Hindawi
Security and Communication Networks
Volume 2023, Article ID 9967129, 12 pages
https://doi.org/10.1155/2023/9967129

Research Article

A Lightweight Cryptographic Algorithm Based on DNA Computing for IoT Devices

Gamil R. S. Qaid [1] **and Nadhem Sultan Ebrahim** [2]

[1]*College of Computer Science and Engineering, Hodeidah University, Al Hudaydah, Yemen*
[2]*College of Computing and IT, University of Bisha, Bisha, Saudi Arabia*

Correspondence should be addressed to Gamil R. S. Qaid; dr.g_qaid@hoduniv.net.ye

Received 19 December 2022; Revised 23 January 2023; Accepted 17 April 2023; Published 8 May 2023

Academic Editor: Chunjiong Zhang

Internet of Things (IoT) applications are used in almost every part of our life, so it is important to protect the sensitive data and information that is transmitted over wireless networks such as images and documents. The IoT devices have limited computational resources; they are called limited devices due to their limited processors and memory size. Traditional encryption methods require a lot of computing power; therefore, it is difficult to implement traditional cryptographic algorithm on IoT processor. Finally, a new, lightweight encryption method based on the DNA sequence is proposed to suit the IoT devices in a way to make an easy and secure the communications among the IoT devices. DNA sequences are very random, so we have used it to make a strong secret key that is hard for attackers to break. The proposed method has an advantage in terms of efficiency and strength. Experiments and security tests show that the proposed encryption system not only has a good encryption effect and can withstand known attacks, but it is also fast enough for real-world use. The DNA key is used to encrypt files using two simple and reliable methods such as substitution and transposition procedures that meet IoT computational requirements. In addition, when compared with other encryption algorithms, the experimental results shows that the key size, encryption time, and distortion preparation are all superior.

1. Introduction

The Internet of Things (IoT) is a critical component of the modern world since it enables people to live more easily and wisely. The Internet of Things is a network that communicates with the physical world. Its fundamental technologies include wireless sensor networks and the Internet. It is a worldwide network of intelligent objects, dubbed "things," loaded with sensors, electronics, and software. In a nutshell, the Internet of Things (IoT) is a collection of gadgets or sensors that create and send data over a wireless network [1]. The main goal of IoT devices is to initially generate the data from different sources then record it, the next phase is to collect the data and get process it, and finally transmit data via communication channels, as well as to control many bigger units on a regular basis. The IoT is expected to continuously grow, with a forecast of 30 billion linked devices in 2020, and that there would be 75 billion IoT devices in the global network by 2025. With such a large number of devices, the breadth of the transmission channel becomes more important. According to Atom Beam, the volume of data exchanged by IoT devices will exceed 90 zettabytes by 2025. The total amount of data sent today is 30 zettabytes [2].

The growth of the Internet of Things has various advantages, since it will affect how people perform ordinary chores and potentially transform the planet. Smart lighting will certainly reduce the consumption of energy though lowering the electricity cost. Smart buildings, healthcare monitoring, smart homes, smart cities, and other human activities are all covered by IoT applications [3]. Handling and securing the massive volume of data created by heterogeneous IoT devices is one of the challenges in IoT applications. The generated data from IoT devices and

applications have become a desirable target for anyone who wants to gain access to such information, such as attackers. The approach is to use cryptographic measures to protect the data, though only granting access to authorized people to decode it. The Data Encryption Standard (DES) and Advanced Data Encryption (AES) algorithms cannot be used to encrypt and protect the security of data generated by IoT devices [4–7].

IoT devices are termed constrained devices since their computation resources are limited with the number of processors and size of memory. As a result, traditional encryption approaches, which need more computing and resource capability, are incompatible with IoT devices. Therefore, there is a need for a new approach of encryption model with lightweight requirements, whereas the encryption model or system for IoT constrained devices takes advantage of combining the features provided by the most used encryption mechanisms to provide a robust data confidentiality efficiently, while easily adapting to emerging and converging technologies like DNA-computing algorithm. The DNA-encryption approach proposed in this study is a simple encryption system for data created by the Internet of Things, such as words and photos. The purpose of this research is to provide a novel DNA-based lightweight cryptography (LWCD) that creates keys for multiencryption rounds by using the DNA sequence as a key and performing some operations on it. Depending on the relevance of the collected data, the block size of the multiencryption rounds can be modified to accommodate IoT devices and provide high robust and solid encryption. LWCD's two main processes are substitution and transposition. As mentioned above, because IoT devices have limited resources from the prospective of processor, memory, storage devices, and limited power, especially when using a battery, they are classified as resource-constrained devices [8].

The large amounts of data generated by IoT devices and transferring it over the Internet to the applications in the destination server in the cloud or on-premises data centers will almost certainly include private data such as personal, medical, or other sensitive data. An unauthorized person can easily hack and divulge this information, or it can be altered while being stored or sent. As a result, an encryption technique is required to safeguard the data's secrecy and integrity [9].

An encryption algorithm is a mechanism for converting plaintext to cipher text to maintain the data's confidentiality and integrity.

Cryptographic procedures are used to protect sensitive data so that only authorized individuals may decode it. Cryptographic technology encrypts data to create encrypted data and allows for secure transmission, which may be meaningless to an invader who does not know the key. IoT devices have grown in the market, with over 15 billion linked devices anticipated at this moment. IoT devices, like the established systems from which they are derived, are equipped with sensors and communicate in some way [10].

The purpose of IoT devices is to regularly gather, process, send data across a communication channel, and control a large number of larger units. The data in question might include everything from a user's heartbeat to the temperature of a room, living habits, and even their whereabouts.

The rest of the paper is organized as follows: Section 2 presents background and related work. Section 3 presents methodology while Section 4 presents the results and analysis. Finally, a conclusion is presented in Section 5.

2. Background and Related Work

IoT system architecture is eventually defined as a four-stage processing in which data transfers from sensors attached to "things" through a network and to a corporate data center or to the cloud for processing, analyzing, and storing. A "thing" in the IoT can be a machine, a structure, or even a human. In the architecture of IoT, processes transfer data in the other direction in the form of commands or instructions that instruct an actuator or physically connected device to perform some tasks to regulate a physical process. If an approaching malfunction is recognized, an actuator could perform something as easy as turning on a light or as serious as shutting down an assembly line. In addition to device and sensors, IoT architecture layers are distinguished to track the consistency of a system through protocols and gateways [11].

