Computers and Electrical Engineering 95 (2021) 107418

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Lightweight cryptography system for IoT devices using DNA

Mohammed Abbas Fadhil Al-Husainy [a], Bassam Al-Shargabi [a,*], Shadi Aljawarneh [b]

[a] Faculty of Information Technology, Middle East University, Amman, Jordan
[b] Faculty of Computer and Information Technology, Jordan University of Science and Technology, Irbid, Jordan

ARTICLE INFO

Editor: Dr. M. Malek

Keywords:
Internet Of Things
Lightweight encryption
Random sequence
Security
Data encryption
Confusion and diffusion

ABSTRACT

Many lightweight encryption algorithms have been developed for tackling the limitations of resources on the Internet of Things (IoT) devices. Such devices have limited processing capabilities in terms of speed, storage, and memory. In this paper, we designed and implemented a flexible lightweight encryption system with strong and simple substitution, and transposition operations to encrypt and decrypt data that meets limited processing capabilities within IoT devices. We used a variable block size to make the proposed system more flexible to be implemented on various IoT devices that have different memory sizes. Besides, the DeoxyriboNucleic Acid (DNA) sequence is utilized to generate random encryption keys that make it hard to break by the criminals. The experimental results of the proposed lightweight encryption system show promising results to be used for any IoT device with respect to memory size and encryption time compared to well-known cryptographic systems. This is achieved by using variable block size to minimize the size of memory use and minimizing the encryption time by using logical and rotate operations. For fulfilling the security goals, the developed lightweight encryption system has shown a better avalanche effect value which is over 50% compared to other existing encryption systems. The avalanche effect proves that such system is capable to secure IoT data against real-time attacks.

1. Introduction

The Internet of Things (IoT) spread out rapidly and is foreseen to grow increasingly throughout the next couple of years. It is anticipated that by 2025 there will be roughly more than 25 Billion associated IoT devices or sensors [1] . As expected that IoT devices and their applications will reach and connect every aspect of our daily life, such devices have the capacity for moving and producing data over a network without the need for human intervention. Nowadays, with the major advancement in IoT enabling technologies beginning with Radio Frequency Identification system (RFID), connectivity, Cloud, and Big data analytics have been introduced in many fields and applications such smart homes, smart cities, water, electricity, green energy, traffic congestion, waste management, disaster alerting, recycling, agriculture, breeding, and healthcare.

The generated data from these devices or sensors might contain sensitive or private data, such as healthcare records for a patient, images of people's faces, and the vehicle plate numbers in checking zones generated from IoT surveillance cameras. All this led to risen the importance of security and privacy of such data. Besides, the vulnerability of IoT devices can be also be exploited and used as bots for Distributed Denial of Service (DDoS) attacks [2–4]. IoT devices need to be secured and their data must be protected from unauthorized access. Therefore, strong and enhanced encryption algorithms should be considered to secure the transmission of such

This paper is for special section VSI-bioc. Reviews processed and recommended for publication by Guest Editor Dr. Xiaochun Cheng.

* Corresponding author.
E-mail addresses: bassam20_152@yahoo.com (B. Al-Shargabi), saaljawarneh@just.edu.jo (S. Aljawarneh).

https://doi.org/10.1016/j.compeleceng.2021.107418
Received 13 August 2020; Received in revised form 13 January 2021; Accepted 27 August 2021
Available online 6 September 2021

sensitive data. Furthermore, the problem of using conventional encryption techniques for securing the data transmission, such as the Advanced Encryption Standard (AES), hashing, Rivest–Shamir–Adleman (RSA), and Elliptic Curve Cryptography (ECC) [4], is that these techniques are only suited the systems that have a reasonable capabilities in terms of power, memory, and processing compared to the IoT devices or sensors where such capabilities do not fit to the limited devices or constraints of IoT devices [5–8].

A large number of lightweight cryptography primitives have been proposed over the conventional methods reached over 100 lightweight encryption algorithms [8,9]. The innovation and implementation of lightweight cryptographic techniques face many challenges due to the memory size, throughput or speed, and power consumption of IoT devices [10]; in addition to the flexibility to implement the lightweight cryptographic system on different platforms for IoT devices. All of this should not reduce the level of data security [11].

Nevertheless, the improvement and implementation of such algorithms continue to move forward; however, it remains a challenge to find suitable algorithms that match the specific needs of the IoT applications [12]. In addition, there must be a way for creating a flexible lightweight encryption algorithm that is based on using simple and strong logical operations in terms of substitution and transposition to encrypt and decrypt data to fit IoT devices computational resources. Also, the lightweight encryption algorithm must use a simple method for keys generation with keeping these keys hard to break by the attackers. Furthermore, the use of a flexible lightweight encryption algorithm that uses various block sizes of data to make the system more flexible to implement it on different memory sizes of IoT devices.

