Chapter

Toward Lightweight Cryptography: A Survey

Mohammed Abujoodeh, Liana Tamimi and Radwan Tahboub

Abstract

The main problem in Internet of Things (IoT) security is how to find lightweight cryptosystems that are suitable for devices with limited capabilities. In this paper, a comprehensive literature survey that discusses the most prominent encryption algorithms used in device security in general and IoT devices in specific has been conducted. Many studies related to this field have been discussed to identify the most technical requirements of lightweight encryption systems to be compatible with variances in IoT devices. Also, we explored the results of security and performance of the AES algorithm in an attempt to study the algorithm performance for keeping an acceptable security level which makes it more adaptable to IoT devices as a lightweight encryption system.

Keywords: cyber, information security, IOT security, networks, cryptography, AES, lightweight cryptography

1. Introduction

An information system is a set of interconnected components that collect, process, store, and transfer information. These components include the physical and software components and the communication networks [1].

Networks enable communications between many devices by connecting them and enabling the most reliable possible connection. Moreover, networks are subject to many attacks due to users and their different directions. Here, the challenge lies in maintaining the security of these networks with their resources and data while maintaining high performance [1–3].

In its simplest sense, Internet of Things (IoT) is a system of various intelligent devices known in our daily lives. These things link and communicate between them and ensure data transfer between them independently via the network without human interaction, a self-control system [4–6]. Smart Cities played an essential role in highlighting IoT. Smart Cities express the concept that depends on the city's technology, as these cities are linked to each other electronically. Information is collected continuously from sensors, monitoring, and computers covering the whole city [5, 6]. "Thing" term can be a sensor network, as safe houses, or in general, any device that can take an IP address and can interact through a network [5].

Security plays an essential role in judging IoT applications strengths. Users wish to have secure IoT software hat is secure in all respects. IoT application's security includes a secure transfer of data, protection from eavesdropping, and unauthorized

access. The system security has become one of the essential critical requirements of the system's core functions [2, 3]. Furthermore, the security aims to achieve what is known as the Confidentiality, Integrity, and Availability (CIA) triad. Finally, one of the most critical security goals is to control access through the Authentication, Authorization, and Accounting (AAA) framework [3, 7].

IoT causes a massive increase in the volume of data. Securing such enormous amount of data requires special efforts. Several technologies may serve this purpose. But the devices used in the combination of smart cities and the IoT vary among themselves in capabilities. Moreover, most of these devices have limited specifications and restrictions [5, 6]. Hence the need to find new technologies that work on these limited capabilities and achieve an acceptable degree of security. Furthermore, since the capabilities are limited, these technologies should be lightweight and rely on simple operations without consuming energy, storage, and processing capacity.

The rest of the chapter organized as follows. We provide clarifications for some concepts related to this field considering cryptography and Lightweight Cryptography (LWC) areas in Section 2. Section 3 provides some researches related to LWC, and Advanced Standard Algorithm (AES). In Section 4, recommendations and findings are discussed. Finally, we conclude the paper and present a vision for future work in Section 5.

2. Background

This section introduces the concepts of IoT, Cryptography, and LWC in Subsections 2.1, 2.2, and 2.3 respectively.

2.1 Internet of thing

IoT today is a hot topic in research. The importance of IoT comes because of keeping pace with the variables of life that call us to exploit everything new in technology, such as computers, cars, TV, refrigerators, and washing machines [5, 6]. **Figure 1** shows IoT Reference Architecture. The figure shows that the IoT system consists of data collector's devices as a sensor used to get the data and data analyzer device like a mobile phone used for data processing to make a decision. These two subsystems communicate and transfer data via a network [5, 6, 8].

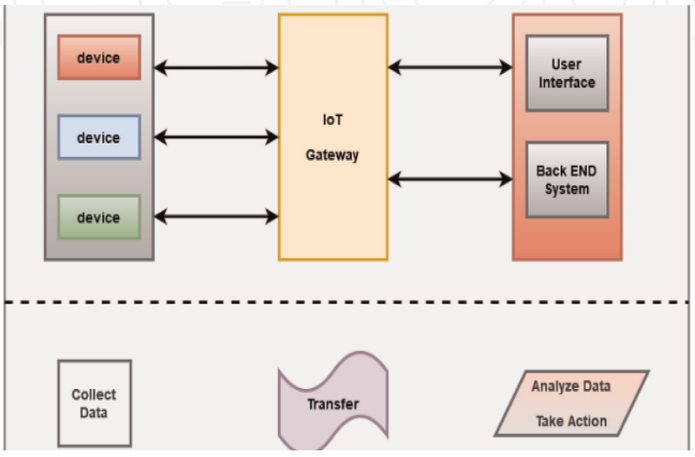

Figure 1.
IoT reference architecture [8].

IoT has dramatically helped to increase the efficiency of work and operations. It relies on a system of self-interaction that means reducing the waiting time for response. As a result, performance gains, and therefore the number of completed processes increases, giving users access to the best possible user services, enhancing the work's actual value [5, 8]. In general, IoT provides a wide range of benefits at the enterprise and individual levels.

Figure 2 presents the concept of IoT. There are many valuable and significant applications for IoT, such as Safe Houses, Health Care, and Farming systems.

Despite the significant benefits of IoT, the IoT suffers from a lack of standardization and is vulnerable to cyber-attacks, data theft, data fraud, botnet attacks, and physical compromises. The reason for this is that the IoT differs from traditional networks. There are two types of IoT devices: those rich in resources, like computers, and those with limited resources, like sensors. The real challenges are in the second type, which has low memory and computing power, short battery life, and Low bandwidth to connect [6, 8]. So, we should be careful about security and privacy [8]. Hence, the challenge is how to design an IoT system efficiently and securely.

2.2 Cryptography

Cryptography is a way to protect data and communications by ensuring that those not authorized to access sent data cannot read and process it [3, 9]. The goals of the encryption process revolve around guaranteeing each of the following [3, 9, 10]:

- *Confidentiality:* Using Encryption to protect data from unauthorized reading.
- *Data Integrity:* Ensures that the message remains the same as sent without changing it by using a unique message digest.
- *Non-Repudiation:* Ensures that the recipient does not deny the message's arrival by proving that the sender sent the message.
- *Authentication*: Proof of an entity identity, which confirms the user's right to access the system or data.
- *Access Control*: Ensures that access to the system or data is limited by preventing unauthorized access and checking their privileges.

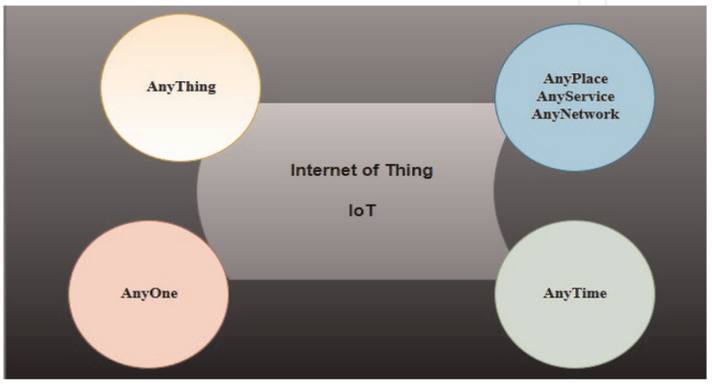

Figure 2.
Concept of IoT [4].

- Still, there are some essential terms related to security worlds, they include:

- *CIA Triad:* In addition to confidentiality and integrity, we still have the concept of availability, which ensures that authorized users can access what they want at any time. Therefore, the CIA triad tries to achieve the three goals emphasized [9].

- *AAA Framework* is responsible for enforcing policies and controlling access over resources. In addition to the authentication previously mentioned, it ensures that the security methods used in the network guarantee [7, 11]:

- *Authorization*: Not much different from access control. It works on the resources the user is allowed to access and use.

- *Accounting*: Directly, it can be defined as a complete monitoring process and writing down all the operations that the user performs to be used further in the accounting, analysis, and planning process.

Figure 3 summarizes CIA triad and AAA framework.