2.1. IoT Architecture. Many of the researchers have offered several architectures, and we can all agree that there is no one consensus on IoT architecture. A four-layer architecture is the most fundamental: perception, network, processing, and applications.

Perception layer: it converts analog signal into digital data and vice versa. It is the initial step of the IoT system, and it encompasses a wide range of "things" or endpoint devices that serve as a link between the physical and digital worlds. They come in a variety of shapes and sizes, ranging from microscopic silicon chips to enormous vehicles. Sensors, actuators, machines, and devices are examples of IoT things that can be grouped into groups based on their functions.

Network layer: it enables data transmission. It is the second level of architecture, and it is responsible of all communications among the IoT infrastructure's devices, networks, and cloud services. There are two methods of connecting the physical layer and the cloud: Direct method using TCP/IP or UDP stack; Using gateways—software or hardware components that handle protocol translation as well as encrypting and decrypting the IoT data.

Processing layer: it is responsible for transforming raw data into useable information. It collects, saves, and analyses information from the previous layer. All of these duties are typically done by IoT systems and are divided into two stages: stages of data collection and data abstraction.

Application layer: it addresses the business requirements. At this layer, software analyzes data to provide solutions to crucial business problems. Many of IoT applications exist, ranging in complexity, functionality, and utilizing of various technologies' stacks and operating systems [11].

2.2. IoT Security. Authenticity, confidentiality, integrity, and availability are common security criteria in any system, and they apply to the Internet of Things as well. IoT has a number of flaws that make security a difficult task, such as the diverse nature of nodes with Internet connectivity and fewer embedded security devices [12]. This section begins with an introduction of security concerns in the IoT environment, followed by a discussion of IoT security requirements and threats, as well as some potential IoT security solutions. Authentication is the process that uniquely identifies the incoming user. It is a critical requirement in the IoT since it is critical to keep data safe from unauthorized devices and people. It is authorized, only users have access to the system and sensitive information [13–15]. The confidentiality is important where personal data between billions of IoT devices and the storage of that data must be secure [16]. If unauthorized access to sensitive information is gained, the components in the Internet of Things that collaborate to offer the intended service are vulnerable to confidentiality assault. The use of an access control technique or a lightweight encryption strategy can maintain confidentiality, which is a fundamental concern. The reliability assures data accuracy and completeness, as well as protecting it from tampering [13]. To assure the message's uniqueness, error detection techniques such as cyclic redundancy check (CRC) might be used. Data must be available to authorized users at all times in the IoT. When data, software services, and hardware are needed, back-end cloud and storage devices must be available. Accessibility to the security service', availability, and continuity must be enhanced to avoid any possible operational disruptions or malfunctions [12]. Because of the unique characteristics of the IoT environment, traditional security techniques are ineffective. Figure 1 depicts the challenges in designing an IoT security system.

The data are exposed to assaults and threats since the items or things in the IoT tend to communicate data autonomously. It is essential that the information be maintained safe and private during the autonomous transfer. While end-to-end information transport is somewhat impervious to attacks, communication across a variety of nodes and sensors is extremely vulnerable to privacy breaches. The vast majority of data shared and collected by various IoT technologies is generally human centric. The amount of information a person or entity is willing to share with others must be verified.

2.3. Lightweight Cryptography. As mentioned above, the IoT devices have a limitation in processor and memory, so these devices require a specific cryptographic method that calls LWC for more easiness and compatibility, by a simple and low computing process. As a result, in order for the LWC to be suited for IoT devices, the block size, key size, number of encryption rounds, and algorithm structured should all be examined [17].

The use of LWC in IoT devices saves both hardware and power usage. There are plethoras of LWC algorithm available nowadays, many expert authors' articles [8, 18, 19] pay attention to them. In the NIST competition, for example, 57 projects were submitted. Ten finalists were chosen in March 2021 [20]. When we look at these algorithms, we notice that they all use the cyclic cipher idea. We decided to take a different route.

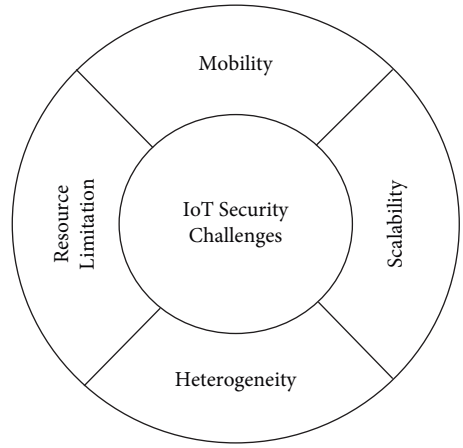

Figure 1: IoT security challenges. A circle represents the elements of the security challenge.

DNA: adenine (A), thymine (T), cytosine (C), and genuine (G) are the four bases of deoxyribonucleic acid (DNA) [21, 22]. According to the Watson–Crick Model, all A and T bases complement each other. The bases C and G complement each other [22–25], where A stands for binary value 00 (decimal value 0), C for binary value 01 (decimal value 1), G for binary value 10 (decimal value 2), and T for binary value 11 (decimal value 3) [24] as demonstrated in Table 1. For each character created using DNA sequences, the suggested method employs complimentary rules. The base pairs are covered by the complementary rule, adenine and thymine can form a pair, whereas cytosine and genuine can form a second pair [21, 25–27], as shown in Table 1. The DNA XOR operation between these bases is shown in Table 2 [24–26, 28].

DNA computing uses a random technique that improves the complexity and security of the encryption, ensuring that the data are well protected from hackers. Another feature of DNA computing is its ability to process data quickly while requiring minimum power and storage. This is evident when encoding plain data with DNA sequences [29, 30], the researchers use DNA either directly or indirectly by utilizing DNA characteristics, hybrid cryptography combines both methods execution in order to improve the security of classical cryptography [31, 32]. To give better data security, most researchers applied DNA by transforming the cipher text to DNA tape, such as encrypting the data using AES and converting it to ASCII code and their equivalent hexadecimal then binary formats, respectively, and lastly DNA tape [33]. Other researchers have employed DNA to improve the security of traditional encryption algorithms such as AES and RSA [34].

The suggested DNA encryption algorithm works by creating a strong and entirely random key for data encryption using the DNA tape. Because of the randomness in generating the encryption key, logical substitution, and a set

Table 1: DNA letters coding.

DNA base	CODE
A	00
C	01
G	10
T	11

Table 2: DNA XOR operation.