In this paper, we introduced a flexible lightweight encryption system based on a simple and strong substitution and transposition to encrypt and decrypt data generated from any type of IoT devices. In addition, the proposed lightweight encryption system is based on the use of variable block sizes of data to fit the limitations of the IoT device's memory and processing capabilities. Furthermore, to achieve a high level of security in the proposed system, we used the DNA sequence to generate three primary cryptographic keys that are extracted based on predetermined indices set by the user on the DNA sequence.

The rest of the paper is organized as follows. The most recent related work discussed in Section 2, the design of proposed encryption systems presented in Section 3. In Section 4, performance evaluation and experimental results are provided and discussed. Finally, the paper is concluded in Section 5.

2. Related work

Many lightweight encryption algorithms have been proposed to secure data transmission of IoT devices with consideration to the resource limitation of such devices. We outlined some of the most recent related work as follows:

A lightweight encryption scheme based on Attribute-Based Encryption (ABE) is presented in [13], where they used ECC to deal with data security and privacy problem of transmitting data over IoT networks. A central authority was used to deal with key generation for attributes and users, but this can be a problem when it comes to a multi-authority application. However, their approach seems fair regarding the computation and communication cost for IoT devices. Moreover, the approach is weak in terms of inflexibility and scalability when revoking attributes, such issues must be dealt with when it comes to lightweight encryption algorithms of transmitted data of IoT devices [14].

A lightweight encryption system was developed in [15], which represents a lightweight symmetric-key block cipher of size 64-bits along with a 64-bits key size. Actually, this approach was based on a combination of a substitution-permutation network and Feistel by using few logical operations along with some substitution and permutation. An additional 5 rounds for key encryption have been added to enhance IoT devices energy consumption. The Feistel network is used to generate confusion and diffusion effects in the encrypted data. The generation and expansion of the key of size 64-bits were performed as in [16] using the XOR logical operation only.

Another lightweight encryption algorithm [17] is based on the Feistel block cipher with 64-bits block size and 128-bits key size that is performed in 31 rounds. The algorithm exploits logical operations such as XOR with the use of 4-bits S-Boxes, left substitution shift by 3 and right substitution shift by 7. This algorithm is based on an encryption key of size 128-bits, where this key is changed using the left substitution shift by 13 as introduced by PRESENT [18]. This algorithm may have good resistance against any attacks, but the computation and memory cost does not fit the limited computation and memory in IoT devices. Furthermore, a 64-bits block encryption algorithm that relies on generalized Feistel structure with a 4 round function is introduced in [19]. One of the proposed algorithms produced a flexible lightweight block cipher that fits the design of IoT devices, where it used 16-bits ultra-lightweight cipher instead of a Simpira V2 permutation as in [20].

A lightweight encryption algorithm named (LEAIoT) presented in [21], the authors exploited the prospects of the symmetric and asymmetric key algorithms. LEAIoT employs a private key that is known for both the sender and receiver. Then it implements a linear block cipher with one key to improve the security of ciphered text through the use of a shared key along with one private key. For the key generation, the authors exploited the symmetric encryption for the prospect for its short processing time that fits IoT devices constraints.

A symmetric encryption algorithm for transferring encrypted text files within the IoT network is presented in [22]; which introduces additional key confusion dynamically for each round of encryption. The algorithm has a 64-bits plaintext with a key size of 128-bits along with two rounds, where each round consists of 32 cycles. The key size of 128-bits was segmented into 4 sub-keys; each sub-key is 32-bits. The first and third sub-keys applied to odd-numbered round while the second and fourth sub-keys applied to even-numbered round to increase complexity for the attackers to predict the key.

An approach presented in [23], to encrypt medical images after converting image format into Quantum format; the approach used a DNA map to increase complexity for key generation. As compared to [24], an image encryption method used a chaos-based pseudorandom sequence generator using a neural network for encryption medical images. In addition to using DNA bit permutation and

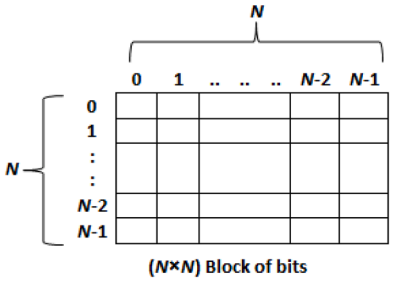

Fig. 1. Example of a data block of size ($N \times N$).

substitution operations for representing the pixel bits of the encrypted images. Another approach based on DNA and ECC for IoT devices [25], where the DNA sequence is used to encrypt data by employing substitution cipher. Their approach relied on replacing each character with a fixed DNA sequence and encrypted using ECC to be sent across the network.

Another fixed DNA sequence approach has been developed in [26] to generate primary keys. The approach used MixColumns and ShiftRows transformations within the AES algorithm that led to improving its resistance against attack. While in [27] an image encryption algorithm based on a finite-state machine and accompanying varying DNA sequences for a key generation was proposed to increase the security level for the encrypted image.