2.2.1 Cryptography algorithms

In cryptography science, encryption transforms original messages (Plain Text) to non-readable data (Cipher Text) using an encryption algorithm. This Cipher Text cannot give anyone any information about the Plain Text except those with the encryption key [9, 12–17]. Therefore, we can perform a simple encryption example by replacing every character in the plain text with its next character in aliphatic order.

P = "Thesis".

Alg.: substitution $P_i = P_{i+1}$.

C = "uiftjt".

There are two main types of encryptions: Asymmetric cipher, and Symmetric cipher, as shown in **Figure 4** [9, 12–17].

2.3 Asymmetric cipher

Asymmetric cipher is conjointly referred to as public-key cryptography. Associate cryptography technique uses a mix of public key and private key. The sender has the receiver's public key, whereas the private key is not known. The receiver ought to produce his try of the general public and private key, publish his public key while not

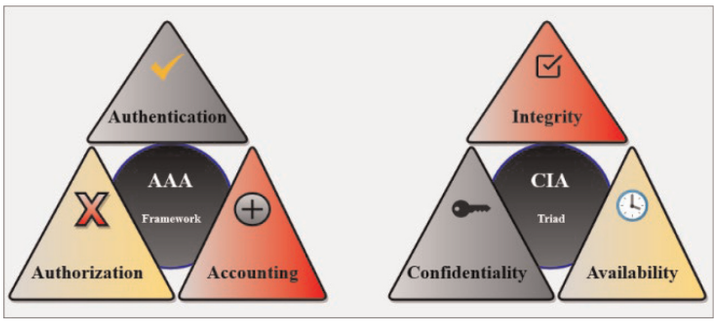

Figure 3.
CIA triad and AAA framework [10, 11].

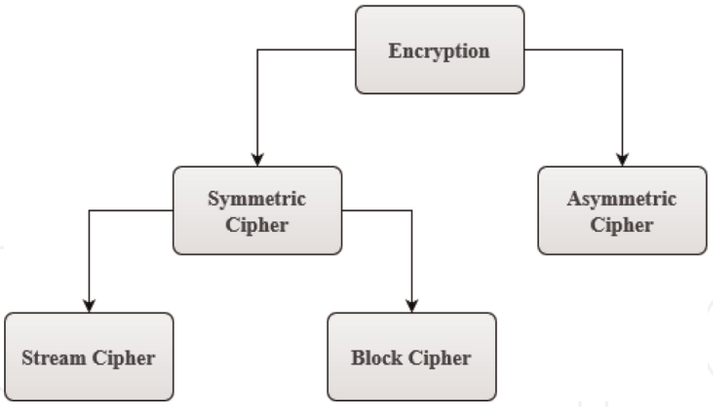

Figure 4.
Encryption models [9].

considering its security. The private key should be a procedure not possible to seek out through the general public key.

Uneven cryptography is employed in authentication and digital signatures. A signed message with the sender's private key proves the sender's identity, and anyone who has that sender's public key can verify it. Thus, the receiver may ensure that the message has not been changed or replaced by the other one that confirms the sender's identity [9, 15, 16].

Rivest–Shamir–Adleman (RSA) algorithm one of the most popular and widely used asymmetric encryption algorithms. It was developed in 1977 by Ron Rivest, Adi Shamir, Leonard Adleman and took its name from them. Besides Encryption, Digital signatures and key exchange are possible using RSA [17, 18].

RSA gained its strength by relying on parsing large integers in the formation of keys. First, two prime numbers are manipulated to create the user's public and private keys. Then, the message is encrypted using the recipient's public key and decrypted exclusively with the recipient's private key. **Figure 5** shows the RSA Process [17].

Although RSA is the most popular and secure asymmetric encryption algorithm in terms of key difficulty, it takes a long time to encrypt and decrypt. Besides, a security flaw appears that encrypting the same message again produces the same encrypted message [18].

ElGamal is an asymmetric cipher based on Diffie–Hellman key exchange. This algorithm gains its strength through the difficulty of finding discrete logarithms. For example, even though we know G^x and G^y, it is challenging to find G^{xy}. This algorithm consists of key generation, encryption, and decryption processes. **Figure 6** shows each of them [19–21].

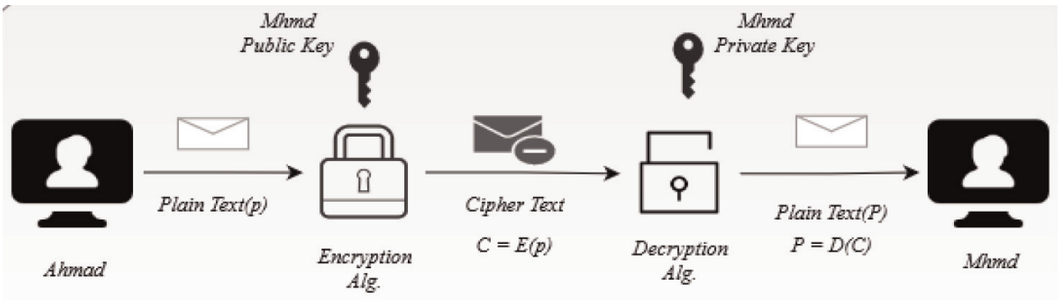

Figure 5.
RSA process [18].

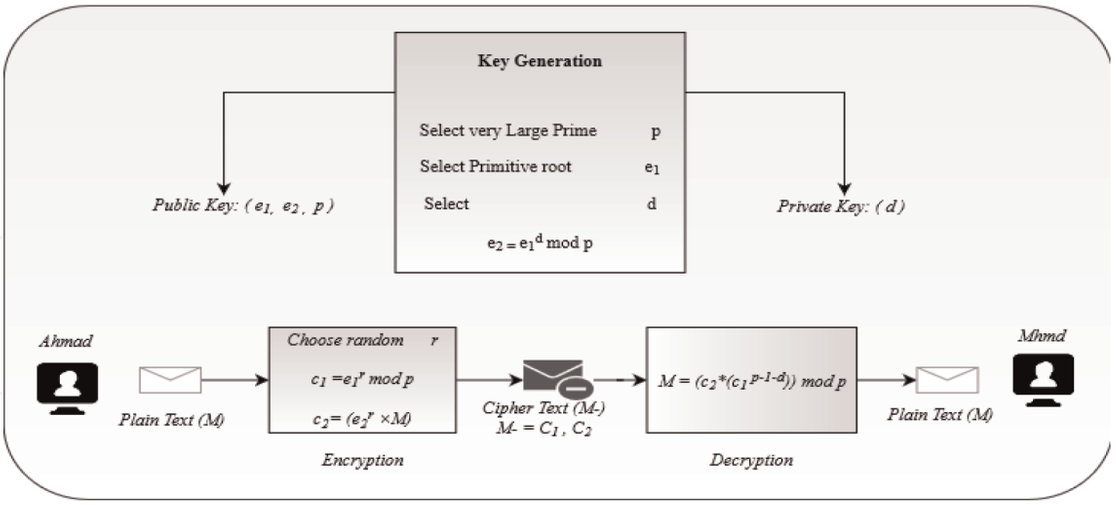

Figure 6.
ElGamal Alg. [20].

Elliptic Curve Cryptography (ECC) it uses the mathematics on elliptic curves. ECC is widely used due to its high security and small size. The difficulty in cracking the elliptic curves that underpin key strength has made ECC more secured and considered as the next generation of RSA [21, 22].

The main difference between ECC and RSA is the strength of the key. A 160-bit key in ECC is equivalent in power to a 1024-bit key in RSA. Considering that there is no linear relationship, doubling the size of the RSA key does not mean that we need to double the size of the RSA key. ECC is characterized by the speed of obtaining the keys and less memory to store them. On the other hand, a challenge for ECC is that it cannot be implemented as efficiently as RSA [22]. **Figure 7** presents the ECC.

Digital Signature Algorithm (DSA) is an algorithm that uses discrete logarithms and standard bases to introduce and validate the notion of a digital signature. Compared to RSA, DSA provides faster key generation.

As a result, it is slower in the encryption process, but it offers better results in the decryption process. DSA is mainly used to verify the sender's identity of a message since it bears his signature, which cannot be duplicated [23]. **Figure 8** presents the DSA mechanism.

