]. (Wen et al. 2024) proposed a method to protect the identity of such faces in remote sensing images, using chaos and DNA cryptography for both secure and efficient results. Promising results have been demonstrated for unmanned aerial vehicle (UAV) deployments [13].

The purpose of color-image encryption with DNA cryptography and hyperchaos is to provide a robust and reliable means for encrypting information in terms of colors whose numerical values change randomly with time due to chaotic behavior. New DNA and hyperchaotic maps for the encryption of digital data (Almakdi et al., 2024) with this prior information allow the system to screen even more precisely, identifying exact matches where one of these precalculated functional variations is within a dangerous gene and thereby allowing for efficient screening to occur during DNA synthesis [14]. The results of Gretton et al. (2024) focus on the use of a random adversarial threshold search for automatic DNA detection to identify harmful genes accurately and securely [15].

By using DNA cryptography and the insertion method, Vidhya and Kumari (2023) proposed a novel security scheme for medical IoT systems. Their work focused on addressing key security concerns in healthcare IoT environments where patient privacy and data protection are paramount. The proposed scheme integrates two primary techniques: (i) key generation via prime numbers via the RSA algorithm and (ii) DNA-based encryption, which leverages DNA sequences for secure data transmission [16].

Selvakumar & Lokesh (2024) propose a cryptographic method to securely communicate healthcare data over the cloud via DNA cryptography and Huffman coding**.** The approach addresses the critical need for maintaining data confidentiality in healthcare, where patient privacy and data security are paramount. With cloud computing becoming increasingly integrated into healthcare, there are heightened concerns about data breaches and unauthorized access due to the centralization of sensitive information [17].

Creating an advanced system to ensure the security of all the data, reviewing those elements utilized within it, and backtesting and improving every single security aspect. Hybrid cryptosystem inspired by DNA cryptography [18]. Its security is strengthened by using a random key and a Mealy machine with state changes. For a mechanism for transmitting data from the transmitter to the recipient without compromising security, a new hybrid system that works on a cycle of encryption and decryption in DNA-based cryptography is discussed [19].

Integration of DNA coding and hyperchaotic systems in encrypting text data hidden inside a color image with an illustration that displays robustness against various forms of attacks ranging from brute statistical attacks to force attacks (Al-Khateeb & Jader, 2020) presents a new way to enhance the security level of textual information as well as hiding methods through the utilization of both DNA sequence operations and hyperchaotic systems to secure encryption [20]. Explain the need for data security in communication and tell them how to secure it via DNA cryptography as a helpful method of encoding any type of arbitrary information. Ao (Ao, 2019) researched data security communication, which focuses on DNA cryptography as a possible solution for exclusive encryption methods, such as transferring sensitive data securely to gears [6].

Highlights the efficiency and security of the proposed cryptosystem compared with existing systems, shielding big data from online threats. (Vaishali & Manohar Naik 2024) proposed a DNA cryptosystem through bioinformatics and Diffie-Hellman[21]. It is the process of key exchange that makes our data communication secure. We evaluate the proposed novel authentication mechanism for both single-node and multinode Hadoop/SPARK clusters to analyse its effectiveness. To create stronger, less penetrable passwords, Balaraju et al. (2024) presented a DNA approach to produce safe and flexible passwords in Hadoop clusters. The algorithm ensures reliable authentication and offers defense against dictionary and brute-force attacks [22].

3. MATERIALS AND METHODS

The proposed dynamic cryptosystem consists of three essential components: key management, encryption, and decryption, as shown in Figure 1.

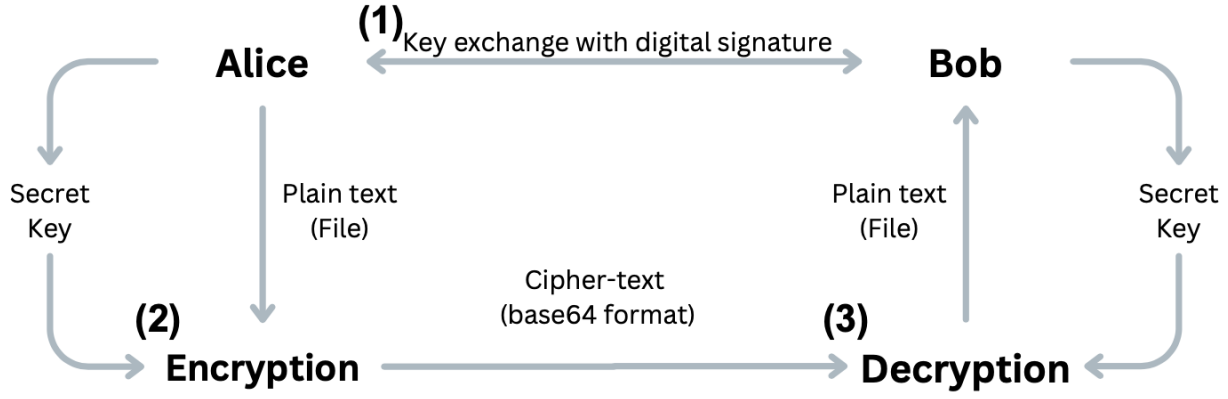

Fig. 1. Dynamic Cryptosystem Architecture

3.1 Key management

ECDH is a protocol that relies on the establishment and exchange of a secret key. This is achieved by generating a pair of keys for each entity involved in the communication. Each party selects a private key, chosen randomly, with a size of 256 bits (in our study). A public key is subsequently generated, representing a point with coordinates xxx and yyy, which is calculated as a point on an elliptic curve, typically by repeatedly adding the base point to itself.