In summary, the proposed lightweight encryption system differs from the above-mentioned systems where our encryption system uses a variable size of keys and data block to make the system more flexible to be implemented on various IoT devices that have different memory sizes. Moreover, the key used in our proposed systems is extracted from the DNA sequence to ensure that the keys used in the encryption process are completely random.

3. Proposed system design

The widespread use of IoT devices with limited resources, that are transmitting confidential data, encouraged the scientists in the information security field to develop effective and lightweight cryptographic systems to provide protection for transmitted data.

Innovation and implementation of a lightweight cryptographic system face a host of challenges due to the implementation size, throughput or speed, and power consumption; in addition to the flexibility to implement the lightweight cryptographic system on different platforms for IoT devices. All of this should not reduce the level of data security.

To achieve these goals, a valuable lightweight cryptographic system is proposed in this paper that takes into account a set of main points:

1 Use only logical operations (such as XOR and Rotate operations) instead of using mathematical operations. This will allow speeding up the implementation of encryption, decryption, and keys generation operations. And makes it easy to run this system on different processors with the minimum capability of the Arithmetic and Logical Unit (ALU).
2 Use a simple structure to represent data, where the inputted data is represented as independent 2D matrices (blocks) of bits. This will help the processor of IoT devices to implement the multitasking concept to easily manage these blocks.
3 Use a variable block size to make the system more flexible to implement it on various IoT systems that have different memory sizes. Fig. 1 shows an example of a data block.

The block size is calculated using equation (1). Based on equation (1), the block size is ranged from a minimum of 64 bits to any block of size ($N^2 \times 64$) bits, where N represents the dimension of the block and is measured in bits.

$$Blocksize = (N \times 8) \times (N \times 8), \ where \ N \geq 1 \tag{1}$$

1 To achieve a high level of security in the key used, three parts of the DNA sequence are extracted based on predetermined indices by the user on the DNA sequence. These parts represent three primary keys used by the system. The use of the DNA sequence ensures that the primary keys entered are completely random. It is necessary to mention here that there are about 163 million DNA reference sequences available publicly. So, the probability of an attacker to successfully guess the correct chosen reference sequence is (1/(163*106)).
2 The key size used is proportional to the dimension of the data block used. This makes the key size suitable for available memory and the processor capability in the IoT system.

Details of the encryption and keys generation operations will be mentioned in the following subsections. Before that, the data structures used in the cryptography system are listed below:

- **Source Data (S):** a set of bytes that represent the data collected by the IoT device to be sent to the Cloud server.
- **Source Data Size (SSize):** refers to the size of the source data and it varies depending on the characteristics of the IoT device.