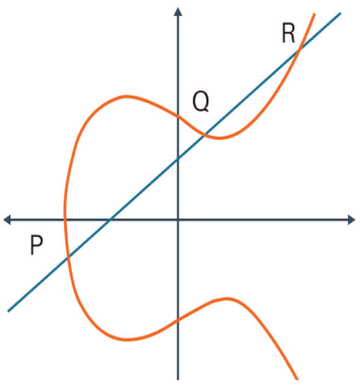

Figure 7.
ECC basics [22].

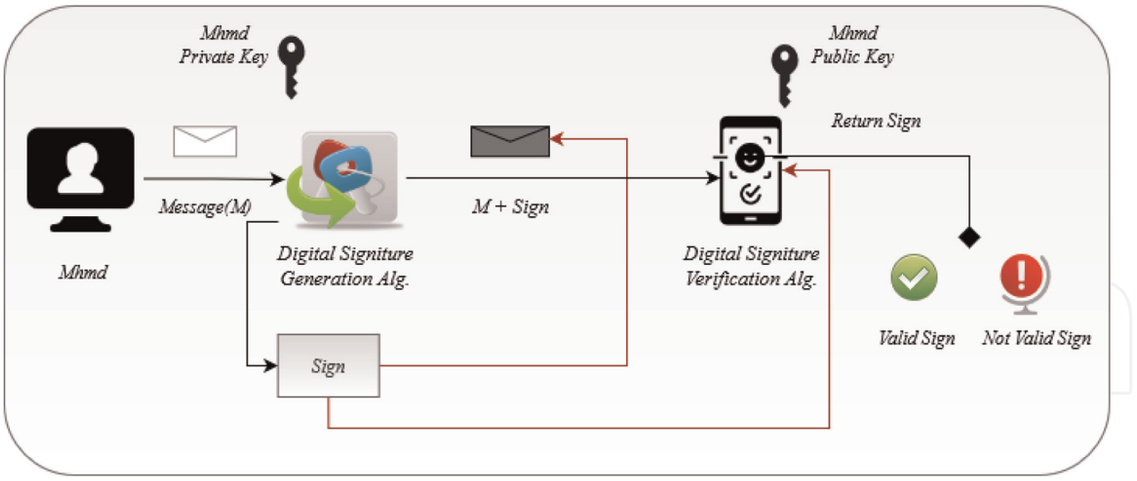

Figure 8.
DSA process [23].

2.4 Asymmetric cipher summary

This section discussed various Asymmetric Cipher algorithms such as RSA, ElGamal, DSA, and ECC. **Table 1** highlight the most comparison points between them. This type of algorithm offers high strength in terms of security, it requires a large amount of processing, which means low performance and draining resources. Therefore, based on the preceding, these algorithms are not compatible with the discrepancy in the capabilities of IoT devices and therefore cannot be used in building security systems in term of encryption. Hence, we find that symmetric encryption is more suitable for such systems. However, this does not detract from its value, as it cannot be dispensed with in verification, key exchange, and signature operations.

Cipher	Key size *(bits)*	Strength	Weakness
RSA	1024 2048 3072 4096	• Low computational time. • Fast.	• Use same module for multi users. • For small messages. • Not Scalable.
ElGamal	1024	• Fast. • Very efficient in hardware imp. • Solve discrete logarithm. • Good Scalability. • Low Power Consumption.	• Require Random Number Generator. • Ciphertext is very Large. • Slow in Signing.
DSA	512–1024 ***(multiple of 64)***	• Authentication. • Integrity. • Non-repudiation.	• Entropy. • Secrecy. • Uniqueness of random signature.
ECC	160 224 256	• Small Key size. • Low storage. • Low transmission time, and power consumption. • Very Fast.	• Ciphertext is large. • High Complexity

Table 1.
Asymmetric ciphers comparison.

2.5 Symmetric cipher

Each sender and receiver share the same secret key in this kind of Encryption. Hence, it uses within the encryption and decryption processes. However, symmetric Encryption has better speed but a lower security level than asymmetric [9, 12, 16, 24]. **Figure 9** shows the general structure of this encryption model. Symmetric ciphers can be used as a block cipher or stream cipher. We will discuss both types in detail in this section.

2.5.1 Stream cipher

In this type of encryption, the data are encrypted bit by bit. Because every encrypted bit is independent of other bits, diffusion and confusion properties are not achieved [9].

This encryption type mainly uses as simple as possible operators in this type of cipher. In most cases, it uses the XOR operation between the plaintext bits and the corresponding key bits. As a result, stream cipher throughput (speed of Encryption) is much *higher* than the block cipher but is considered less secure than Block Cipher [9, 12–16, 24].

Rivest Cipher 4 (RC4) is a stream cipher algorithm proposed by Ron Rivest in 1987. It later became a widely used algorithm from being a personal algorithm due to its speed and simplicity. RC4 has been frequently used to encrypt network traffic [25]. This algorithm uses byte-oriented operations with a variable key size. Simply put, RC4 relies on an XOR operation between each piece of plaintext with a small portion of the key to produce the ciphertext. And the decoding process is only a reflection of this process. However, with the development of computers, it became possible to break this algorithm easily. However, RC4 can be considered secure if the initial bytes of the key are ignored [25–27].

Salsa20 is a synchronous stream cipher suggested by Bernstein. The number 20 indicates the number of rounds, but this can be reduced to 12 or 8 as needed. *Salsa20* relies on simple operations such as rotation, addition, and XOR, making it a high-speed algorithm, which makes it secure against timing attacks [27, 28].

Sosemanuk is a synchronous stream cipher with variable key length. It has good properties of confusion and diffusion for a low cost. Furthermore, the Mux operation

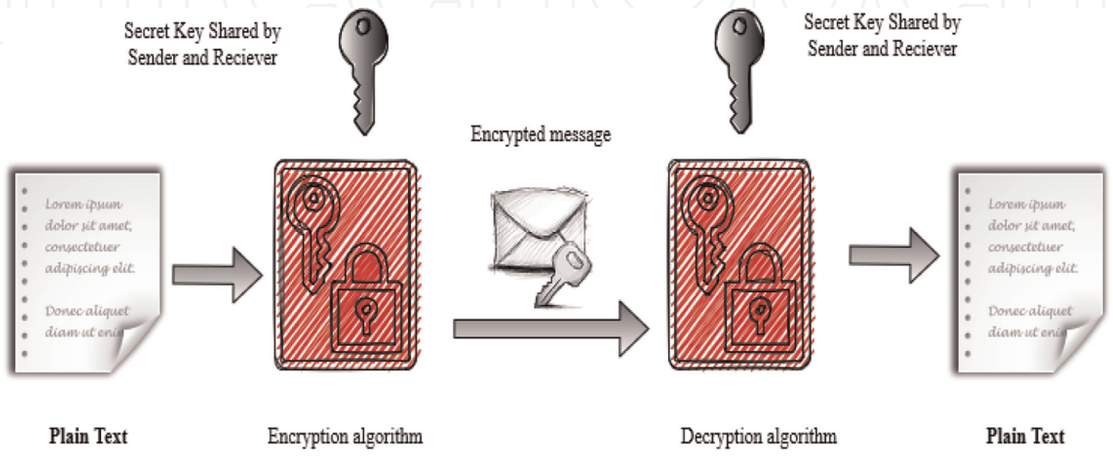

Figure 9.
Simple symmetric model [9].

is secure against algebraic and fast correlation attacks. Finally, Sosemanuk has good performance due to the internal static data [29].

Table 2 provides a brief comparison of these algorithms, following our discussion and our review of their definitions and specifications.

From this comparison, we note that the RC4 algorithm is optimal for use, as it is more robust and available in more than one version to suit the system in which it will be used. However, in light of the fact that stream ciphers offer high speed and low security and the requirement for keys to be the same size as plaintext, none of these algorithms are suitable for use as a foundation for building an IoT system.

2.5.2 Block cipher

In Block Cipher, the plaintext is divided into blocks based on encryption algorithm structure [12]. This type of Encryption has an execution time slower than the stream cipher. So, the encryption throughput of stream cipher is much higher than the block cipher [9, 23]. In contrast, a block cipher provides better security than the stream cipher against some well-known attacks. Moreover, the essential properties of the secure ciphertext, which are the confusion and the diffusion properties, are included inside block ciphering algorithms. Based on these facts, we can nominate one block cipher algorithm to build our algorithm for the IoT after reviewing it and choosing the most appropriate based on its specification and results.