The process of key exchange between Alice and Bob is illustrated in Figure 2. After Alice and Bob generate their key pairs, each party exchanges their public key with the other. Additionally, certain parameters, such as the modulus and the reference point, must be agreed upon.

The subsequent phase begins with each party performing a series of operations. First, Alice uses her private key, the base point, and the modulus to determine the A value, which serves as the initial and global value to be shared. Similarly, Bob uses his private key to determine the B value. Second, Alice signs a value via her private key and transmits it to Bob. In turn, Bob signs his calculated value and shares it with Alice. Third, Alice verifies the signature of the signed value received from Bob and subsequently calculates the shared secret key via her private key, the modulus, and the initial and global values that Bob previously calculated. In the same way, Bob derives the shared secret key, which corresponds to a point on the elliptic curve with specific X and Y coordinates.

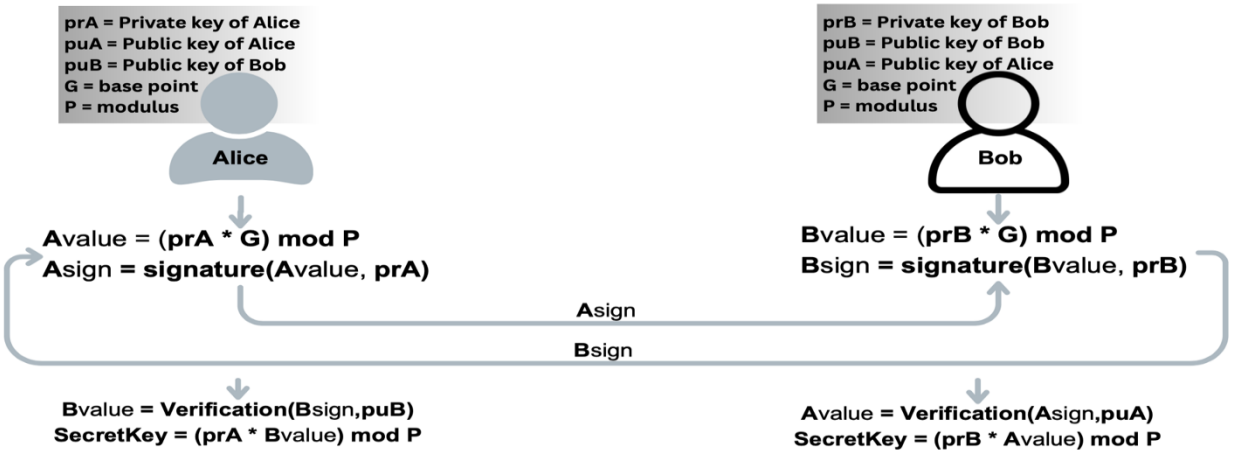

Fig. 2. Key exchange with a digital signature.

3.2 Encryption

The cryptosystem begins with two inputs: the file (plain text) and the secret point key (x, y) from the previous key management. Located at the sender's end (Alice), the proposed encryption algorithm is used to encrypt the file, which is then sent to the receiver (Bob). This phase involves eight steps of data conversion, as shown in Figure 3.

Step 1: Read the file (plain text) byte-byte and convert it to a binary data format.

Step 2: Encode the binary data format into DNA encoding via one of the eight DNA encoding rules shown in Table 1. The rule is selected on the basis of the formula (x coordinate of the secret point key mod 8)+1.

Step 3: Prepare the DNA encryption key from the secret point key via the following equation:

$$DNA\ key\ =\ str(secret\ point\ key\ (x,y))\ +\ str(x\ coordinate\ *\ y\ coordinate) \quad (1)$$

The key is then converted to DNA encoding by selecting the same DNA rule number, and the key is repeated until it matches the length of the DNA-encoded plain text.

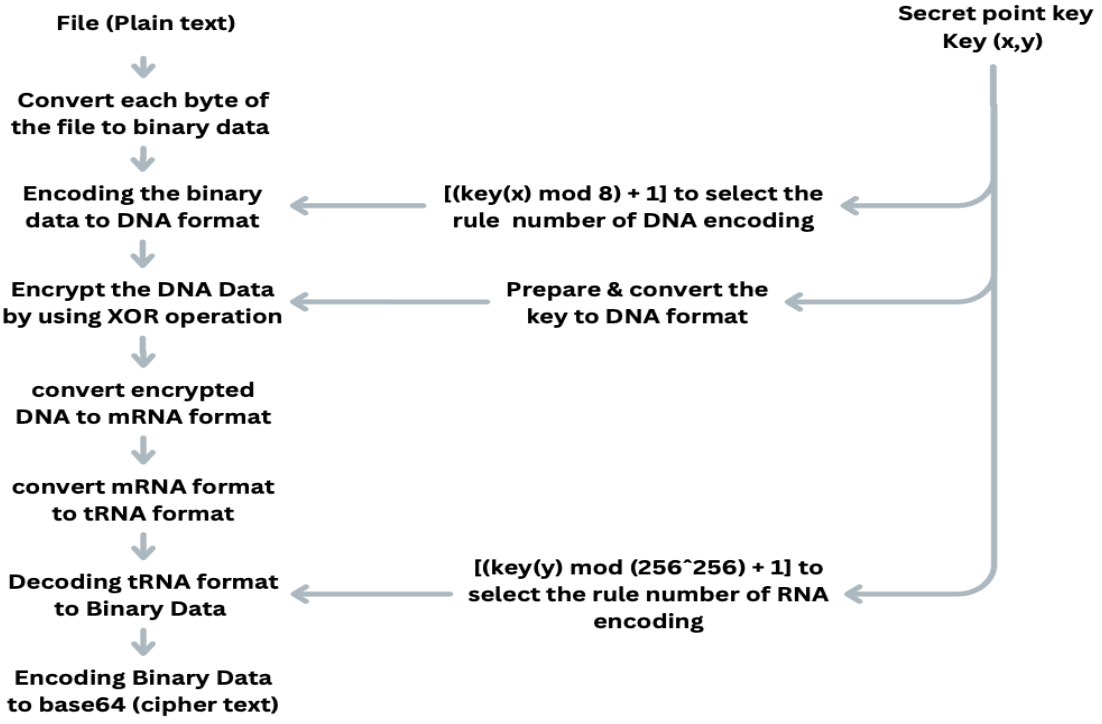

Fig. 3. Encryption System

Step 4: Encrypt the DNA-encoded plain text with the DNA key via the DNA-XOR operation to obtain the encrypted DNA data.