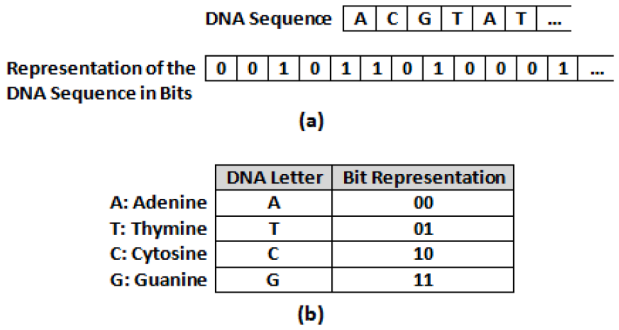

Fig. 2. (a) DNA sequence, (b) DNA letters and their representations in bits

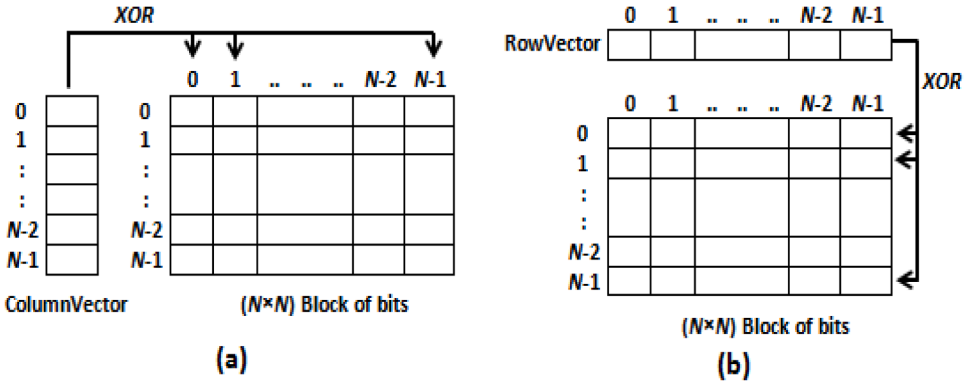

Fig. 3. XOR operation applied on bits on **SBlock** (a) Columns, (b) Rows

- **DNA sequence (DNA):** refers to the DNA sequence used by the cryptography system to extract the primary keys. The DNA sequence consists of a random sequence of four letters (A: Adenine, T: Thymine, C: Cytosine, and G: Guanine). Fig. 2.a shows a portion of the DNA sequence. In the cryptography system, each of the four DNA letters is represented as two bits as shown in the table in Fig. 2.b.
- **Source Data Block (SBlock):** represents a 2D matrix that contains some bytes of the source data S. The size of SBlock is measured using equation (1). Fig. 1 shows an example of a data block.
- **SBlock Dimension (SBlockDim):** represents the dimension of SBlock and it is measured in bits. The dimension value is an integer ≥ 8.
- **DNA Row Vector (RowVector):** is a vector of bits of length SBlockDim that is extracted from the DNA and it is used by the cryptography system, in rows processing operations and key generation, to encrypt SBlock.
- **DNA Column Vector (ColumnVector):** is a vector of bits of length SBlockDim that is extracted from the DNA and it is used by the cryptography system, in columns processing operations and key generation, to encrypt SBlock.
- **DNA Mutator Vector (MutatorVector):** is a vector of bits of length SBlockDim that is extracted from the DNA and it is used by the cryptography system to generate keys.
- **Encrypted Data (E):** a set of bytes that represent the source data **S** after the encryption operation is complete. The encrypted data **E** will be sent from the IoT device to the connected cloud server.

The encryption phase of the proposed lightweight cryptographic system consists of four stages: Preprocessing stage, Substitution stage, Transposition stage, and Key generation phase.

3.1. *Preprocessing stage*

In this stage, a set of operations is performed as follows:

- Split the source data **S** into a set of blocks **SBlock**, the number of blocks, which is created from **S**, is calculated using equation (2).

$$NoOfBlocks = SSize/Blocksize \tag{2}$$

- Represent the DNA sequence as a vector of bits using the mapping table in Fig. 2.b.

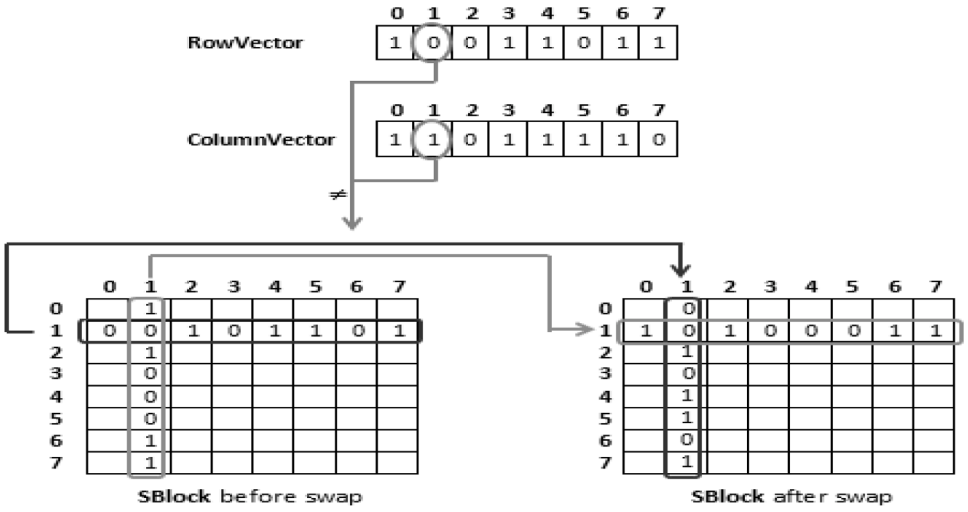

Fig. 4. An example of the transposition operation.

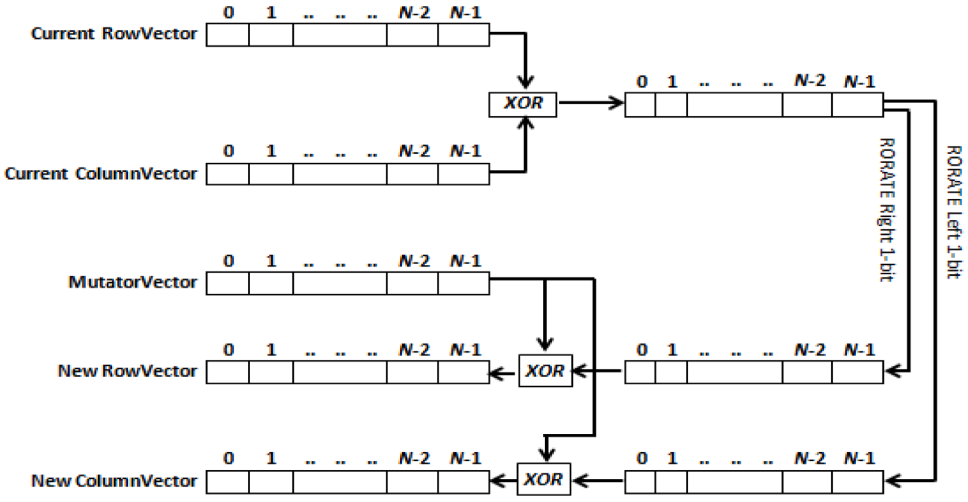

Fig. 5. Steps to generate new keys (vectors) from the current **RowVector** and **ColumnVector** vectors.