Data Encryption Standard (DES) is a symmetric encryption algorithm that uses a seemingly 64-bit key, of which 56 bits are used as the practical key over 16 rounds of the 48-bit subkeys, to encrypt data of a fixed length of 64 bits. The apparent key's remaining 8 bits are utilized to verify for parity. In decryption, the same process is employed in reverse [30, 31]. **Figure 10** shows an example of DES encryption.

Even though this algorithm has been widely adopted due to its speed and ease of use, it suffers from a serious security weakness in reality. The use of DES with a short key makes it very fragile, especially using a brute force attack, which is easy to use in this case. In addition, there are many attacks, such as Davie's attack and offensive Linear and differential cryptanalysis, which are theoretical attacks [30, 31].

An improved version of the encryption algorithm has been created to solve the security issues with DES. This method is as simple as applying the DES algorithm precisely three times. We now have three keys, each of which is 56 bits long. As a result, the implementation technique differed in the keys utilized. There were several versions because the relationship of the three keys affects the extent of the algorithm's power in the previously described. Triple DES (3DES), which used three distinct keys with a total of 156 actual bits, was thought to be very powerful [28]. However, 3DES will not be used by the end of 2023 as we move to more secure generations for encryption [32].

Stream cipher	**Key size$_{bits}$**	**Data size $_{bits}$**	**Rounds**	**Speed$_{CPB}$**
RC4	1–2048	2046	1	7
SALSA20	128 256	512	20	3.91
SOSEMANUK	128–256	32	20–32	5.6

Table 2.
Stream cipher comparison.

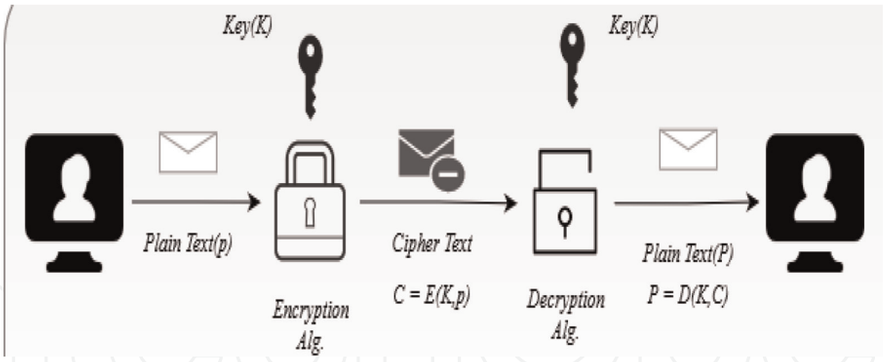

Figure 10.
DES algorithm [30].

Blowfish is a symmetric cipher technique that uses a 64-bit block and a variable-length encryption key as needed. In terms of speed, Blowfish is a good algorithm, but the amount of security it provides varies depending on the length of the key employed. As a result, even though no genuine threats have been detected, it has gotten less attention than other algorithms [32–33].

AES is one of the most famous and prominent symmetric encryption algorithms that has been introduced to be a quantum leap in this field. AES has outstanding performance and an excellent security level compared to its peers.

AES deals with data blocks with a fixed size of 128 bits in length, in addition to providing flexibility in choosing the size of the key according to the required degree of security. From here, it appears that AES has three versions according to the size of the key, namely AES-128, AES-192, and AES-256 with 10, 12, and 14 rounds, respectively. Each process uses several operations to encrypt a data block [34–36]. **Figure 11** represents the flow of the AES algorithm.

The working mechanism of AES is based on the use of the design principle known as the permutation and substitution network, and this mechanism is represented by using the following arithmetic operations:

- *SubBytes:* Using a predefined look-up table known as Rijndael S-Box, each byte will be replaced with another one. Without any linear relation.

- *ShiftRows:* The row elements are swapped by shifting them cyclically to the left.

- *MixColoumns:* Using a linear transformation relationship, the change of all column elements is combined so that they affect each other to increase the level of difficulty through the propagation property.

- *AddRoundKey:* The data cells are combined with the subkey cells generated for this round using XOR operation.

The need for key expansion comes from the fact that each AES round needs a key of a specific length based on the criteria mentioned earlier. Therefore, when using AES-128, we need 11 keys depending on the number of rounds. Key derivation is done using the AES Key Schedule algorithm, which expands the key using a key schedule [34, 35].

AES distinguished itself from its peers in improving its performance for systems dealing with large amounts of data by integrating these steps and running them on a

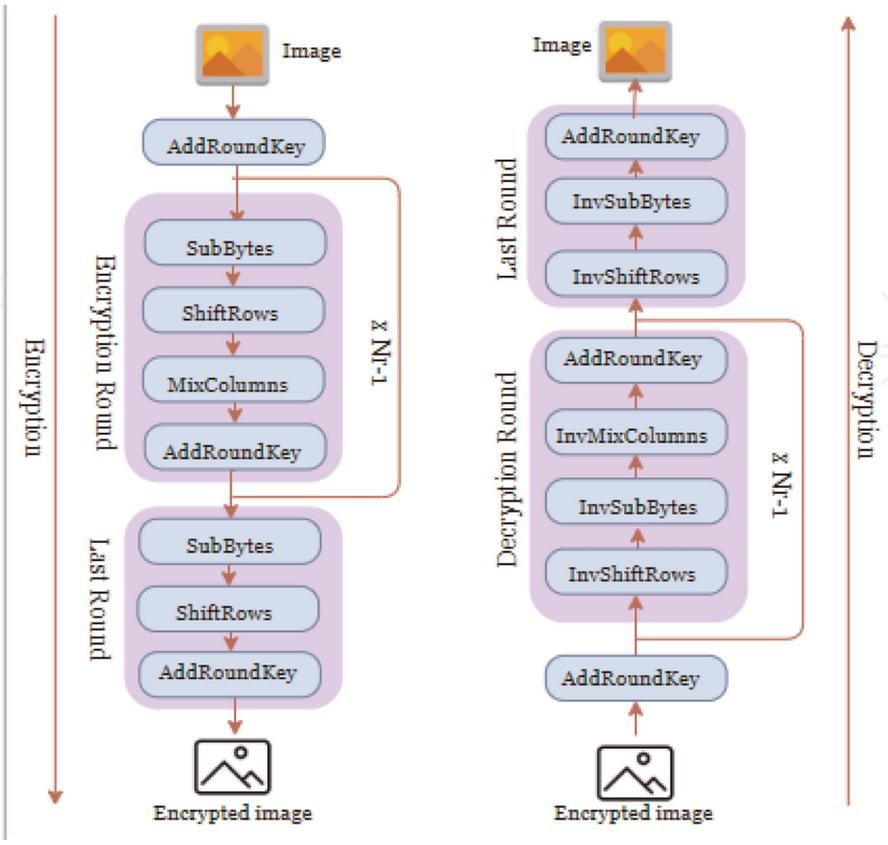

Figure 11.
AES algorithm [31].

byte-oriented approach. This approach only converts its arithmetic operations into a series of look-up tables [35].

3. Modes of operation

In Block Cipher, a fixed size block is handled at a time. Usually, the data size is much larger than the block size. Hence, the data is divided into a set of blocks. Each block is encrypted as one unit, the relationship, and dependency between encrypted blocks relaying on the encryption mode. Several modes have been developed to accommodate the variety of applications that will use Encryption. The process of selecting the required mode depends on many factors such as error propagation, the level of security, pre-processing, parallelization, and the speed of Encryption and decryption [12, 36]. These modes are as follows:

- *Electronic Code Book (ECB)*: It is an explicit and imperative coding process. It is considered the simplest since the text is split and each block is encrypted independently [36].

- *Cipher Block Chaining (CBC)*: This mode has constituted a development from the ECB. The block encryption process has become dependent on the result of the previous block encryption, which increased the data dependency on each other and made it possible non-deterministic. In this mode, the plaintext XOR-ed with the result of the previous block encryption before the encryption process [36].