TABLE I. DNA ENCODING RULES [23].

Base	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
A	00	00	11	11	10	10	01	01
C	01	10	01	10	11	00	11	00
G	10	01	10	01	00	11	00	11
T	11	11	00	00	01	01	10	10

DNA encoding rules are used to convert DNA sequences into binary codes. Each nucleotide base (A, C, G, T) is represented by a unique binary code according to different rules. These rules can be used in various applications, including bioinformatics, cryptography, and data storage. Each base is consistently assigned a unique binary code in each rule. For example, 'A' is represented as 00 in Rules 1 and 2 but as 11 in Rules 3 and 4.

Step 5: Convert the encrypted DNA data to Messenger RNA (mRNA) encoding via Table 2.

TABLE II. DNA ENCODING-TO-MRNA CONVERSION TABLE [23]

DNA	mRNA
A	U
C	G
G	C
T	A

DNA is used as a template to create messenger RNA (mRNA). Each DNA base pairs with a complementary mRNA base. Adenine (A) in DNA pairs with uracil (U) in mRNA. Cytosine (C) in DNA pairs with guanine (G) in mRNA.

Step 6: Convert mRNA encoding to transfer RNA (tRNA) encoding by using Table 3.

TABLE III. MRNA ENCODING TO TRNA CONVERSION TABLE [23]

mRNA	tRNA
A	U
C	G
G	C
U	A

mRNA is used as a template to synthesize proteins. Each mRNA base pairs with a complementary tRNA base, which carries the corresponding amino acid. Adenine (A) in mRNA pairs with uracil (U) in tRNA, cytosine (C) in mRNA pairs with guanine (G) in tRNA, guanine (G) in mRNA pairs with cytosine (C) in tRNA, and uracil (U) in mRNA pairs with adenine (A) in tRNA.

Step 7: Convert the tRNA encoding data to binary data by using the tRNA encoding table, which has 256 entities selected randomly, as shown in Table 4. In this step, each of the four encoding tokens is converted to eight bits of binary data. The rule number is selected with the help of the y coordinate from (the secret point key mod 256) + 1.

TABLE IV. TRNA ENCODING TABLE

Base	Rule1	Rule2	Rule3	…	…	Rule254	Rule255	Rule256
AAAA	00000000	10110000	11100001	…	…	00111110	01010100	11100110
AAAC	00110001	10000110	00000000	…	…	00101110	11100001	01100010
AAAG	01001010	11010011	10000110	…	…	10000001	01110110	01111011
AAAU	01100001	11101100	10000001	…	…	11010011	10000110	10101110
…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…
UUUA	00001111	01010101	10001101	…	…	01011111	01100001	01110011
UUUC	01010100	10001101	10011011	…	…	01110010	11001010	11101100
UUUG	11100001	00111101	10110000	…	…	01110000	01111110	01101111
UUUU	10110000	01110001	01100110	…	…	10110100	10111001	10101111

The table shows the binary encoding of DNA sequences according to multiple bases, ranging from base 1 to base 256. An 8-bit binary code under each base represents each DNA sequence (e.g., AAAA, AAAC). Each encryption process selects one rule on the basis of the secret key. As shown in Figure 4, the RNA representation we use is four tokens for each of the eight bits of data, so by selecting 256 random values for each rule, we obtain 256 (or more if required).

Step 8: Encode the binary data to the base64 string, which represents the cipher text to be sent to the receiver entity (Bob).

3.3 Decryption

The decryption system starts with two inputs, the base64 string (plain text) and the secret point key (x, y) located at the receiver entity (Bob), uses the proposed decryption algorithm to decrypt the base64 string and then returns the file (plain text). This phase contains eight steps of data conversion, as shown in Fig. 5.

Step 1: Start with the Base64 data (ciphertext), which Bob received from Alice and decoded into binary data.

Step 2: Convert binary data to tRNA with the help of the y coordinate from (the secret point key mod 256) + 1 to select the rule number to be used from Table 4.

Step 3: Convert tRNA to mRNA via the information in Table 3.

Step 4: Convert the mRNA to DNA format by using Table 2.