XOR	A	T	C	G
A	A	T	C	G
T	T	A	G	C
C	C	G	A	T
G	G	C	T	A

of rules for transposition, this technique is known for its great encryption robustness and strength.

In comparison with the other techniques, the proposed DNA approach has the shortest encryption time, recording around 4.2500 Sc. and 4.9211 Sc. for decryption.

Secure IoT (SIT): it encrypts data with n-bits block cipher and requires n-bit key. The algorithm's architecture is a hybrid of festal and uniform substitution-permutation networks [35]. The summary of the linked works is shown in Table 3 [36].

In our research, a new lightweight encryption algorithm was proposed based on the computing of DNA sequences that is suitable for IoT devices' computation resources. The suggested algorithm's key generation is fully random and based on the DNA sequence, making it extremely hard to crack. Furthermore, depends on the unpredictability character of the DNA sequence and its robustness that satisfies the capabilities of IoT computing, the created key is used to make a logical, simple, and solid confusion and diffusion to the plain images.

3. Methodology

Because DNA cryptography is a rapidly evolving and promising sector in data security, we provide a new DNA-based encryption model in this article. The unpredictability of the DNA tape was used to provide a strong encryption and decryption key that could be employed in symmetric ciphering applications. Due to its strong quality, DNA has been used in this algorithm. The encryption technique is robust and difficult to hack because it is based on high randomness, and it is also may be ideal for IoT devices with limited RAM and CPU, that uses the DNA tape according to the sequence binary string representation in addition to the key generation process.

This approach provides some lightweight encryption standards such as less complexity, robust architecture, high throughput, less execution time, less memory requirement, and good immunity against linear and differential attacks. The approach suggested in this paper is a new lightweight cryptographic algorithm based on DNA computing, which includes collecting data from IoT devices then convert it to text and choose a DNA cassette for creating a secret key, the next step is the divided source data will be converted to text and will be encrypted into two bytes at a time as well the segment and the input DNA cassette in a specific order; for each secret key, extract 16 bits-secret key from the DNA tape letters, apply XOR technique to perform a substitution between each segment, and the DNA secret key and then apply a series of rules based on DNA tape to the XOR operation results and transpose them. Repeat the entire operation until the DNA tape is finished and the data is entirely encrypted, calculate the performance of the proposed encryption approach and compare the archived results with the result of previous methods.

The suggested lightweight encryption algorithm based on DNA computing that achieves the following objectives:

(1) Capable of encrypting and decrypting image files and text

(2) Does not necessitate a large number of resources (memory space and processor time)

(3) Achieves a high level of data security during transmission; this can be accomplished by changing the key used on a regular basis

The proposed approach's flowchart is shown in Figure 2, and the series implemented in number of stages can be summarized as follows (Algorithm 1):

4. Result and Analysis

For the suggested cryptographic paradigm, three key goals have been established. To test the suggested encryption scheme, we will assume the source data are photos in this part. First and foremost, the encryption technique must be light enough to run on the IoT device processor. This means that as few resources as feasible (processor time and memory) are used. Second, to ensure a high level of safety for the information conveyed, the key used in the encryption method must be changed on a regular basis. Third, the key size (in bits) utilized in the encryption technique should be as large as feasible to make it difficult for attackers to break. Fourth, the encryption algorithm must result in the largest amount of data distortion. This effect can be quantified by calculating the encrypted image's peak signal-to-noise ratio (PSNR) and statistically by comparing the source and encrypted images' histograms.

The proposed cryptographic solution was implemented utilizing a computer system with an Intel (Core-i5) 2.50 GHz CPU and 8.0 GB RAM and a MATLAB programming (R2019a) version. The suggested cryptographic approach's objectives are studied and tested in this section. Each test is evaluated by comparing it to previously published methods.

A discussion of the findings aided in the formulation of some conclusions.

The proposed approach is implemented on images and text files.

Table 3: Summary of related work according to text file.

Cipher	Device	Block size (bit)	Key size (bit)	Code size (byte)	RAM (byte)	Encrypted-key schedule	Encryption (cycles)	Decryption (cycles)
AES	AVR	128	128	23464	720	2424	5225	5242
RC5	AVR	64	128	20444	360	30744	5244	5239
PRINCE	AVR	64	128	23838	176	675	7044	7047
HIGHT	AVR	64	128	13716	288	1615	3459	3543
LBLOCK	AVR	64	80	23718	306	4824	4772	4799
PICCOL	AVR	64	80	1534	126	1563	12630	12709
LILLIPUT	AVR	64	80	3908	276	12778	10934	11424
TWINE	AVR	64	80	2204	214	5047	10303	10183
RoadRunneR	AVR	64	80	1426	142	967	3658	3682
LED	AVR	64	80	4108	358	369	66950	71061

4.1. Result Analysis

4.1.1. Image Encryption Result

(1) Histogram Analysis. The image's histogram represents the number of pixels that paint the image. Image histogram-analyzing aids in determining the quality of image encryption. The uniform distribution should be present in a ciphered image histogram; Figures 3 and 4 depicts some concentrated values for the plain image, whereas the ciphered images have more flat values, indicating the suggested system can withstand statistical attacks.

(2) Key Space and Sensitivity Metrics. The key space reveals that all available keys have been used. Here, chaotic sequences are created and employed in conjunction with precision values of 10–15 to achieve accurate refinement, resulting in a larger key space of $(1015)6 = 1090 = 2298$, making this strategy resistant to brute force and dictionary assaults. The term "key sensitivity" relates to how much a change in the key can affect the ability to generate a ciphered picture. Again, factors like NPCR (number of pixel changing rate) and UACI (unified average changing intensity) can be used to determine this. Even a tiny change in the key might bring out more diffusion or permutation in an image, therefore a smart technique is always sensitive. As a result, the proposed strategy is said to be resistant to differential and statistical attacks.

(3) Correlation Coefficient Analysis. The adjacent pixel value depicts the relationship between two pixels that are next to each other. The C correlation coefficients should be computed horizontally, vertically, and diagonally between two neighboring pixels as follows:

$$C_{X,Y} = \frac{\mathrm{Cov}\,(X, Y)}{\sqrt{D(X)D(Y)}}, \tag{1}$$

where X and Y are adjacent pixels.