- Extract three primary vectors of bits from the DNA sequence (**RowVector, ColumnVector,** and **MutatorVector)** to use them as keys in the encryption phase.

3.2. Substitution operation stage

The substitution operation is implemented on the rows and columns of each **SBlock**. In this cryptographic system, the logical XORing operation is applied between the bits in **RowVector** with the corresponding bits in each row of **SBlock**, and between the bits in **ColumnVector** with the corresponding bits in each column of **SBlock**. This operation will make changes in **SBlock** bits. It is necessary to mention here that the **RowVector** and **ColumnVector** would be different for each **SBlock** because these vectors will be changed during the key generation stage. Fig. 3 shows the XOR operation that is applied in rows and columns of **SBlock**.

3.3. Transposition operation stage

The transposition operation is performed to change the bit locations in **SBlock**. This operation involves swapping N bits of the row and column at the same index. The swapping process is performed (or not) depending on the similarity or difference of the bit value in the same index in the **RowVector** and **ColumnVector**. This is done based on the following two conditions:

- If (**RowVector [i]** $\neq$ **ColumnVector [i]**) then exchange ***N*** bits in row **i** with ***N*** bits in column **i** of the **SBlock**.

Table 1
Size of block and key(s) used in lightweight cryptographic systems [9],[8],[30].

Cryptographic System	Block Size (bit)	Key(s) Size (bit)
Proposed algorithm	64	24
Blowfish	64	32
SIMON	64	96
RC5-20	64	128
Chaskey	128	128
AES	128	128
Twofish	128	128
Robin	128	128

- If (**RowVector [i]** = **ColumnVector [i]**) then no exchange is done.

Fig. 4 shows an example of the transposition operation that is performed based on the above conditions.

3.3. Key generation stage

In the cryptographic system each **SBlock** of data is encrypted using different **RowVector** and **ColumnVector**. This is done by generating new vectors for use in the next **SBlock**. In this stage, only two logical operations (XOR and ROTATE) are used to generate new keys. The use of logical operations will speed up the encryption operation. Fig. 5 demonstrates steps that are followed to generate new keys (vectors) from the current **RowVector** and **ColumnVector** vectors.

4. Performance evaluation & discussion

In order to evaluate the proposed cryptographic system and to ensure that it succeeds in achieving the predetermined goals. Several experiments were conducted that focused on two criteria:

- The amount of memory used to implement the proposed cryptographic system.
- The time required to complete the encryption operation.

The security level achieved by using the proposed cryptographic system is tested as recommended in [28] using the following criteria:

- The confusion and diffusion in the encrypted data using Peak Signal to Noise Ratio (PSNR) and histogram metric.
- The size of the key used in the system.
- The randomness in the key used in the system.
- Change the key used by the system.

The evaluation of the proposed cryptographic system has been conducted by comparing the recorded results in the experiments with well-known cryptographic systems [13,29].

4.1. Memory size

The flexibility of using different block sizes, in the proposed cryptographic system, makes the system suitable to implement in the IoT devices having different platforms. As mentioned before, the block size is calculated using equation (1) and it is ranged from a minimum of 64 bits to any block of size ($N^2 \times 64$) bits, where N represents the dimension of the block and is measured in bits. Also, the total size of the three keys used depends on the block size used and which is calculated using equation (3).

$$\text{Total Size of Three Keys} = \sqrt{\text{Blocksize}} \times 3 \tag{3}$$

In Table 1, we can note that the proposed system succeeded in reducing the memory used by minimizing the size of the data block and the key size used when compared to other cryptographic systems.

4.2. Encryption time

Many IoT devices continuously transfer data to the cloud server, which requires providing the IoT devices with a fast cryptographic system to encrypt the transferred data in the shortest possible time. Most lightweight cryptographic systems try to do this through:

- Reduce the number of rounds in the encryption phase.
- Implement the cryptosystem directly on hardware rather than a software implementation.

Table 2
The encryption time of the proposed and AES cryptographic systems.

Data	Data Size (byte)	Encryption time (msec)	
		Proposed	AES
Data 1	49152	139	750
Data 2	709200	531	1910
Data 3	889200	911	1323
Data 4	950878	1030	1890

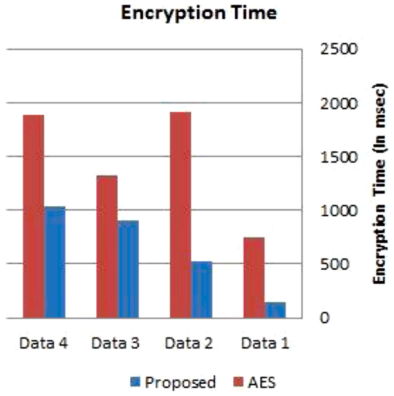

Fig. 6. The graphical chart of encryption time values in Table 2.