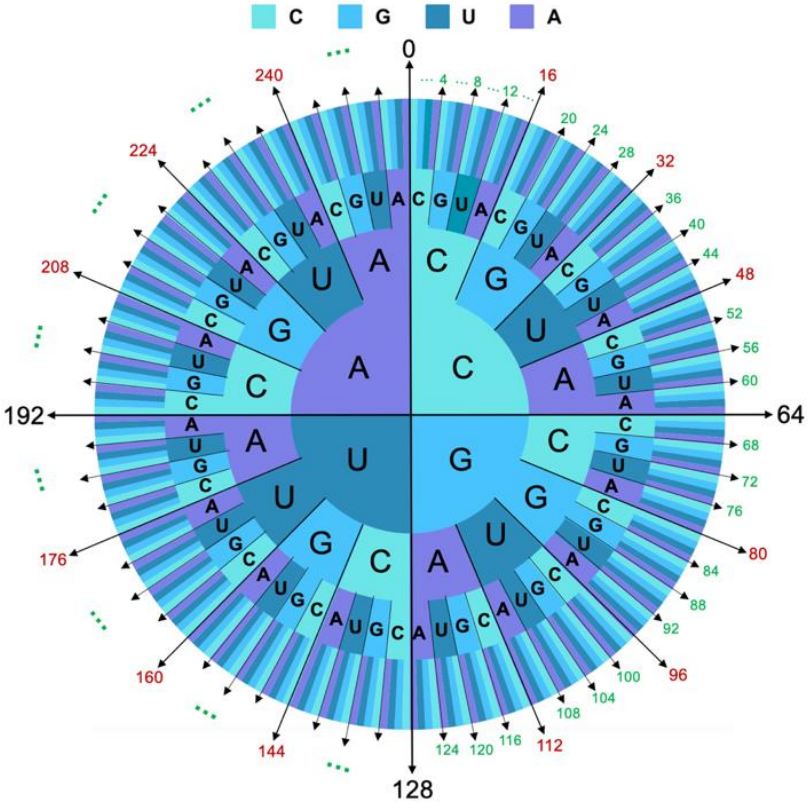

Fig. 4. Four-bit RNA representation

Step 5: Prepare the DNA decryption key from the secret point key via equation (1). The key is then converted to the DNA encoding by selecting the same DNA rule number and repeating the key to obtain the same length of the DNA encoding of the DNA data.

Step 6: Decrypt the DNA data with the DNA key via the DNA-XOR operation to obtain the decrypted DNA data.

Step 7: Decode the DNA data to binary data format depending on the eight DNA encoding rules shown in Table 1, which are selected with the help of the x coordinate from (the secret point key mod 8) + 1.

Step 8: Convert binary data to bytes and write each byte to a file to obtain the file (plain text) back.

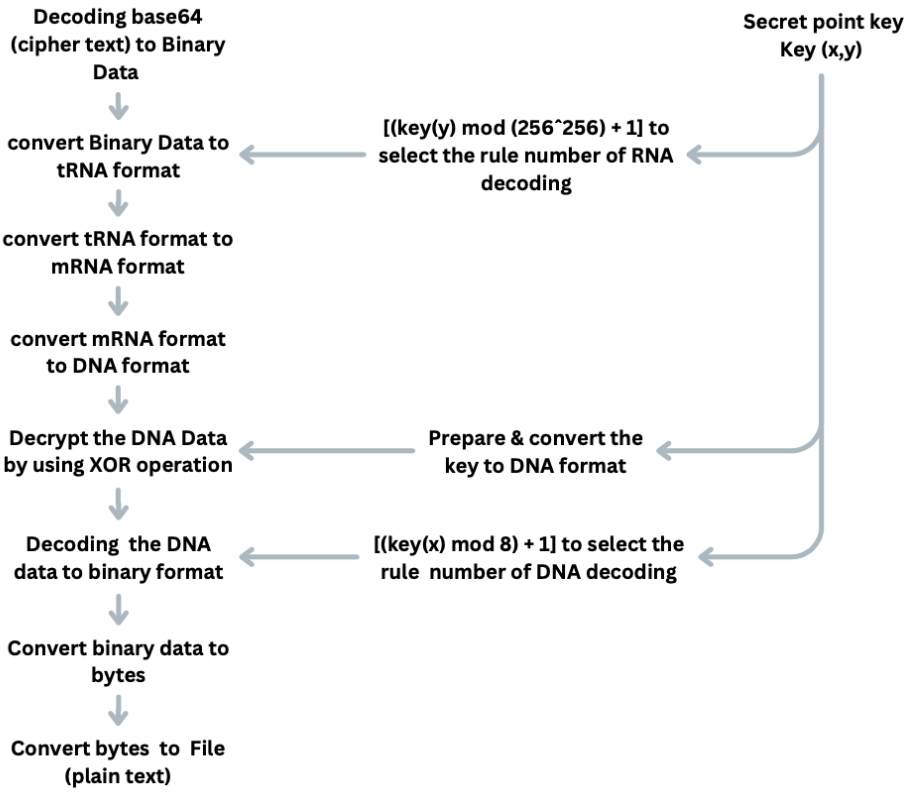

Fig. 5. Decryption System

4. RESULTS AND DISCUSSIONS

The proposed algorithm converts the file (plaintext) into data in the form of bytes and then enters these data into the abovementioned encryption procedures to obtain the ciphertext. This file can be of various formats, such as Word, PDF, image, audio, video, etc. To implement this algorithm to ensure the secure sharing of the files, the following example shows the data produced from each step of the proposed algorithm procedures by using Kanaan.jpeg, which is an image file in plain text, and the secret key is as follows:

Key (x, y) = (27001931682204286861469007024618058498465619925259721786611752372916664828761,
58126487129813588038652165772513998914773761740492159043050819328758108903842)

On Alice's side, the encryption process is applied as follows:

Step 1: Read the file byte by by byte.

(b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x00\x00H\x00H\x00\x00\xff\xe1\x00XExif\x00\x00MM\x00*\x00\x00\x00\x08\x00\x02\x01\x12\x00\x03\x00\x00\x00\x01\x00\x01……')

The data are then converted to binary format.

(1111111111011000111111111110000000000000000100000100101001000110010010010100011000000000000….)