Cov (X, Y) is the covariance between two pixels X and Y. It is given as follows:

$$\mathrm{cov}(X, Y) = \left(\frac{1}{N}\right)\sum_{i=1}^{N}\left(X_i - E(X)\right)\left(Y_i - E(Y)\right), \tag{2}$$

where $E(X) = (1/N)\sum_{i=1}^{N} X_i$

$$D(X) = \left(\frac{1}{N}\right)\sum_{i=1}^{N}\left(X_i - E(Y)\right)^2. \tag{3}$$

This analysis includes calculating three adjacent pixel's correlation for each plain cipher image: vertically, horizontally, and diagonally. Figure 5 represents the horizontal importance of neighboring elements in the image before and after encryption. It reveals a dramatic drop in the importance of nearby elements.

(4) Information Entropy. Entropy is the most important property of a disorder, or more accurately, unpredictability, according to information theory, is a metric that assesses the unpredictability of an image:

To find the entropy $H(X)$ of a source X, use the following formula:

$$H(x) = -\sum_{i=0}^{255} P(X_i) * \log P(X_i). \tag{4}$$

A resilient encryption method has an entropy value of 8 in theory. The information entropy of plain and encrypted photos $(256 * 256)$ is shown in the table below. The last equation calculates the entropy. Because the findings were so near to 8, they were considered satisfactory. The information entropy of several systems is compared in Table 4.

Other algorithms were outperformed by the proposed algorithm, which is close to the value 8, as shown in Table 5.

(5) Different Attacks and Chosen Plaintext Attack. Two indicators are employed in the analyses of a different attack to inspect on the influence of a one-bit difference between the original image and the ciphered one, NPCR and UACI, or the number of pixels change rate and united average changing intensity, are the acronyms for the number of pixels changing rate and unified average change intensity, respectively.

They are calculated as follows:

$$\mathrm{NPCR} = \frac{\sum_{i,j} D(i, j)}{W * H} * 100\%,$$

$$\mathrm{UACI} = \frac{\sum_{i,j} C1(i, j) - C2(i, j)}{255} * 100\%, \tag{5}$$

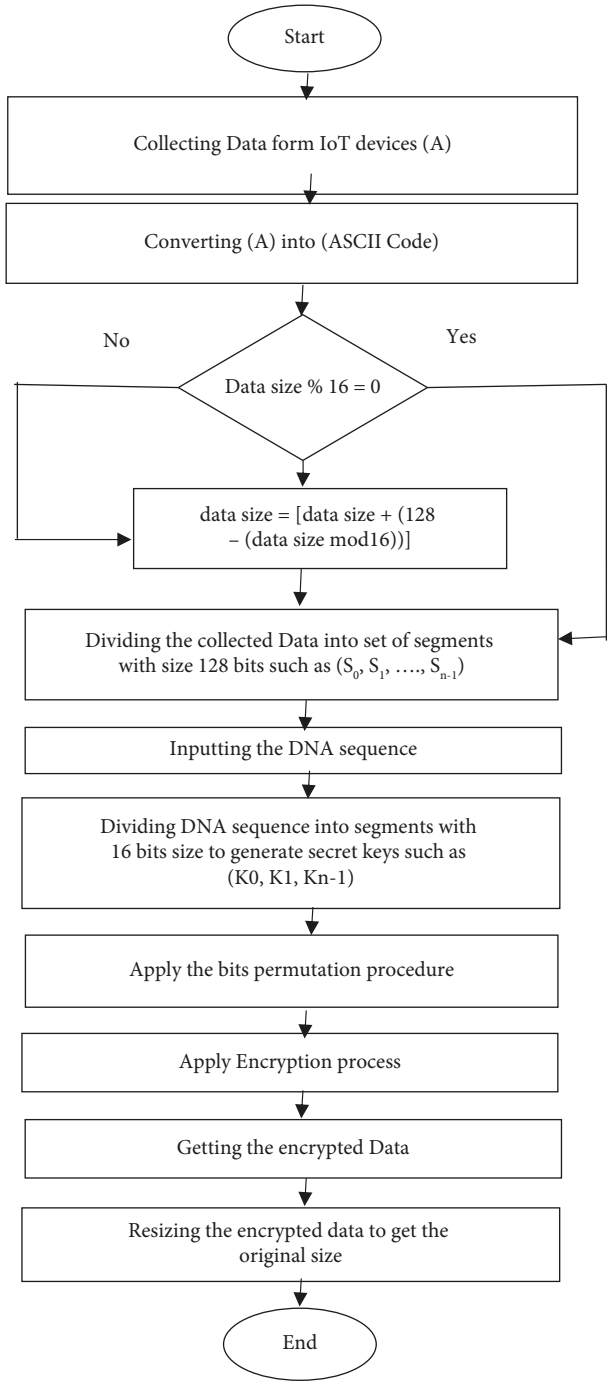

Figure 2: Flowchart of DNA encryption algorithm.

where H and W are the height and width of the ciphered image, respectively. $C1(i, j)$ is the encrypted image before the change in one pixel of the plain image and $C2(i,j)$ is the image after the change.

Values of NPCR and UACI are shown in Table 6, both are close to the optimum values. The obtained result values were 33.45% for UACI and 99.62% for NPCR. This proves that the algorithm exhibits high sensitivity towards changes in the original image, even if they are quite small. This means it can resist different types of attacks.

The software is given in the phrase "Welcome to My Module of Encryption," and the encryption and decryption results are as follows:

Input Text: Welcome to My Module of Encryption

Encrypted Text:

AGGTAGCCCTGTTCCGTTAGGTTTAATATCTGG
TGTTAGTTGCTCGGACCGATATGGTACGTA
GACTCAAATGCCTTGGTATCGAAGGCACGC
TTCTTCGGCTTCTACGACCTGCCCCCCTGTCCA
GATATTCTGTATACGCACGATA

Return Text: Welcome to My Module of Encryption

The message in this case contains a variety of characters such as capital and small letters, numbers, and special characters, and the encryption result length is appropriate, so no additional insertion bits are required. As a result, within the user-generated sequence, the user must utilize a segment length of one.

With a text sample of "Health," the proposed model for text encryption utilizing DNA sequences is demonstrated. The key will be computed first, and then it will be utilized to generate the final DNA sequences. Meanwhile, to get encrypted text, DNA encoding rules, as well as single-point crossover, mutation, and complementary rules, are applied. The same operation is repeated and reversed during the decryption process, using the same key value.