- *Cipher Feedback (CFB)*: Looking at the CBC mode, this mode also relies on the result of the previous block as feedback to the present block and some other variables to increase the resistance to attacks [36].

- *Output Feedback (OFB)*: There is no difference between it and CFB except in some minor details that increased the resistance to bit errors and reduced the relationship of Encryption to plaintext [36].

- *Counter (CTR)*: It is a counter-based CFB. This mode is mainly based on maintaining the synchronization of the counter between the sender and receiver [36].

In general terms, without going into details of each mode. **Table 3** compares these modes.

After discussing the previous block cipher algorithms such as DES, 3DES, Blowfish, and AES, after reviewing the definition and specifications of each, **Table 4** provides a brief comparison of these algorithms.

From this comparison, we found that the stream has better performance and complexity, but it is not guaranteeing the diffusion, can be reversed easily, and it is providing less security. Because of that, we conclude that the block cipher is better solution since it provides more security in the case of text-based and image-based encryption.

3.1 Symmetric cipher summary

In this section, we summarize the symmetric cipher algorithms. **Table 5** compare stream and block cipher algorithms.

3.2 Cryptography summary

After discussing the cryptography algorithms and classifying them into Asymmetric and Symmetric, we reviewed their definition and specifications of each type. **Table 6** provides a brief comparison of these algorithms.

Mode		ECB	CBC	CFB	OFB	CTR
Padding Required		Yes	Yes	No	No	No
Error Propagation		No	All next block	Next block	No	No
Parallel	Enc	Yes	No	No	No	Yes
	Dec	Yes	Yes	Yes	No	Yes
Pre-Comp		No	No	No	Key	Yes
$Speed_{/5}$	Enc	5	2	1	3	4
	Dec	2	1	4	3	5
Security		Low	High	High	High	Medium
As Stream		No	No	Yes	Yes	Yes

Table 3.
Encryption modes comparison.

	DES	3DES		Blowfish	AES		
Key $_{bits}$	56	112	168	32-448	128	192	256
Block $_{bits}$	64	64		64	128		
Rounds	16	48		16	10	12	14
Security	Not	Secure		Moderate	Secure		
Speed	Slow	Very Slow		Fast	Fast		
Scalability	No	Yes		Yes	Yes		

Table 4.
Block cipher comparison.

	Stream	Block
Design	Complex	Simple
Data	Handle 1 byte at a time	Split data into a set of blocks
Number of bits	Depending on Block size	1 bit
Complexity	High	Low
Speed	Fast	Slow
Resources	Require more resources	Require fewer resources
Confusion and Diffusion	Confusion	Confusion and diffusion
Reversing	Simple	Hard
	Cannot take block cipher properties	It can be as a stream,

Table 5.
Stream vs. block cipher.

3.3 Lightweight cryptography

NIST defined LWC as a cryptosystem whose features have been optimized to meet the requirements of devices of varying specifications, especially resource-constrained devices [37]. From this definition, we conclude that all cryptography terms can be LWC if it is possible to legalize its need for resources to ensure the desired effect. Thus, asymmetric Encryption is an exception due to its complexity and demand for high resources. On the other hand, symmetric Encryption can be used in these systems if it is properly exploited.

Depending on the critical challenges mentioned before, we found that the LWC algorithm should use little memory and power and provide good performance while maintaining the required level of security [38]. Therefore, the factors of LWC requirements can be explained as follows [39]:

- *Key Size:* Longer Key size is better for security, but it requires more complexity and power.

- *Block Size:* smaller block size is more familiar with IoT since the big block size requires more CPU, memory, and power.

	Asymmetric	Symmetric
Keys	Two keys; one for Encryption and the other for decryption	Single Key for Encryption and decryption
Key Exchange	Not a problem	Big Problem
Relation between number of keys and receivers	# of Keys = (# of receivers) *2	# Of Keys = # of receivers
Cipher Size	Same or Larger than plain text size	Same or Smaller than plain text size
Speed	Slow	Fast
Data Size	Used for small data	Used for Large data
Provide	Confidentiality, authenticity, and non-repudiation	Confidentiality
Key Encryption and recourses utilization	High	Low
Examples	RSA, ElGamal, ECC, and DSA	RC4, Salsa20, Sosemanuk, DES, 3DES, Blowfish, and AES

Table 6.
Asymmetric cipher comparison.

- *The number of rounds:* Fewer rounds are better since the rounds require more computation and resources.

- *Structure:* The structure here is the way of managing the trade-off between all previous factors to find the optimal combination to ensure an acceptable level of performance and security.

Many LWC algorithms provide different performance and security strengths. And after studying many related studies, we find that there have been some trends in relying on stream cipher due to its high efficiency in terms of performance. Still, most of the algorithms were based on block cipher since it offers better security but with a significant performance improvement [38]. We highlight some of these LWC algorithms in the following sections depending on its base as a stream or block.

3.4 Stream LWC

This section presents some LWC algorithms based on stream cipher methodologies.

A4 is a very efficient lightweight stream cipher that uses LFSR and FCSR. The key feature of A4 is the ease of implementation and high security. In addition, A4 has proven itself in resistance to brute-force and algebraic attacks [39].

New Lightweight Stream Cipher (NLSC) is a chaos-based algorithm that uses an 80-bit secret key, two Nonlinear Feedback Shift Registers (NFCR), and three multiplexers. NFCR has good security, making it resistant to statistical attacks and providing good performance [38, 39].

3.5 Block LWC

This section presents some LWC algorithms based on block cipher methodologies.

PRESENT is an LCW algorithm that relies on Substitution-Permutation Network (SPN). It was suitable for limited hardware as it uses an 80-bit key. However, it was noted that it takes 32 rounds to encrypt 64 bits. Another version uses a 128-bit key, but it requires more computations [38].

GIFT is an enhanced PRESENT version; it uses a lighter S-Box with minimal rounds and a faster key scheduling algorithm. These properties enable it to provide more throughput. It is also available in more than one version depending on the required throughput. These versions are; GIFT-64 and GIFT-128. With a 64-bit block size that requires 28 rounds and a 128-bit block size that requires 40 rounds, respectively [38].

KATAN is an algorithm that outperforms PRESENT by saving 48% of the power. KATAN uses an 80-bit key and handles different text sizes 32, 48, and 64 bits. However, its downside is that it requires 254 rounds to complete this process [38].

The National Security Agency developed Simon as an improved algorithm that uses rounds cycles but uses a lot of arithmetic operations. It offers many different key sizes as 64-bit, 72-bit, 96-bit, 128-bit, 144-bit, 192-bit, and 256-bit that handle 32-bit, 48-bit, 64-bit, 96-bit, and 128-bit block size through 32, 36, 42, 44, 52, 54, 68, 69, and 72 rounds. While SPECK is the same as SIMON, it supports exact block sizes and keys, but 22, 23, 26-29, and 32-34 rounds [38].

RECTANGLE is a very LWC algorithm, which is different from PRESENT. It Relies on lighter SPN with 25 rounds. This reduced algorithm significantly speeded up the execution based on Bit-slice, as it relies on parallel swapping and replacement [40].

SIT is an algorithm that combines Feistel and SP network and takes five rounds to handle 64 bits of text with 64 bits of text as a key. It mainly consists of two parts: the first is for key expansion, and the second is for the encryption section. Key expansion

	Key $_{bits}$	**Block $_{bits}$**	**Rounds**	**Sec.**	**Characteristics**
A4	128	—	—	—	Secure High performance
NLSC	80	—	—	—	Secure Good performance.
PRESENT	80 128	64	32	80%	Low memory Suitable for small data
GIFT	128	64 128	28 40	85%	Simple Fast Key Scheduling. High throughput
KATAN	80	32 48 64	256	—	Inefficient. Low throughput Energy consuming
SIMON	64–256	32–128	32–72	67%	High performance Easy and Flexible
SPECK			22–34	58%	As SIMON but optimized for software
RECTANGLE	80 128	64	25	60%	Fast Hardware Friendly
SIT	64	64	5	—	Fast Key Scheduling High throughput Need low energy

Table 7.
LWC summary.

relies on simple operations such as concatenation, shifting, addition, and XOR. As a result, this algorithm achieves high throughput and low power consumption [38].