Step 2: Convert the binary data to the DNA encoding format. By using the DNA encoding rules in Table 1, which select rule 2, with the help of [(x coordinate of the secret key mod 8) + 1], we obtain the following data:

(AAAAAGCTAAAAACTTTTTTTGTTGTCCGTGCGTCGGTGCTTTTTTTGTT….)

Step 3: In this step, the DNA encryption key is prepared and converted to the DNA encoding key format via the secret key via equation (1).

(TCGATTTTTGCGTATGGCCTTCTCTTGTTCCTGCTGGTGCCGTTTT….)

Step 4: Encrypt DNA data with a DNA encryption key via the DNA-XOR operation.

(AAAAAGCTAAAAACTTTTTTTGTTGTCCGTGCGTCGGTGCTTTTTTTGTT….)

DNA-XOR

(TCGATTTTTGCGTATGGCCTTCTCTTGTTCCTGCTGGTGCCGTTTT….)

To obtain DNA encrypted data as.

(AGCTAGCTACGCAGTGGCCTTATCGTACGCACTCTGTCGAGTGCCGTGTT….)

Step 5: Convert the DNA encoding from Step 4 to the mRNA encoding via Table 2 to obtain the mRNA data encoding.

(UCGAUCGAUGCGUCACCGGAAUAGCAUGCGUGAGACAGCUCACGGCACAA….)

Step 6: Table 3 shows the conversion of mRNA data encoding to tRNA encoding.

(AGCUAGCUACGCAGUGGCCUUAUCGUACGCACUCUGUCGAGUGCCGUGUU….)

Step 7: Convert the tRNA data to binary data format. By using the tRNA encoding rules in Table 4, which select rule 163 with the help of [(y coordinate of the secret key mod 256) + 1], we obtain the following data:

(1000101010001010100100110010101110111100100111100011110101011011010100001000111001111001011….)

Step 8: encode binary data (ciphertext) to base64 format and send it into Bob in this form.

(b'MTAwMDEwMTAxMDAwMTAxMDEwMDEwMDExMDAxMDEwMTExMDExExMT….')

On the other hand, the decryption process is applied as follows:

Step 1: Decode the received ciphertext from base64 to binary format to obtain

(1000101010001010100100110010101110111100100111100011110101011011010100001000111001111001011….)

Step 2: Convert binary data to tRNA encoding format by using the tRNA encoding rules in Table 4, which selects rule 163. With the help of the [(y coordinate of the secret key mod 256) + 1], we obtain the following data:

(AGCUAGCUACGCAGUGGCCUUAUCGUACGCACUCUGUCGAGUGCCGUGUU….)

Step 3: Table 3 shows the conversion of tRNA data encoding to mRNA encoding.

(UCGAUCGAUGCGUCACCGGAAUAGCAUGCGUGAGACAGCUCACGGCACAA….)

Step 4: Convert the mRNA-encoding data to DNA-encoding data with the help of Table 2 to obtain the encrypted DNA data format.

(AGCTAGCTACGCAGTGGCCTTATCGTACGCACTCTGTCGAGTGCCGTGTT….)

Step 5: In this step, the DNA encryption key is prepared and converted to the DNA encoding key format via the secret key via equation (1).

(TCGATTTTTGCGTATGGCCTTCTCTTGTTCCTGCTGGTGCCGTTTT….)

Step 6: Decrypt the encrypted DNA data format with the DNA encryption key via the DNA-XOR operation.

(AGCTAGCTACGCAGTGGCCTTATCGTACGCACTCTGTCGAGTGCCGTGTT….)

DNA-XOR

(TCGATTTTTGCGTATGGCCTTCTCTTGTTCCTGCTGGTGCCGTTTT….)

To obtain the DNA decrypted data as

(AAAAAGCTAAAAACTTTTTTTGTTGTCCGTGCGTCGGTGCTTTTTTTGTT….)

Step 7: Convert the DNA decrypted data to binary format by using the DNA encoding rules in Table 1, which select rule 2 with the help of [(x coordinate of the secret key mod 8) + 1]. We obtain the following data:

(1111111111011000111111111110000000000000000010000010010100100011001001001010001100000000000….)

Step 8: Convert the binary data format to byte as follows:

(b'MTAwMDEwMTAxMDAwMTAxMDEwMDEwMDExMDAxMDEwMTExMDExExMT….')

and write it to a file to obtain the plaintext.

TABLE V. ENCRYPTION AND DECRYPTION TIMES

File type	File size	File size in bytes	Encryption time	Decryption time	No. bytes per second for Encryption
.jpeg	44KB	44478	1.1 seconds	1.2 seconds	40434
.mp3	770KB	768978	16.2 seconds	19.66 seconds	47467
.png	823KB	819669	19.4 seconds	22.5 seconds	42250
.mp4	1.3MB	1252030	32.6 seconds	33.5 seconds	38405
.docx	3.1MB	3061298	68.3 seconds	77.7 seconds	44821
.pdf	6.4MB	5964353	129.5 seconds	157.5 seconds	46056

This table lists various file types along with their sizes, encryption, and decryption times, and the number of bytes processed per second during encryption. Both the encryption and decryption times increase with increasing file size. Larger files generally take longer to encrypt and decrypt, which is expected due to the increased amount of data. The variation in bytes per second for encryption suggests that different file types may have different levels of complexity or overhead associated with their encryption processes. Understanding these performance metrics is crucial for optimizing the encryption and decryption processes, especially when dealing with large volumes of data. This helps in planning and managing resources effectively, ensuring that data security measures do not significantly impact system performance.