Therefore, when there is a new session starting the communications with new text input as the previous one health occurs, it generates a key value that is different from the previous session's key. As a result, the proposed method will have a high level of dynamicity and randomness. It is tough to obtain the key computationally because the keys generated and utilized during the various sessions/transactions were different. As a result, an attacker will be unable to obtain the plaintext. Table 7 displays the time necessary for encryption and decryption processes in seconds. Encryption and decryption times are frequently found to be comparatively close and take less time.

Table 7 displays the time necessary for encryption. Therefore, when there is a new session starting the communications with new text input as the previous one health occurs, it generates a key value that is different from the previous session's key. As a result, the proposed method will have a high level of security and communication.

Table 8 demonstrates the suggested DNA cryptographic method's memory allocation (in bytes) for disk encryption.

Table 9 shows the comparison of the proposed method and previous various lightweight algorithms and the suggested approach are displayed. The disparities are dependent on RAM, file size, and the quantity of encryption and

(1) Collecting data from IoT devices
(2) Converting the collected data into text of ASCII (American Standard Code for Information Interchange) code
(3) Checking the data size (counting the bits of data)
(1) if the data size mod 16 = 0 go to 4
(2) else do data size = [data size + (16 – (data size mod 16))]
(4) Dividing the collected data into a set of segments with size 16 bits such as $(S_0, S_1, \ldots, S_{n-1})$
(5) Generating the DNA sequence from imputing data
(i) Dividing the DNA sequence into a set of segments with size 128 bits to generate secret keys such as (K_0, K_1, K_{n-1})
(6) Apply the bit of permutation procedure
(a) The permutation procedure for 16 successive bits such as the following:
$\mathbf{S_0(0)}, \ldots \mathbf{S_0(15)}, \mathbf{S_1(0)}, \ldots \mathbf{S_1(15)}, \ldots\ldots\ldots \mathbf{S_{15}(0)}, \ldots \mathbf{S_{15}(15)}$
(b) The permutation procedure is as follows
Start
If $S_0(0) = S_0(1)$ then bit(1)↔bit(2)
If $S_0(1) = S_0(2)$ then bit(2)↔bit(3)
⋮
If $S_0(14) = S_0(15)$ then bit(14)↔bit(15)
⋮
⋮
If $S_{15}(0) = S_{15}(1)$ then bit(1)↔bit(2)
If $S_{15}(1) = S_{15}(2)$ then bit(2)↔bit(3)
If $S_{15}(14) = S_{15}(15)$ then bit(14)↔bit(15)
End
(7) Apply encryption process for each segment S by swapping the first four bits of $(S_0, \ldots, S_{n-1})$ with the last four bits of $(K_0, \ldots, K_{n-1})$, respectively
(8) Getting the encrypted data
(9) Resizing the encrypted data to get the original size

Algorithm 1: Steps of encryption algorithm based on DNA.

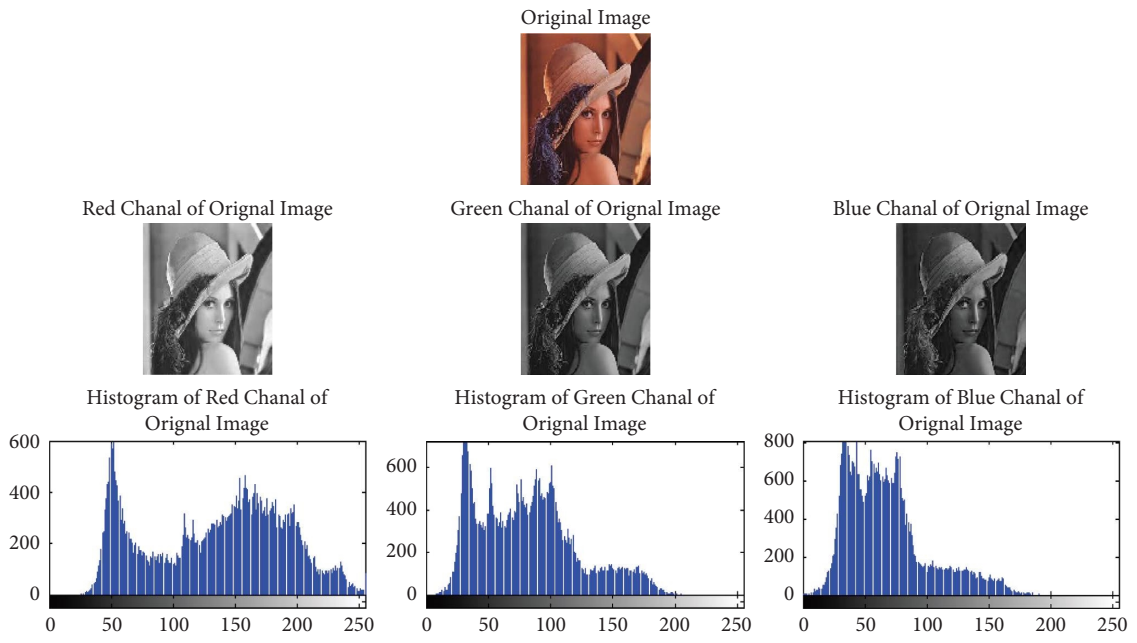

Figure 3: Original image of lena and its RGB histogram.