3.6 LWC summary

After discussing various LWC algorithms such as LSC, A4, NLSC, PRESENT, GIFT, KATAN, SIMON, and SPECK, RECTANGLE, and SIT. **Table 7** highlight the most comparison points between them.

4. Literature review

This section discusses the most recent related research. After studying these researches, we categorized them into two groups. The first group, including [40–50], reviews LWC and defines its essential requirements. The second group discusses AES versions that are proposed to be compatible with LWC requirements [51–60].

4.1 Lightweight cryptography related works

This section summarizes some researches that introduce the concept of LWC in terms of terminology, requirements, and how to implement them in line with the available capabilities.

Manifavas et al. [40] discussed lightweight encryption algorithms, focusing on streaming encryption, which provides high performance with simple operations, making it suitable for the capabilities of IoT devices, especially when the text length is unknown or continuous. The results showed the superiority of symmetric encryption in performance. Still, most of the streaming algorithms were not secure, as after analyzing 31 algorithms, it was found that only 6 were secure.

Buchanan et al. [41] emphasized the IoT's security and privacy challenges. Also, the researchers review the trends of designing lightweight algorithms after explaining alternatives to traditional cryptography methods that fit the composition of the IoT. Finally, after reviewing the challenges in terms of physical and software implementation, the study recommended that when developing LWC solutions, the following should be noted:

- Resorting to small blocks and a short key constitutes a security weakness and leads to faster wear of CBC mode.

- The number of operations is directly proportional to the size of the inputs; in lightweight symmetric cipher almost twice.

- The algorithm architecture must be adapted to new applications and better integrate with existing protocols.

Based on the previously mentioned recommendations, the following are the methods presented by this study that can be included when designing a lightweight security system for the IoT [41]:

- ***Hashing***: is a mathematical algorithm that assigns data of arbitrary size (often called "message") to a fixed-size bit matrix (the "message summary"). It is a one-way

function that is practically useless to reverse or reverse the account. Ideally, the only way to find a message that produces a particular hash is to forcibly search for potential inputs to see if they have a match or use a rainbow table of identical hash.

- ***Streaming***: it is a symmetric key cipher in which the plaintext is combined with a string of pseudorandom, keystream characters. In-stream cipher, each plaintext character is encoded with its corresponding character from the stream key to giving the ciphertext characters. An alternative name is state encoding, stream cipher, where the encoding of each character depends on the current state. The character is usually a bit and operation (XOR) or exclusive-or in practice.

- ***Block***: It's an encryption method that applies a deterministic algorithm along with a symmetric key to encrypt a block of text, rather than encrypting one bit at a time as in stream ciphers. For example, a typical block cipher, AES, encrypts 128-bit blocks with a key of a predefined length: 128, 192, or 256 bits. Block ciphers are Pseudo-Random-Permutation (PRP) families that operate on a fixed-size block of bits. PRPs are functions that are computationally indistinguishable from random permutations and, therefore, are considered reliable until their unreliability is proven.

Sehrawat et al. [42] presented a detailed comparison between several algorithms compatible with the IoT and after conducting cryptanalysis attacks. This study also showed that block ciphers had attracted the attention of many researchers as a basis for developing LWC algorithms. Finally, this study also recommended the requirements for the future of LWC algorithms.

Dutta et al. [43], reviewed the encryption solutions that can be used in the IoT by comparing some LWC that can fit with the nature of IoT devices. Researchers believe that symmetric encryption is the closest to suit the heart of the IoT. They also found that the modified AES algorithm provides a suitable security solution to the restrictions imposed by the capabilities of IoT devices after studying many algorithms like *DES, 3DES, Blowfish*, etc.

After choosing AES as a standard and reliable algorithm and achieving the desired goal, the researchers analyzed the performance of a set of versions of the algorithm implemented in previous studies by sorting them into two parts as follow [43]:

- **Recent Research Work on AES for IoT**: Many implementations achieved good results in high productivity, low energy, and minimal costs.

- **Recent Research Work on AES for IoT Focusing MixColumns and S-box**: The researchers focused on this aspect of the hardware implementation. Delay and reducing the area are the main goals of algorithm development, so the main challenge that exists to date is to improve Mix-column round and S-box operations. There are many implementations as the Serpent Algorithm that were previously developed to meet the challenges mentioned [43]. With these designs, we can provide good results to achieve these goals.

The researchers also presented a study of attacks on AES that should be monitored and found solutions such as Differential Fault Analysis Attacks and wireless interceptive side-channel attack techniques. These attacks can be resisted through the use of dummy keys and XOR operations [43].

Rajesh et al. [44] presented the Novel Tiny Symmetric encryption Algorithm (NTSA), which provides better confusion for each round which leads to better security level. The comparison centered with the TEA algorithm is considered one of the most attractive algorithms because of its ease of implementation and less memory usage. Its main problem is to use the same key for all rounds, which reduces the level of security and its poor performance. The results show that NTSA outperforms many other security algorithms and achieves better performance, making it more suitable for IoT and embedded devices.

Gunathilake et al. [45] discussed the future applications of LWC, how to implement it, and the challenges it faces. The study also touched on the existing LWC algorithms previously mentioned in our research and confirmed the effectiveness of the modified AES algorithm in this field.

Usman et al. [46] reviews the light encryption algorithms that fit the nature of the IoT after identifying the obstacles to using traditional algorithms, such as the low power capacity of the devices. Researchers believe that the security of big data flowing through the IoT is the main problem, as this weakness may overwhelm the advantages of IoT applications. Therefore, considering the capabilities of these devices represented in the low capacities, it was necessary to think of new methods that require simpler arithmetic operations and less memory while providing an acceptable degree of security. In addition to what has been mentioned, these methods must consider the diversity of devices, their different capabilities, and the protocols used to have the ability to integrate and adapt to this diversity. And now we still have the issue of privacy, as the IoT, with the vast amounts of data circulating, must provide the user with the possibility of appropriate control over his data [46]. The researchers considered that symmetric encryption is best suited for the IoT because asymmetric encryption requires higher capabilities. And the following are some of the symmetric encryption algorithms that have been reviewed [46].

Abutair et al. [47] believe that despite their importance, smart cities still face the challenge of balancing the quality of service and maintaining the privacy and security of information. This study summarized the difficulty of achieving this balance as follows:

- Design limitations and limited capabilities make this environment an easy target for hacking.
- The truth of the data may be injected to cause damage, leading to great disasters.
- Difficulty of building a standardized system due to different manufacturers.

The researchers studied many lightweight algorithms used in the IoT. Based on this study, an infrastructure has been proposed that provides a specific degree of privacy and security for the IoT. This study concluded that some modern algorithms such as *CLEFIA* and *TRIVIUM* achieved terrible results compared to the old algorithms, especially *TRIVIUM*, which gave disastrous results [48]. The study explains the structure of smart cities. Without going into details here, the aspect that concerns us is the necessity of providing IoT devices with algorithms that meet the guarantee of authentication, integration, and confidentiality to protect the network from threats. Such as *Corrupted Data, Replay Attacks, IP Spoofing, Identity Usurpation, DoS/DDoS Attacks, and, Data Leakage* [47]. This study presented a new design that depends on the capabilities of the device that will be added. Based on these capabilities, the

appropriate lightweight algorithm is selected for it. The mechanism of this design can be summarized as follows:

- *Input:* Device specifications
- *Knowledge Base:* minimal requirements for each lightweight algorithm.
- *Output:* The appropriate algorithm for this device.

After testing many algorithms by changing some factors, the researchers found that the algorithm closest to adapting to the majority of IoT devices is the *AES* algorithm, with the need to reduce its resources [47].

Ramadan et al. [48] introduced a LWC algorithm called LBC-IoT that handles 32-bit blocks with a key of up to 80 bits. This algorithm is based mainly on the Feistel structure, along with simple operations such as XOR that do not consume power and 4-bit S-boxes. The results indicate the strength of this algorithm against attacks in addition to its acceptable performance, and it is considered a promising algorithm for implementation on small and very restricted devices.