Table 3
The PSNR values of the encrypted data using both proposed and AES systems.

Data	PSNR (db)	
	Proposed	AES
Data 1	8.00	8.05
Data 2	7.13	7.15
Data 3	8.11	8.09
Data 4	7.56	7.63

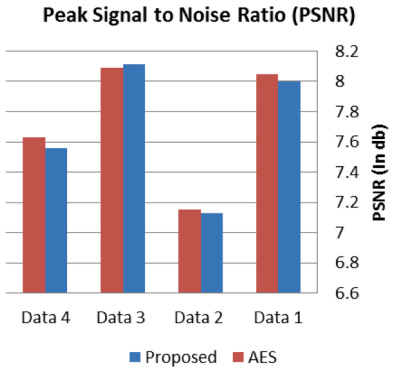

Fig. 7. The graphical chart of PSNR values in Table 3.

- Reduce the use of arithmetic operations in the encryption phase and replace these operations with logical operations such as XORing operation.
- It is necessary to mention here that all these strategies should not affect the level of security that must be achieved.

In the proposed cryptographic system, only logical operations are used in the encryption phase (XORing and ROTATing as described before), and all these operations are performed in one round only. Table 2 shows the encryption time in the proposed

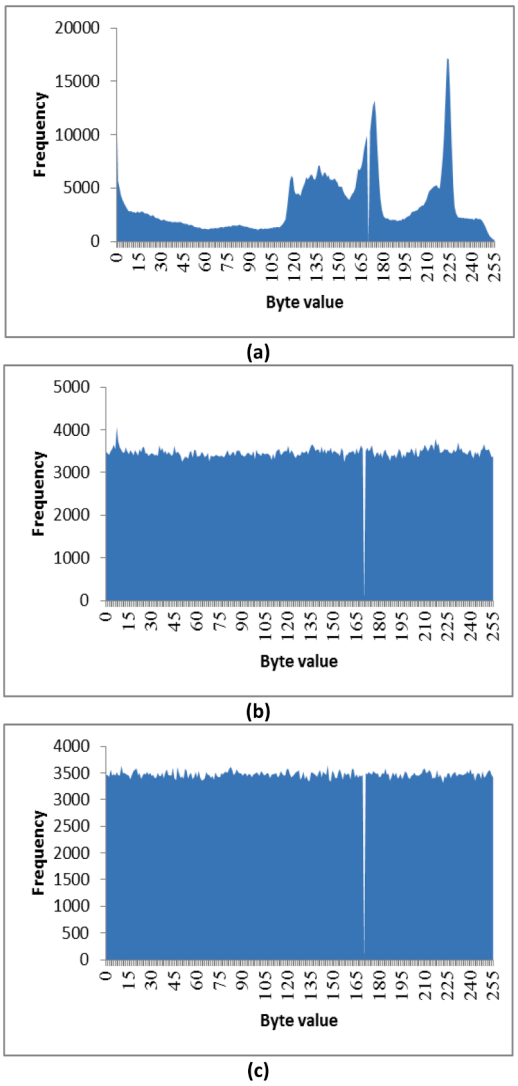

Fig. 8. Histogram of (a) source data (b) encrypted data using the proposed system (c) encrypted data using AES system.

cryptographic system and AES system which represents the reference system for the lightweight cryptographic systems. The times recorded as illustrated in Fig. 6 shows that the proposed cryptographic system has significantly reduced the encryption time very much compared to the AES system.

4.3. Security level

One of the main objectives of any encryption system is to cause a high percentage of confusion and diffusion in the source data in order to produce encrypted data. The effects of confusion and diffusion can be measured numerically using Peak Signal to Noise Ratio (PSNR) measurement by using equation (4) and (5) respectively. Table 3 shows the PSNR values that are recorded during experiments, where the PSNR value for the proposed lightweight cryptographic system proved to be comparable with AES as illustrated in Fig. 7 for a different data size.

$$NMAE = \frac{\sum_{k=0}^{SSize-1} |S(k) - E(k)|}{SSize} \times 100 \tag{4}$$

$$PSNR_{db} = 10.\log_{10}\left(\frac{Max_I^2}{NMAE}\right) \tag{5}$$

Where: Max_I is the maximum possible byte value of the data $\boldsymbol{S}$. And db refers to a decibel.
Also, these effects can be checked statistically by checking the histogram of the byte values, where the flatness in the histogram of

Table 4
Entropy values of the source and the encrypted data.

Data	Data Size (byte)	Entropy		
		Source	Proposed	AES
Data 1	49152	4.172361	7.994617	7.994232
Data 2	709200	4.163458	7.997011	7.996892
Data 3	889200	4.245891	7.998321	7.998300
Data 4	950878	4.262732	7.999982	7.999970

Table 5
The calculated values of the avalanche effect for "Data 3".