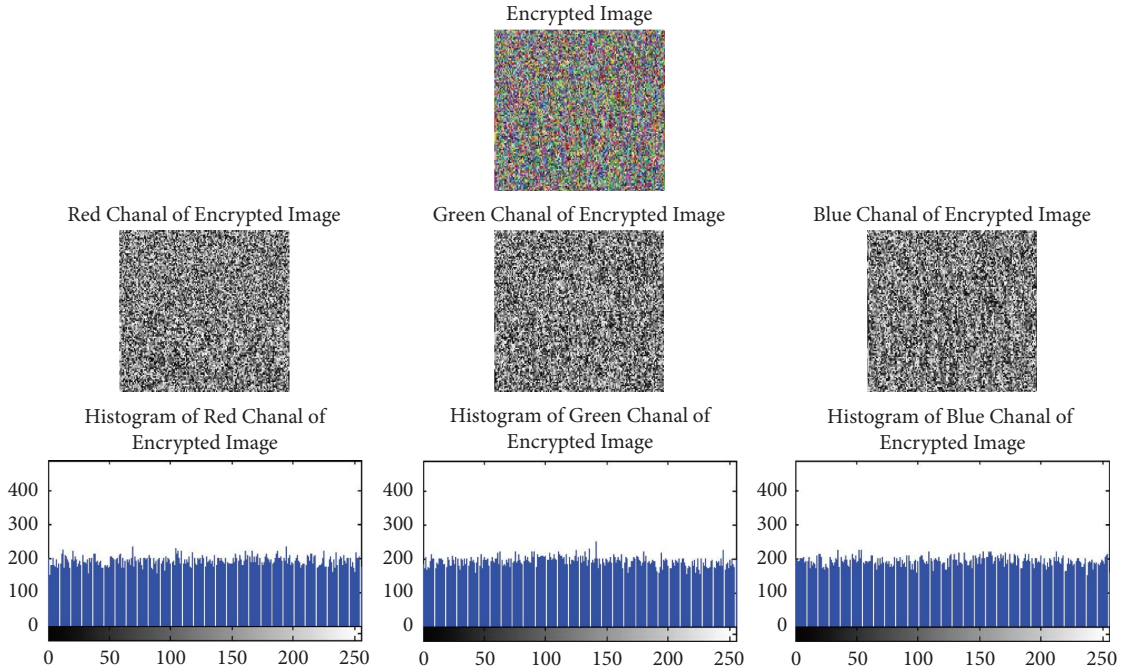

Figure 4: Encrypted image of lena and its RGB histogram.

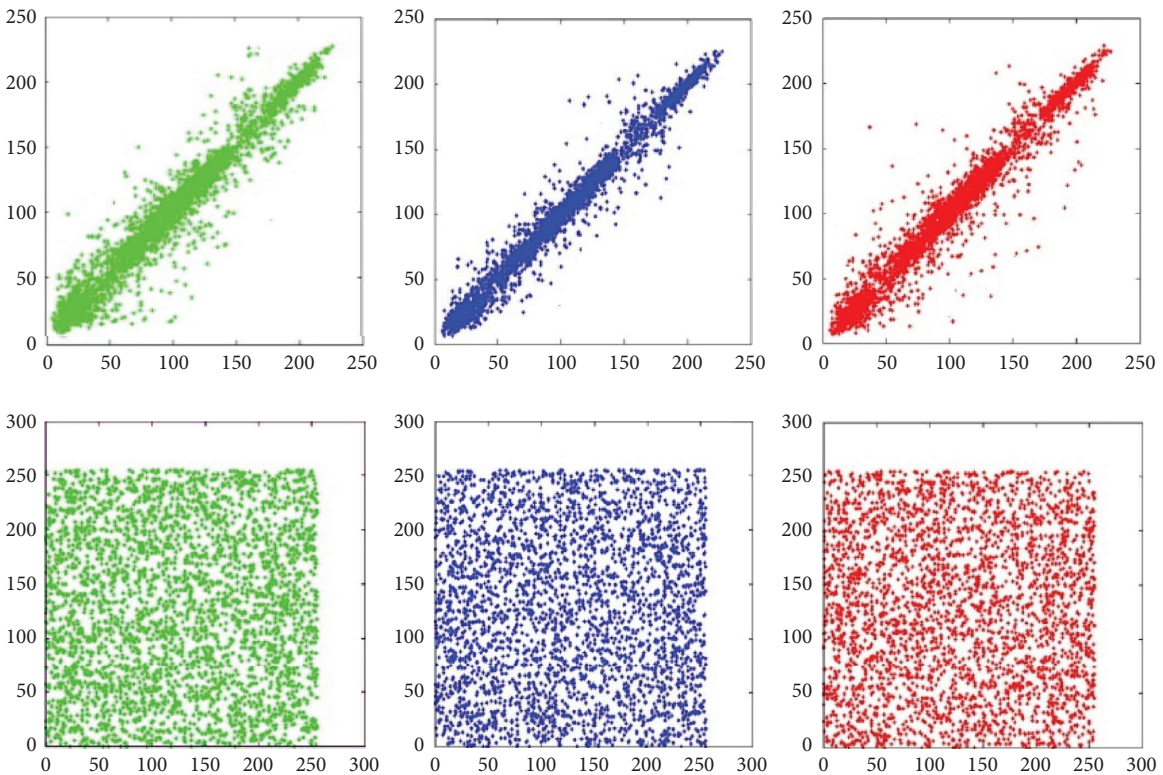

Figure 5: Correlation of two horizontally adjacent pixels in the plain and encrypted image of lena.

Table 4: Information entropy of encryption of some plain images.

Images	Plain images			Encrypted images		
	R	G	B	R	G	B
Pepper	7.3009	7.5570	7.0929	7.9974	7.9973	7.9972
Lena	7.1655	7.5578	6.8571	7.9985	7.9982	7.9984
Baboon	7.6987	7.4251	7.5809	7.9970	7.9970	7.9971

Table 5: Comparison of information entropy across methods about image lena.

Algorithm	Information entropy		
	R	G	B
The proposed	**7.9985**	**7.9982**	**7.9984**
[37]	7.9973	7.9972	7.9969
[38]	7.9966	7.9972	7.9967
[39]	7.9973	7.9969	7.9971
[40]	7.9974	7.9976	7.9975

Bold values are the highest results that we got from the proposed system.

Table 6: Compared values of UACI and NPCR.

Algorithms		Images		
		Pepper	Lena	Baboon
Proposed	UACI	33.46	33.44	33.45
	NPCR	99.62	99.64	99.60
[41]	UACI	33.32	33.46	33.55
	NPCR	99.58	99.67	99.60
[42]	UACI	33.46	33.44	33.46
	NPCR	99.61	99.62	99.60
[43]	UACI	33.52	33.50	33.47
	NPCR	99.60	99.61	99.61
[44]	UACI	33.46	33.44	33.49
	NPCR	99.60	99.62	99.62

Table 7: Required time for encryption and decryption.

Case	Encryption time (sec)	Decryption time (sec)
Single-word	0.0285	0.0357
Multiwords	0.0320	0.0476

Table 8: Memory requirements for encryption.

Input text size (byte)	Encrypted text size (byte)	Decrypted text size (byte)
4	4.08	4
36	36.32	3

Table 9: Propose method compared with different ciphers in terms of code size, RAM, and encryption/decryption (cycles) that implemented on AVR device.