Periasamy et al. [49] proposed a lightweight block cipher mechanism that works on 8-bit processing, as their study indicates that this algorithm is superior to its counterparts. According to the researchers, this algorithm derives its strength from the strength of the encryption in the compensation boxes. In terms of performance, the design of the compensation boxes played marginally using the Multi sequence Linear Feedback Shift Register and reliance on simple operations such as XOR, shifting, and registers to reduce space required and optimization in power consumption and speed.

Thabit et al. [50], researchers introduced a New LWC Algorithm (NLCA) to secure cloud computing applications. This algorithm uses a 16-byte key based on Feistel and substitution permutation. This algorithm succeeded in achieving confusion and diffusion by introducing some logical operations into the algorithm's formula, such as Shifting, Swapping, and XOR. One of the advantages of this algorithm is the flexibility, such as AES, where the number of rounds and the length of the key are variable according to the application's needs. The results also indicate that this algorithm provides a good level of security and performance, which makes it suitable for these applications.

In this section, we discuss many LWC related researches. **Table 8**, focus on the key points that have been discussed in IoT cryptography related works and summarize them.

4.2 AES related works

In this section, we summarize some researches that present some AES-based system, discuss these systems and highlight the differences in these AES versions to reach the best possible ways to improve the performance and strength of this algorithm more.

Javed et al. [51], presented a new design for the AES algorithm to make it suitable for mobile devices and speed it up despite the limitations of the hardware specifications. After reviewing the mechanism of the standard AES algorithm, the researchers discuss the improvement that was made to AES implementation and the motives that were relied upon in this optimization as follows:

Study	Key points
2015 [40]	• Discuss many LWC algorithm. • It shows that the symmetric encryption is very good in performance, but most symmetric algorithms are not secure.
2018 [41]	• Discuss IoT security challenges. • Recommendations to be followed when developing LWC.
2018 [42]	• Compare many security algorithms that compatible with IoT. • It shows that the block cipher algorithms are more suitable to be used. • It also recommended the requirements for the future of LWC algorithms.
2019 [43]	• Discuss some encryption techniques that can be used in IoT. • Discuss AES algorithm.
2019 [44]	• Propose NTSA which provide good security level.
2019 [45]	• Discuss the future of LWC and its challenges.
2020 [46]	• Discuss some LWC algorithms. • It recommended the adoption of symmetric encryption because asymmetric encryption requires powerful resources, and this is what IoT devices lack.
2020 [47]	• Discuss many LWC algorithms. • The study found that modern algorithms did not meet the requirements due to poor results. • The study recommended the use of AES due to its strength, provided that it is configured to improve performance.
2021 [48]	• Propose LBC-IoT which provide very good performance with low power consumption.
2021 [49]	• Propose a new lightweight block cipher which provide a good security and performance.
2021 [50]	• Propose NLCA which used to secure the cloud, and it provide very good security with accepted performance.

Table 8.
LWC related works summary.

- This optimization used a 10-byte look-up table for round constant and two 256-bytes look-up tables for S-box and InvS-box. The constant round means that the three rightmost bytes are always 0. Thus, XOR performed only on the leftmost byte of the word. The round constant differs from one round to the other.

- ***In MixColumns***, the multiplication with 02 can be performed by a left shift and bitwise XOR with 1b.

- ***In ShiftRow***, Using the row index as a specific number (i), each row is rotated to the left by i. This implies that the first row will not be rotated.

- ***In RoundKey***, a rounded key is added to the State matrix by a simple bitwise XOR operation: a sum in the field GF (2^8). Each round key is obtained from the key schedule.

- There are two ways to implement Key scheduling: (1) key unrolling (2) On the fly key generation. This study implements key unrolling because that *On the fly key*

generation approach is costly in clock rounds and need 16 bytes of additional memory to store the last round keys for the decryption [51].

The results of this study showed that the performance of the proposed method gives better results, as it provides 3 times better encryption speed and is about 20 times better in round keys calculations. This design outperformed its predecessor by 20 times while reading data from the hard disk and encrypting it if the data was greater or equal to 1 MB [53].

Abhijith et al. [52], presented an improved model for implementing the AES algorithm by slicing and integrating the internal processes of the algorithm. This new version used Block-Ram and 10 levels of pipelines to improve efficiency and productivity. The results indicate that this enhanced version significantly enhances performance and the possibility of integrating it with other systems.

Bui et al. [53] worked on finding an improved version of AES in several ways. First, reduce the combinational logic and number of records by organizing the data path. Second, the clock gateway strategy, key expansion, and minimization of data activities contributed to reducing the algorithm's energy use. Here are the modifications that have been implemented to achieve the above improvements:

- By using the Low Power S-Box, power consumption is reduced.
- Logic relationships were reduced by manipulating data by columns after eliminating ShiftRow.
- Using a special mechanism to load data and encryption keys limits the number of records.
- Finally, the clock gate scheme worked in reducing energy consumption.

These modifications were additions that can be used without modifying the algorithm. As for the fundamental alterations in the algorithm, they were represented as follows [53]:

- ***Thirty-Two-Bit Datapath Optimizations:*** The Advanced Low Power Encryption Standard (AES) can be used in smaller applications such as small-scale IoT devices. Proposed 32-bit AES data paths to meet low energy consumption and small space requirements. We only use the 32-bit data path in MixColumns.
- ***Substitution Box:*** The S-box takes several input bits (m) and converts them into a certain number of output bits (n), where n is not necessarily equal to m. m $\times$ n S-box can be performed as a search table with 2 million words each n bits. Fixed tables are usually used.
- ***Key Expansion Optimizations:*** The expansion was implemented in VHDL, resulting in ascending design and test methodology. This choice also ensures that the code can be transferred to different vendors' devices. The code and simulation were manufactured using Altera MAX + PLUS II version 7.21 Student Edition. The FPGA family was selected for execution from Altera Flex 10 K. It's part of an 8-bit execution with a 128-bit block and a 128-bit key. Because the goal

of improvement is to reduce consumption, to suit it for mobile applications, the structure is directed to minimize space.

The results show that the proposed version offers the same PRESENT algorithm in energy use. Also, the proposed system is resistant to the attack of power correlation analysis with less than 20,000 traces, which seeks to expose the data path. Also, the data path in case of parallelism provides it with more robustness. Finally, this design uses different key sizes, which contributes to providing various levels of security as needed [55].

Mamoun et al. [54] provided a comprehensive explanation of the AES algorithm. The study presented a new model for the AES algorithm to enhance its security level by adding an XOR operation to an extra byte of s-box and using an additional random key. The results indicate that this modification contributed to improving the level of AES security variably due to the randomness of the added key. The results also showed that this modification improved confusion and increased time security.

Umer et al. [55] tested AES using different techniques depending on the resources of the target devices, the results were characterized by varying in nature according to the techniques used. Among these techniques were used; Parallelization and storage of s-box and key expansion, as it has been noted that the introduction of such technologies helps in optimizing the exploitation of resources to provide better results.

Daoud et al. [56], the researchers present an optimization of the AES algorithm using Vivado High-Level Synthesis (HLS), and their results show significant progress in increasing the throughput of the proposed algorithm, which was implemented on the FPGA only using flip flops and look-up tables. Since optimizing commands in Hardware Description Languages (HDL) is not easy and time-consuming, HLS improves the algorithm with less effort. HLS is an automated process that deals with high-level programming languages such as C that is used to ease the struggles that HDL requires in the development process, debugging, and provide flexibility in meeting system requirements. HLS tool synthesized compiled core AES functions in an RTL block, and sub-functions were divided into sub-blocks at higher system levels. Below is a review of the improvements that this study made to the AES algorithm [58]:

- ***Key Expansion-based Implementation:*** key expansion process combined with the encryption process so that the two processes will run simultaneously during each round.

- ***SW-based Implementation:*** Key extension process is performed before the encryption process to obtain 11 different 128-bit keys based on AES-128 design.

- ***High Throughput-based Optimization:*** The algorithm has made some special optimizations to increase the encryption throughput.

The main objective of this study was to achieve the maximum throughput in encryption. The process that most positively affected the results is integrating key expansion with encryption. By comparing the effects of frequency, productivity, and area utilization, it appears to us that the proposed design in this study has outperformed the previous strategies [56].