Number of bits changed in the encryption key	Avalanche Test Value (%)
1	50.250
5	50.489
25	50.615

the encrypted data indicates to achieve good confusion and diffusion effects that occurred in byte values. Fig. 8 shows an example of the histogram of the source and encrypted data generated using the proposed and AES systems.

An entropy measure is usually used to evaluate the efficiency of the encryption method where it is difficult for a digital data to predict the content if its information entropy is high. The entropy is calculated using equation (6).

$$\text{Entropy} = -\sum_{i=1}^{n} P_i.\log_2(P_i) \tag{6}$$

Where n is the number of different data values and P_i is the probability of occurring the data value. Table 4 lists the entropy values of the source and the encrypted data. It can be noted that entropy values recorded by the proposed system for different size of data show a promising results compared to AES.

To check the sensitivity of the encryption method to any minor changes in the parameters, an avalanche effect test is used. A high-quality encryption method is the one that when there is a slight change in input (either in the key or the source data), it must cause major changes in the encrypted data. The calculated values of the avalanche effect using equation 7, for "Data 3", are listed in Table 5 for a different number of bits altered. The change in one-bit, five-bits, and 25-bits respectively of the key for the proposed system resulted in a great change of the encrypted data which is more than 50%.

$$\text{Aval. Effect} = \frac{\textit{Number of changed bits in key used}}{\textit{Total number of bits in encrypted data}} \tag{7}$$

In light of the obtained results of the proposed cryptographic system, it was proved that the system is adaptable to various computational resources of IoT devices. Also, the proposed system succeeded to achieve an adequate level of protection for the data and it can be used effectively by any IoT applications.

5. Conclusion

In this paper, we have introduced a flexible lightweight encryption system for IoT devices. The system has used a changeable data size and strong logical operations to encrypt and decrypt data to meet the limitations in IoT devices in terms of memory and processing capabilities. The proposed encryption system as proved a better performance regarding encryption time compared to AES. The proposed encryption system also exploits DNA sequence for keys generation to achieve high randomness in the keys in order to make the keys hard to break by the attackers. In addition, good confusion and diffusion effects were achieved by the proposed encryption system. Moreover, the avalanche test value for the proposed encryption system was over 50% which means it's protected against statistical analysis attacks. Therefore, the proposed system represents a promising approach to be used in most IoT devices that have different capabilities.

CRediT authorship contribution statement

Mohammed Abbas Fadhil Al-Husainy: Conceptualization, Methodology, Software, Investigation. **Bassam Al-Shargabi:** Writing – original draft, Writing – review & editing, Conceptualization, Investigation. **Shadi Aljawarneh:** Formal analysis, Writing – review & editing, Investigation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to

influence the work reported in this paper.

Acknowledgment

The authors are grateful to the Middle East University, Amman, Jordan for the full financial support granted to this research project.