Terms	Algorithms						
	AES	RC5	PRINCE	HEIGHT	[44]	[45]	Proposed method
Block size (bit)	128	64	64	64	64	64	16
Key size (bit)	128	128	128	128	80	64	64
Code size in byte	23464	20444	23838	13716	1364	2094	1687
RAM	720	360	176	288	18	16	17
Encrypted key schedule	2424	30744	675	1615	1407	1218	1038
Encryption (cycles)	5225	5244	7044	3459	3359	1401	975
Decryption (cycles)	5242	5239	7047	3543	3434	918	788

decryption cycles. The table shows that the suggested method outperforms the others by operating with less code, RAM, loops, and the quantity of encrypting and decrypting cycles.

5. Conclusions

Security in the Internet of Things is still a hot topic for research. It has aroused a great deal of scientific interest recently. Devices in the Internet of Things (IoT) network have limited resources and are low powered and underwhelming. When evaluating their resource restrictions, factors such as battery life, processing power, and memory footprint are all taken into consideration. There will still be options for creating fresh solutions and changing current security precautions. This is because there needs to be a balance between the security strength and the limitations of the outmoded Internet and resource-constrained IoT devices. For IoT devices, a lightweight method is devised and implemented in this work. For the purpose of protecting data transit for IoT devices, the suggested encryption strategy also considers IoT device restrictions in terms of processor speed, memory size, and power consumption. According to the findings of the experiments, the suggested method achieves less processing time and memory space than existing methods while providing a high level of security of the communicated data through a constant change of the key used for encrypting the transmitted IoT data. Furthermore, the suggested architecture's key size for encrypting transmitted data is sufficiently large to be difficult for adversaries to crack. The security analysis and performance evaluation show that the suggested approach offers great security and is suited for the resource-constrained nature of IoT. This study provides a number of significant advances to the field of IoT security, the research shows the effectiveness of lightweight blocks ciphers is affected by using the best key size and the good memory consumption results from using the lightweight encoder blocks at their ideal size. The key schedules with simple cycles use less electricity and have fewer encoding and decoding cycle; the proposed research also reduced the number of encoding and decoding cycles and energy usage by using straightforward mathematical operations to reduce the number of rounds for encoding and decoding, using the DNA approach and transforming different types of data into ASCI code enhances the complexity of the encryption, it suggested algorithm offers reliable, tamper-proof, and lightweight information transfer in IoT applications at cheap cost. Furthermore, the suggested encryption model takes into account the processing time, memory space, and power consumption limitations of IoT devices. Our proposed model achieves less processing time and less memory space than existing ways while ensuring a high level of security of the communicated data through a constant change of the key used for encrypting the transmitted IoT data, according to the experimental results. Furthermore, the key size utilized to encrypt the transmitted data in the proposed architecture is big enough to make it difficult for attackers to breach.

Data Availability

The data used to support the findings of this study are included within the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. Shargabi and A. Husainy, "A new DNA based encryption algorithm for internet of things," in *Proceedings of the International Conference of Reliable Information and Communication Technology IRICT 2020: Innovative Systems for Intelligent Health Informatics*, pp. 786–795, Berlin, Germany, June 2021.

[2] M. A. Khan and K. Amit, "Scalable design and processor technology for iot applications," in *Internet of Things: A Hardware Development Perspective*, M. A. Khan, Ed., CRC Press, Boca Raton, FL, USA, 1st edition, 2022.

[3] N. Saleh Alghamdi and M. Ayoub Khan, "Energy-efficient and blockchain-enabled model for internet of things (IoT) in smart cities," *Computers, Materials & Continua*, vol. 66, no. 3, pp. 2509–2524, 2021.

[4] A. Munusamy, M. Adhikari, M. A. Khan et al., "Edge-centric secure service provisioning in IoT-enabled maritime transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, pp. 1–10, 2021.

[5] I. Hussain, M. C. Negi, and N. Pandey, "A secure IoT-based power plant control using RSA and DES encryption techniques in data link layer," in *Proceedings of the International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, pp. 464–470, IEEE, Dubai, United Arab Emirates, December 2017.

[6] Y. Yang, "ASTREAM: data-stream-driven scalable anomaly detection with accuracy guarantee in IIoT environment," *IEEE Transactions on Network Science and Engineering*, 2022.

[7] M. Ayoub Khan and Y. P. Singh, "On the security of joint signature and hybrid encryption," in *Proceedings of the 2005 13th IEEE International Conference on Networks Jointly Held with the 2005*, November 2005.

[8] M. A. Khan and N. S. Alghamdi, "A neutrosophic WPM-based machine learning model for device trust in industrial internet of things," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 4, pp. 3003–3017, 2021.

[9] H. Abualese, T. Al-Rousan, and B. Al-Shargabi, "A new trust framework for EGovernment in cloud of things," *International Journal of Electronics and Telecommunications*, vol. 65, pp. 397–405, 2019.

[10] M. Khalifa, F. Algarni, M. Ayoub Khan, A. Ullah, and K. Aloufi, "A lightweight cryptography (LWC) framework to secure memory heap in Internet of Things," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 1489–1497, 2021.

[11] I. Architecture, "The pathway from physical signals to business decisions," 2020, https://www.altexsoft.com/blog/iot-architecture-layers-components/.

[12] M. A. Khan and K. A. Abuhasel, "Advanced metameric dimension framework for heterogeneous industrial Internet of things," *Computational Intelligence*, vol. 37, pp. 1367–1387, 2021.

[13] M. Dammak, O. R. M. Boudia, M. A. Messous, S. M. Senouci, and C. Gransart, "Token-based Lightweight Authentication to Secure IoT Networks," in *Proceedings of the 16th IEEE Annual*

Consumer Communications & Networking Conference (CCNC), IEEE, Las Vegas, NV, USA, January 2019.

[14] W. U. Khan, X. Li, A. Ihsan, M. A. Khan, V. G. Menon, and M. Ahmed, "NOMA-enabled optimization framework for next-generation small-cell IoV networks under imperfect SIC decoding," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 22442–22451, 2022.

[15] M. A. Khan and K. A. Abuhasel, "An evolutionary multi-hidden Markov model for intelligent threat sensing in industrial internet of things," *The Journal of Supercomputing*, vol. 77, no. 6, pp. 6236–6250, 2021.

[16] Z. Kubba and H. Hoomod, "Developing a lightweight cryptographic algorithm based on DNA computing," *AIP Conference Proceedings*, vol. 2290, no. 1, Article ID 40013, 2020.

[17] M. Usman, "Lightweight encryption for the low powered iot devices," 2021, https://arxiv.org/abs/%202012.00193.