The proposed system provides important functions, including message secrecy through encryption and authentication and digital signatures through the application of ECDH and ECDSA at the key exchange stage. To obtain the cipher-text, as explained above, the plain text, which uses the symmetric encryption (secret key) and relies on an encryption key, consists of a point of X and Y coordinates of 77 digits per coordinate and goes through seven substages of transformations, where the conversion of data to DNA encoding uses eight different rules for this purpose, then encrypts these data using the key and the DNA-XOR operator and converts the resulting data to RNA and then encodes it to binary using 256 possible cases or rules, where we expand and convert 256 cases or possible rules to 256256 rules, thus taking all possible cases for each rule, randomly selecting 256 rules, encrypting those data to the bace64 encoding to facilitate transmission via the internet and network without loss of data.

As mentioned earlier, the data sent, information, or messages are kept out of reach of the attackers, increasing the system resistance to many attacks. First, the large size of the key used and the large number of rules available for the transformation of the RNA into binary make the resistance of the system very high to attacks such as brute force attacks, and it is impossible to guess the secret key at the same time. Knowing the cipher text and plain text will never lead to finding the secret key by using meeting-in-the-middle attacks because of the dynamic approach of selecting the rules for converting RNA to binary. Second, they are very resistant to factoring attacks when symmetric encryption is used. Third, because of the seven substages of the encryption process, the cipher text is professional, radical, and completely different from the plain text and thus has a high degree of security and resistance to differential cryptography.

We used six files of different sizes and types, as shown in Table 5, to evaluate the proposed system's performance on the basis of the encryption and decryption times, as shown in Figure 6. The results showed an approximate convergence in the calculated times with a difference in the file type, but the average number of bytes encrypted per second ranged between 38,000 and 47,000.

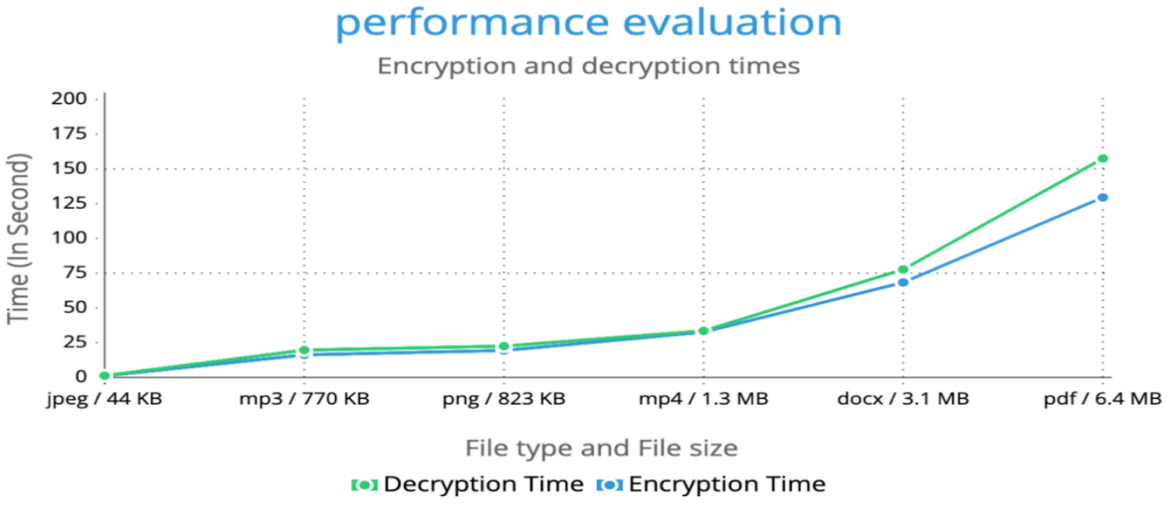

Fig. 6. Encryption and Decryption Times

The study presents the results of an encryption method NIST evaluated by statistical tests to ascertain the regularity of p values and the ratio of successful sequences.

- **Principal Findings of the Results:**
- **Frequency Test:** Eight out of 10 sequences were successful, yielding a p value of 0.035174.
- **Block** frequency test: Nine of the 10 sequences were successful (p value = 0.213309).
- **Cumulative sums test:** Successfully passed eight and nine out of 10 sequences, with p values varying from 0.122325 to 0.350485.
- **Runs Test**: Achieving success in eight out of ten sequences with a p value of 0.739918.
- Longest Run Test: Successfully passed 8 out of 10 sequences, yielding a p value of 0.534146.
- **FFT Test:** Successfully passed seven out of ten sequences with a p value of 0.534146.
- Nonoverlapping template test: The results varied from 6 to 10 successful sequences, with p values ranging from 0.000003 to 0.911413.
- **Serial Test:** Demonstrated inadequate performance, with merely 1 out of 10 and 3 out of 10 sequences succeeding (p value = 0.000000).
- **General observations:** Most of the tests yield satisfactory passing rates (7/10--10/10 sequences) and acceptable p values.

Poor results in the sequential test indicate weaknesses in the randomness of the algorithm's results. This affects the strength and security of the encryption, making it vulnerable to attacks that exploit patterns or predictability. Dynamic encryption mitigates these risks by ensuring that each encryption operation differs completely from its predecessor and follower, thereby complicating the process of predicting and exploiting patterns to reveal parts of the key or plain text.

For the entire NIST test report, visit "*https://github.com/DrKanaanJalal/NIST-tests/blob/main/the%20result*".

The system incorporates multiple encryption levels based on DNA cryptography and enhances security by integrating the concept of dynamic encryption. This dynamic approach ensures that encryption keys and rules are continuously updated, making the system more secure and resistant to quantum attacks. By dynamically selecting DNA and RNA bases for encryption and periodically changing the encryption parameters, the system effectively counters modern security threats, including those posed by quantum computing. This multilayered, adaptive encryption strategy provides robust protection for sensitive data, ensuring that potential vulnerabilities are minimized and making it exceedingly difficult for attackers to compromise the system.