References

[1] Al-Shargabi B, Sabri O. Internet of Things: An exploration study of opportunities and challenges. In: 2017 International Conference on Engineering & MIS (ICEMIS). IEEE; 2017. p. 1–4.
[2] Beg A, Al-Kharobi T, Al-Nasser A. Performance Evaluation and Review of Lightweight Cryptography in an Internet-of-Things Environment. In: 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS). IEEE; 2019. p. 1–6.
[3] Abbas Fadhil Al-Husainy M, Al-Shargabi B. Secure and Lightweight Encryption Model for IoT Surveillance Camera. Int J Adv Trends Comput Sci Eng 2020;9:1840–7.
[4] Sepranos D, Wolf M. Challenges and Opportunities in VLSI IoT Devices and Systems. IEEE Des Test 2019;36:24–30.
[5] Aljawarneh S, Yassein MB, Talafha WA. A resource-efficient encryption algorithm for multimedia big data. Multimedia Tools Appl 2017;76:22703–24. https://doi.org/10.1007/s11042-016-4333-y.
[6] Aljawarneh SA, Moftah RA, Maatuk AM. Investigations of automatic methods for detecting the polymorphic worms signatures. Future Gen Comput Syst 2016;60:67–77.
[7] Aljawarneh S, Yassein MB. A multithreaded programming approach for multimedia big data: encryption system. Multimedia Tools Appl 2018;77:10997–1016.
[8] Buchanan WJ, Li S, Asif R. Lightweight cryptography methods. J Cyber Security Technol 2017;1:187–201.
[9] Al-Husainy MAF, HAA Al-Sewadi. Implementing Binary Search Tree Concept for Image Cryptography. Int J Adv Sci Technol 2019;130:21–32. https://doi.org/10.33832/ijast.2019.130.03.
[10] Roy S, Rawat U, Sareen HA, Nayak SK. IECA: an efficient IoT friendly image encryption technique using programmable cellular automata. J Ambient Intell Human Comput 2020:1–20.
[11] Abualese H, Al-Rousan T, Al-Shargabi B. A New Trust Framework for E-Government in Cloud of Things. Int J Electron Telecommun 2019;65:397–405. https://doi.org/10.24425/ijet.2019.129791.
[12] Biswas A, Majumdar A, Nath S, Dutta A, Baishnab KL. LRBC: a lightweight block cipher design for resource constrained IoT devices. J Ambient Intell Human Comput 2020:1–15.
[13] Yang W, Wang R, Guan Z, Wu L, Du X, Guizani M. A Lightweight Attribute Based Encryption Scheme with Constant Size Ciphertext for Internet of Things. In: ICC 2020-2020 IEEE International Conference on Communications (ICC). IEEE; 2020. p. 1–6.
[14] Yao X, Chen Z, Tian Y. A lightweight attribute-based encryption scheme for the Internet of Things. Future Gen Comput Syst 2015;49:104–12.
[15] Usman M, Ahmed I, Aslam MI, Khan S, Shah UA. SIT: a lightweight encryption algorithm for secure internet of things. ArXiv Preprint ArXiv:170408688 2017.
[16] Barreto P, Rijmen V. The Khazad legacy-level block cipher. Primitive Submitted to NESSIE 2000;97:106.
[17] Patil J, Bansod G, LiCi Kant KS. In: A new ultra-lightweight block cipher. 2017 International Conference on Emerging Trends & Innovation in ICT (ICEI). IEEE; 2017. p. 40–5.
[18] Bogdanov A, Knudsen LR, Leander G, Paar C, Poschmann A, Robshaw MJB, et al. PRESENT: An ultra-lightweight block cipher. In: International workshop on cryptographic hardware and embedded systems. Springer; 2007. p. 450–66.
[19] Yeoh W-Z, Sen Teh J, Sazali MISBM. μ2: A Lightweight Block Cipher. Computational Science and Technology: 6th ICCST 2019, 603. Malaysia: Kota Kinabalu; 2019. p. 281. 29-30 August 2019.
[20] Gueron S, Mouha N. Simpira v2: A family of efficient permutations using the AES round function. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer; 2016. p. 95–125.
[21] Habib MA, Ahmad M, Jabbar S, Ahmed SH, Rodrigues JJPC. Speeding up the internet of things: Leaiot: A lightweight encryption algorithm toward low-latency communication for the internet of things. IEEE Consumer Electron Mag 2018;7:31–7.
[22] Rajesh S, Paul V, Menon VG, Khosravi MR. A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices. Symmetry 2019;11:293.
[23] Devi RS, Balaguru RJB, Amirtharajan R, Praveenkumar P. A Novel Quantum Encryption and Authentication Framework Integrated with IoT. Security, Privacy and Trust in the IoT Environment. Springer; 2019. p. 123–50.
[24] Rarhi K, Saha S. Image Encryption in IoT Devices Using DNA and Hyperchaotic Neural Network. Design Frameworks for Wireless Networks. Springer; 2020. p. 347–75.
[25] Barman P, Saha B. DNA encoded elliptic curve cryptography system for IoT security. Int J Comput Intell IoT 2019;2.
[26] Al-Wattar AH, Mahmod R, Zukarnain ZA, Udzir NI. A New DNA-Based Approach of Generating Key-dependent ShiftRows Transformation. ArXiv Preprint ArXiv:150203544 2015.
[27] Khan S, Han L, Lu H, Butt KK, Bachira G, Khan N-U. A new hybrid image encryption algorithm based on 2D-CA, FSM-DNA rule generator, and FSBI. IEEE Access 2019;7:81333–50.
[28] Turan MS, McKay KA, Çalık Ç, Chang D, Bassham L. Status report on the first round of the NIST lightweight cryptography standardization process. Gaithersburg, MD: National Institute of Standards and Technology; 2019. NIST Interagency/Internal Rep(NISTIR).
[29] Dhanda SS, Singh B, Jindal P. Lightweight Cryptography: A Solution to Secure IoT. Wirel Pers Commun 2020:1–34.
[30] Fadhil Al-Husainy MA, Al-Sewadi HA, Masadeh SR. Lightweight cryptosystem for image encryption using auto-generated key. J Eng Appl Sci 2018;13:7418–25. https://doi.org/10.3923/jeasci.2018.7418.7425.

Mohammed Abbas Fadhil Al-Husainy, received the M.Sc. and Ph.D. degrees in 1996 and 2002, respectively. His research interests are in the area of multi-media data processing, scheduling algorithms, information, system security, cryptography, and steganography algorithms.

Bassam Al-Shargabi, received his Ph.D. and M.Sc in Computer Information Systems from the Arab Academy for Banking & Financial Sciences (Jordan) in 2009 and 2004, respectively. Currently, he is an Associate professor at the Faculty of Information Technology, Middle East University, Amman-Jordan. His current research interests are in Natural language processing, information retrieval, Data Security, and IoT.

Shadi Aljawarneh, is a full professor within the Department of Software Engineering, at the Jordan University of Science and Technology. His research is centered in software engineering, web and network security, e-learning, AI, machine learning, and cloud computing.