[18] S. Nandy, M. Adhikari, M. A. Khan, V. G. Menon, and S. Verma, "An intrusion detection mechanism for secured IoMT framework based on swarm-neural network," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 1969–1976, 2022.

[19] C. gov, "Lightweight cryptography finalists," 2021, https://csrc.nist.gov/Projects/lightweightcryptography/%20finalists.

[20] S. Mumthas and A. Lijiya, "Transform domain video steganography using RSA, random DNA encryption and huffman encoding," *Procedia Computer Science*, vol. 115, pp. 660–666, 2017.

[21] S. K. Pujari, G. Bhattacharjee, and S. Bhoi, "A hybridized model for image encryption through genetic algorithm and DNA sequence," *Procedia Computer Science*, vol. 125, pp. 165–171, 2018.

[22] N. Sasikaladevi, K. Geetha, and A. Revathi, *Emote – Multi-layered Encryption System For Protecting Medical Images Based On Binary Curve*, Journal of King Saud University – Computer and Information Sciences, Riyadh, Saudi Arabia, 2019.

[23] J. Chen, Z. Zhu, L. Zhang, Y. Zhang, and B. Yang, "Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption," *Signal Processing*, vol. 142, pp. 340–353, 2018.

[24] T. Li and M. Yang, "A novel image encryption algorithm based on a fractional-order hyperchaotic system and dna computing," *Hyperchaotic Fractional-Order Systems and Their Applications*, Hindawi Complexity, vol.2017, Article ID 9010251, 13 pages, 2017.

[25] R. Enayatifar, F. G. Guimarães, and P. Siarry, "Index-based permutation-diffusion in multiple-image encryption using DNA sequence," *Optics and Lasers in Engineering*, vol. 115, pp. 131–140, 2019.

[26] X. Wu, K. Wang, X. Wang, and H. Kan, "Lossless chaotic color image cryptosystem based on DNA encryption and entropy," *Nonlinear Dynamics*, vol. 90, no. 2, pp. 855–875, 2017.

[27] Y. Bhavani, S. S. Puppala, B. J. Krishna, and S. Madarapu, "Modified aes using dynamic s-box and dna cryptography," in *Proceedings of the Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, Palladam, India, December 2019.

[28] M. Sohal and S. Sharma, *BDNA-A DNA Inspired Symmetrickey Cryptographic Technique to Secure Cloud Computing*, Journal of King Saud University – Computer and Information Sciences, Riyadh, Saudi Arabia, 2018.

[29] Y. Wu, L. Zhang, S. Berretti, and S. Wan, "Medical image encryption by content-aware DNA computing for secure healthcare," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, p. 2089, 2023.

[30] H Al, "A new lightweight proposed cryptography method for IoT," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 4954–4958, 2020.

[31] A. E. El-Moursy, M. Elmogy, and A. Atwan, "DNA-based cryptography: motivation, progress, challenges, and future," *Journal of Software Engineering and Intelligent Systems*, vol. 3, no. 1, pp. 67–82, 2018.

[32] P. Radoglou-Grammatikis, K. Rompolos, P. Sarigiannidis et al., "Modeling, detecting, and mitigating threats against industrial healthcare systems: a combined software defined networking and reinforcement learning approach," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 2041–2052, 2022.

[33] N. S. Kolte, K. V. Kulhalli, and S. C. Shinde, "DNA cryptography using index-based symmetric DNA encryption algorithm," *International Journal of Engineering Research and Technology*, vol. 10, pp. 810–813, 2017.

[34] A. S. Pradeeksha and S. S. Sathyapriya, "Design and implementation of dna based cryptographic algorithm," in *Proceedings of the 2020 5th International Conference on Devices, Circuits and Systems (ICDCS)*, Coimbatore, India, March 2020.

[35] Al-Ahdal, Al-Rummana, and Deshmukh, "Security analysis of a robust lightweight algorithm for securing data in internet of things networks," *Turkish Journal of Computer and Mathematics Education*, vol. 12, pp. 133–143, 2021.

[36] Y. Zhang and D. Xiao, "Self-adaptive permutation and combined global diffusion for chaotic color image encryption," *AEU - International Journal of Electronics and Communications*, vol. 68, no. 4, pp. 361–368, 2014.

[37] A. U. Rehman, X. Liao, R. Ashraf, S. Ullah, and H. Wang, "A color image encryption technique using exclusive-OR with DNA complementary rules based on chaos theory and SHA-2," *Optik*, vol. 159, pp. 348–367, 2018.

[38] X. Chai, X. Fu, Z. Gan, Y. Lu, and Y. Chen, "A color image cryptosystem based on dynamic DNA encryption and chaos," *Signal Processing*, vol. 155, pp. 44–62, 2019.

[39] S. Zhou, P. He, and N. Kasabov, "A dynamic DNA color image encryption method based on SHA-512," *Entropy*, vol. 22, no. 10, p. 1091, 2020.

[40] B. Idrees, S. Zafar, T. Rashid, and W. Gao, "Image encryption algorithm using S-box and dynamic Hénon bit level permutation," *Multimedia Tools and Applications*, vol. 79, no. 9-10, pp. 6135–6162, 2020.

[41] A. G. Mohamed, N. O. Korany, and S. E. El-Khamy, "New DNA coded fuzzy based (DNAFZ) S-boxes: application to robust image encryption using hyper chaotic maps," *IEEE Access*, vol. 9, pp. 14284–14305, 2021.

[42] E. Zarei Zefreh, "An image encryption scheme based on a hybrid model of DNA computing, chaotic systems and hash functions," *Multimedia Tools and Applications*, vol. 79, 2020.

[43] X. Chai, Z. Gan, K. Yuan, Y. Chen, and X. Liu, "A novel image encryption scheme based on DNA sequence operations and chaotic systems," *Neural Computing & Applications*, vol. 31, no. 1, pp. 219–237, 2019.

[44] A. H. A. Al-Ahdal, G. A. Al-Rummana, and N. K. Deshmukh, "A robust lightweight algorithm for securing data in internet of things networks," in *Sustainable Communication Networks and Application*, P. Karuppusamy, I. Perikos, F. Shi, and T. N. Nguyen, Eds., Springer, Singapore, 2021.

[45] A. H. A. Al-Ahdal, "A new lightweight block cipher design for securing data in IoT devices," *International Journal of Computer Science and Engineering*, vol. 8, no. 10, 2020.