To ensure the robustness of the proposed dynamic encryption system, which is based on DNA and RNA, it is crucial to discuss potential attacks and security threats and demonstrate how the method is immune to them. Some common threats and how the proposed system addresses them are as follows:

- **Brute Force Attacks:**

Threat: Attackers try all possible keys until the correct one is found.

Defense: The dynamic nature of the encryption, with a vast number of possible RNA base pairs (256 * 256), makes brute force attacks impractical because of the enormous number of combinations.

- **Quantum Computing Attacks:**

Threat: Quantum computers can solve complex problems much faster than classical computers can, potentially breaking traditional encryption.

Defense: The proposed system's dynamic encryption rules and the use of both DNA and RNA bases create a highly complex and variable encryption scheme, making it resistant to quantum attacks.

- **Side-channel Attacks:**

Threat: Attackers gain information from the physical implementation of the encryption system (e.g., timing information, power consumption).

Defense: Implementing constant-time algorithms and masking techniques can mitigate side-channel attacks. The dynamic nature of the encryption also adds an extra layer of complexity.

- **Cryptanalysis Attacks:**

Threat: Attackers use mathematical techniques to break the encryption.

Defense: The use of DNA and RNA bases, along with dynamic encryption rules, increases the complexity and variability of the encryption, making cryptanalysis significantly more challenging.

5. LIMITATION OF THE STUDY

While the proposed dynamic encryption system based on DNA and RNA encryption shows promising results, there are several limitations to consider:

1. **Complexity and Implementation:** A system's complexity might pose challenges in practical implementation, especially in environments with limited computational resources. The development and maintenance of such a dynamic system could require significant expertise and resources.
2. **Performance Overhead:** Although the system is designed to be time efficient, the additional layers of encryption (DNA and RNA) may introduce performance overhead, particularly for very large files or real-time applications.
3. **Scalability:** The scalability of the system needs to be thoroughly tested. As the volume of data increases, the system's performance and efficiency might be impacted. Ensuring consistent performance across different types of files and sizes is crucial for widespread adoption.
4. **Quantum Computing Threats:** While the system aims to resist quantum computing attacks, the evolving nature of quantum algorithms means that continuous updates and improvements will be necessary to maintain security. The effectiveness of the system against future quantum computing advancements remains to be seen.
5. **Interoperability:** Integrating this encryption system with existing infrastructure and protocols could be challenging. To facilitate seamless adoption, compatibility with various file formats and systems needs to be ensured.

6. CONCLUSION

This study discusses a dynamic multilayer encryption algorithm based on DNA in the first layer and ribonucleic acid (RNA) in the second layer. This research addresses the increasing threats posed by the development of quantum computing and the need for advanced encryption techniques to address modern security issues. The proposed approach, which randomly selects from a set of 256 * 256 RNA base pairs, provides a highly secure dynamic encryption mechanism, making it resistant and effective against quantum attacks.

The study reveals two significant key findings. First, the encryption technique demonstrates exceptional efficiency across various file types, ensuring thorough data transformation and robust protection against both conventional and quantum attacks. The algorithm's impressive encryption speed, ranging from 38,000--47,000 bytes per second, highlights its practicality for real-world applications. Second, the implementation of dynamic encryption significantly enhances security by reducing the likelihood of decryption, particularly in anticipation of advancements in quantum computing. These findings underscore the algorithm's potential to provide a highly secure and efficient encryption solution in an evolving digital landscape.

The proposed encryption algorithm offers several notable advantages. First, its dynamic nature significantly enhances security by providing robust protection against decryption attempts, thereby ensuring data confidentiality and integrity. Second, the algorithm's efficiency and ability to handle various file types and sizes make it highly scalable and suitable for a wide range of applications. Finally, by addressing potential threats from quantum computing, this study presents a forward-looking solution that provides future-proof data security against emerging computational challenges. These advantages collectively underscore the algorithm's potential to offer a secure, adaptable, and resilient encryption solution in an evolving digital landscape.

The implications of this study are far-reaching and significant. First, the practical applications of the encryption method are evident in its potential deployment across critical sectors such as healthcare, military, and commercial industries, where protecting sensitive information is of paramount importance. Second, the study underscores the need for continued research and development in the field of quantum-resistant algorithms, highlighting the importance of remaining ahead of emerging computational threats. The findings thus have the potential to shape new security standards and policies, ensure robust protection against advanced computational threats, and enhance overall data security in an increasingly digital world. Together, these implications underscore the contribution of this study to the development of encryption technology and its critical role in ensuring data security in the future.

Conflicts of interest

The authors of this work report no conflicts of interest.

Funding

No funding.

Acknowledgement

We are grateful for all the support the Computer Science Department/University of Mosul/Iraq provides.

References

[1] C. J. Lee, M. M. Kimball, E. C. Deussing, and T. D. Kirsch, "Use of Information Technology Systems for Regional Health Care Information-Sharing and Coordination During Large-Scale Medical Surge Events," *Disaster Med Public Health Prep*, vol. 18, p. e1, Dec. 2024, doi: 10.1017/dmp.2023.218.

[2] C. Manthiramoorthy, K. M. S. Khan, and N. A. A, "Comparing Several Encrypted Cloud Storage Platforms," *International Journal of Mathematics, Statistics, and Computer Science*, vol. 2, pp. 44–62, Aug. 2023, doi: 10.59543/ijmscs.v2i.7971.

[3] Q. Lai and G. Hu, "A Nonuniform Pixel Split Encryption Scheme Integrated With Compressive Sensing and Its Application in IoMT," *IEEE Trans Industr Inform*, pp. 1–11, 2024, doi: 10.1109/TII.2024.3403266.

[4] Oluwatoyin Ajoke Fayayola, Oluwabukunmi Latifat Olorunfemi, and Philip Olaseni Shoetan, "DATA PRIVACY AND SECURITY IN IT: A REVIEW OF TECHNIQUES AND CHALLENGES," *Computer Science & IT Research Journal*, vol. 5, no. 3, pp. 606–615, Mar. 2024, doi: 10.51594/csitrj.v5i3.909.

[5] G. E. Al-Kateb, I. Khaleel, and M. Aljanabi, "CryptoGenSec: A Hybrid Generative AI Algorithm for Dynamic Cryptographic Cyber Defence," *Mesopotamian Journal of CyberSecurity*, vol. 4, no. 3, pp. 150–163, Sep. 2024, doi: 10.58496/MJCS/2024/013.

[6] S.-I. Ao, *World Congress on Engineering : WCE 2019 : 3-5 July, 2019, Imperial College London, London, U.K.* 2019.

[7] C.-T. Berezin, S. Peccoud, D. M. Kar, and J. Peccoud, "Cryptographic approaches to authenticating synthetic DNA sequences," *Trends Biotechnol*, Feb. 2024, doi: 10.1016/j.tibtech.2024.02.002.

[8] M. Subhi, O. F. Rashid, S. A. Abdulsahib, M. K. Hussein, and S. M. Mohammed, "A Novel Anomaly Intrusion Detection Method based on RNA Encoding and ResNet50 Model," *Mesopotamian Journal of CyberSecurity*, vol. 4, no. 2, pp. 120–128, Aug. 2024, doi: 10.58496/MJCS/2024/011.

[9] M. Rapacioli, R. Katz, and V. Flores, "Rules governing the genetic code degeneracy/redundancy and spatial organization of the codon informative properties," *Front Appl Math Stat*, vol. 10, May 2024, doi: 10.3389/fams.2024.1340640.

[10] S. T. Parvathy, V. Udayasuriyan, and V. Bhadana, "Codon usage bias," *Mol Biol Rep*, vol. 49, no. 1, pp. 539–565, Jan. 2022, doi: 10.1007/s11033-021-06749-4.

[11] T. Négadi, "Fibonacci-like Sequences Reveal the Genetic Code Symmetries, Also When the Amino Acids Are in a Physiological Environment," *Symmetry (Basel)*, vol. 16, no. 3, p. 293, Mar. 2024, doi: 10.3390/sym16030293.

[12] A. S. Demirkol, M. E. Sahin, B. Karakaya, H. Ulutas, A. Ascoli, and R. Tetzlaff, "Real time hybrid medical image encryption algorithm combining memristor-based chaos with DNA coding," *Chaos Solitons Fractals*, vol. 183, Jun. 2024, doi: 10.1016/j.chaos.2024.114923.

[13] H. Wen, Z. Xie, Z. Wu, Y. Lin, and W. Feng, "Exploring the future application of UAVs: Face image privacy protection scheme based on chaos and DNA cryptography," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 1, Jan. 2024, doi: 10.1016/j.jksuci.2023.101871.

[14] S. Almakdi, I. Ishaque, M. Khan, M. S. Alshehri, and N. Munir, "Key dependent information confidentiality scheme based on deoxyribonucleic acid (DNA) and circular shifting," *Heliyon*, vol. 10, no. 1, Jan. 2024, doi: 10.1016/j.heliyon.2023.e23572.

[15] D. Gretton *et al.*, "Random adversarial threshold search enables automated DNA screening", doi: 10.1101/2024.03.20.585782.

[16] E. Vidhya and R. Kiruba Kumari, "Medical Data Security in IOT Using DNA Cryptography and Insertion Method," *Data Analytics and Artificial Intelligence*, vol. 3, no. 2, pp. 21–25, Jan. 2023, doi: 10.46632/daai/3/2/5.

[17] K. Selvakumar and S. Lokesh, "A cryptographic method to have a secure communication of health care digital data into the cloud," *Automatika*, vol. 65, no. 1, pp. 373–386, Jan. 2024, doi: 10.1080/00051144.2023.2301240.

[18] S. Almakdi, I. Ishaque, M. Khan, M. S. Alshehri, and N. Munir, "Key dependent information confidentiality scheme based on deoxyribonucleic acid (DNA) and circular shifting," *Heliyon*, vol. 10, no. 1, Jan. 2024, doi: 10.1016/j.heliyon.2023.e23572.

[19] O. F. Rashid, "Text Encryption Based on DNA Cryptography, RNA, and Amino Acid," 2023.

[20] Z. N. Al-Khateeb and M. Jader, "Encryption and hiding text using DNA coding and hyperchaotic system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 2, p. 766, Aug. 2020, doi: 10.11591/ijeecs.v19.i2.pp766-774.

[21] R. Vaishali and S. Manohar Naik, "A DNA Cryptosystem Using Diffie–Hellman Key Exchange," *SN Comput Sci*, vol. 5, no. 3, Mar. 2024, doi: 10.1007/s42979-024-02607-9.

[22] J. Balaraju, P. R. Rao, V. Biksham, P. V. R. D. Prasada Rao, and P. Tumuluru, "International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Dynamic Password to Enforce Secure Authentication Using DNA," 2024. [Online]. Available: www.ijisae.org

[23] W. Peng, S. Cui, and C. Song, "One-time-pad cipher algorithm based on confusion mapping and DNA storage technology," *PLoS One*, vol. 16, no. 1, p. e0245506, Jan. 2021, doi: 10.1371/journal.pone.0245506.