Proceeding from the fact that the AES algorithm is considered the best secure algorithm currently available and can be adapted to IoT devices. Rokan et al. [57] provided an integrated security system for the IoT called *Modified Lightweight AES*

(***MLAES***) that includes two integrated systems; The first one is a *Secure Encryption* based on a lightweight version AES integrated with Chaos Maps. The second is a *Secure Authentication* using a chaotic hash function based on SHA3-256-bits. The following is a review of the three main phases of this system:

- ***Lightweight Modified AES:*** The goal of mitigating and optimizing AES is to reduce computational complexity, execution time and reduce required iterations and memory used. One of the most important modifications is the use of 4 chaos keys, which increases the randomness of results, which means enhancing system security. The first modification in the algorithm uses *shifting operations*, *data blocks*, and *logical functions*. MLAES uses two sub-boxes, each dealing with 64 bits of data. The second modification is to make the number of times of rounds and ShiftRow are executed dynamically based on a dynamic number. This number is generated depending on some chaos keys that change with each iteration. Finally, the last modification is to eliminate the MixColumns operation due it its complexity and high execution time by replacing it with some XOR operations, SHA3-128, and shift operations.

- ***Modified Sub-Bytes(S-Box):*** The s-box represents one of the complex operations in MLAES and is directly related to the degree of security of the design; S-Box takes 128 bits of data and divides it into 16-bit blocks. Every 64 bits of data is sent to a sub-S-box, where the system contains 2 S-Boxes. The S-Box shifted after each iteration using K to change its values.

- ***The Proposed IoT Security System:*** As mentioned earlier, besides the MLAES encryption process described in the previous points, the proposed system includes a hashing stage using SHA3-256.

The study results indicate that despite the modification to AES, the level of security remained strong, in addition to the significant improvement in its performance and the specifications required for its operation. Perhaps the most prominent result was that this system passed the NIST tests, which means that the system is resistant to linear differential attacks and brute force attacks [57].

Farooq et al. [58], given the discrepancy between the capabilities of IoT devices, explored five implementations of the AES algorithm. These applications use modifications and improvements to the AES algorithm. The applications indicate the disparity in the results, as each of these applications fits a specific category of IoT devices. Therefore, the study recommended moving away from comprehensiveness and not limiting encryption to one algorithm for all devices, but instead relying on the device's capabilities to choose the optimal AES version for use.

Nagalakshmi et al. [59], given the discrepancy between the capabilities of IoT devices, presented some strategies for implementing AES with a set of other systems to suit these devices of varying powers, and the study also touched on the use of LFSR. The results indicate a security improvement, the ability to check signatures, and random checks without significantly affecting performance.

Salim et al. [60] presented the development of an AES algorithm called multi-key AES. The name came concerning the fact that this proposal uses the AES algorithm but uses several keys as the secret key is used to configure a variable number of keys using ECC. The study specialized in implementing this algorithm in the IoT, provided that it is used on devices capable of running this algorithm. The results indicated that this

Study	Key points
2010 [40]	• Use new look-up tables for S-box and InvS-box. • Optimize MixColumn, ShiftRow, and RoundKey. • These optimizations enhance AES performance.
2017 [41]	• Reduce combinational logic and number of records. • Use Low- Power S-Box, and clock gate scheme. • Eliminate ShiftRow. • These optimizations enhance AES performance.
2019 [42]	• Using Vivado HLS which enhance the throughput of AES.
2019 [43]	• Propose MLAES which provide a secure encryption AES-based algorithm and secure authentication used chaotic hash function. • Enhance AES by use 4 chaos keys to improve security. And use two sub-boxes.
2014 [44]	• Improve AES by use slicing and integrating processes, block-RAM, and 10 level pipelines. • These modifications enhance the performance of AES.
2017 [45]	• Provide a comprehensive study of AES. • Enhance AES by adding an XOR operation to an extra byte which enhance the security of AES. • This enhancement improves the time security and confusion.
2021 [46]	• Use AES but with several keys based on ECC. • This optimization improves AES security, but it did not enhance its performance.
2020 [47]	• Using 5 modified AES models and studying their results, the study recommended not relying on the same algorithm on all devices, but rather choosing the appropriate algorithm for the capabilities of each device.
2020 [48]	• Modifying AES by using LFSR. • This modification enhances the security of AES, but it did not improve its performance.
2017 [49]	• By testing AES, this study shows that the improvement in AES performance can be done by parallelization storage of S-Box, and key expansion.

Table 9.
AES related works summary.

modification did not affect the algorithm's performance, but it contributed to improving its security.

In this section, we discuss many AES-based related researches. **Table 9**, present the summary of some researches that worked on modifying AES to adapt it with IoT.

5. Evaluation

This section presents the ways of evaluating algorithms and a brief discussion of this study.

5.1 Evaluation

The evaluation process should address performance evaluation and security evaluation to ensure the power of the algorithm. To evaluate performance, we will initially need to calculate the following:

- ***Execution Time***: is one of the essential parameters for evaluation performance. It measures the time needed to encrypt and decrypt a specific data size [61, 62].

- ***Throughput***: it reflects how much data can be processed during a time. It presents the average of data in kb divided by the average Encryption or Decryption time.

As for security, we will initially need to account for:

- ***Key Time Security:*** the time to attack the algorithm using brute force. Which is related to key size [9], [63].

- ***Histogram:*** study the uniformity of data distribution [9, 61].

- ***Confusion:*** study the relationship between ***ciphertext*** and ***key***; this relation should be robust. In simple words, the changing of 1-bit in secret key should lead to a significant change in ciphertext [62].

- ***Diffusion:*** study the relationship between ***ciphertext*** and ***plain text***; in simple words, changing 1-bit in plain text should affect the ciphertext highly [62].

- ***NIST Tests:*** These tests attempt to test the randomness of binary sequences produced by an algorithm. These tests focus on different types of non-randomness that could exist in a binary sequence. It was released by the National Institution of Standards and Technology (NIST) as a suite for testing PRNGs that contains 188 tests, including 15 main tests [9, 63].

		ECB	CBC	CFB	OFB	CTR
Key Time Security				2^{128}		
Enc. Time (s)	1	1.136	1.132	1.224	1.374	1.355
	2	228.899	243.123	246.811	242.619	241.472
Dec. Time (s)	1	1.32	1.41	1.14	1.05	1.12
	2	299.26	306.90	242.76	248.48	244.63
Enc. Throughput	1	1278.24	1063.87	1098.20	1192.91	1132.80
	2	1134.55	1037.38	1028.12	1051.15	1046.48
Dec. Throughput	1	987.11	866.45	1091.38	1169.09	1113.62
	2	885.73	826.89	1038.35	1047.64	1040.17
Histogram		8194.44	240.91	251.80	274.40	257.44
Confusion (%)		50.08	50.04	49.93	50.11	49.95
Diffusion (%)		0.08	50.16	49.82	0.01	0.01
NIST		13	15	15	15	15

[1] *155 KB Data.*
[2] *31 MB Data.*

Table 10.
AES evaluation results.

5.2 Summary

Based on all that was mentioned previously, studies have confirmed that stream cipher provides better performance than block cipher. Still, a block cipher is superior to a stream cipher in terms of security especially when we looking to better confidentiality. Some previous studies also indicated that lightweight stream cipher did not succeed much on the security front. From here, we can be sure that the basis in our research should be based on a block cipher with its security strength while trying to improve it in the level of performance [64].

We believe that using a recognized and standard algorithm to improve it would be better at the current stage. Most previous studies confirmed that the choice fell on AES due to its superiority. In appendix A, we review the summary of the results of the AES algorithm test in terms of performance and security to be a starting point for improvement [64]. These results are shown in **Table 10**.

These results showed that AES provide an acceptable degree of security according to this evaluation criteria, such as Key security, Histogram, NIST, Confusion, and Diffusion. But to prove that, we will use more security tests in future work such as Mapping, Correlation, Unified averaged changed intensity, and Number of Changing pixel Rate. On other hand, the result of performance testing can be improved by changing or replacing some core functions on AES.

6. Conclusion

In this chapter, a detailed study of computer security has been conducted. After clarifying different kinds of cryptography, LWC has been addressed, considering its basics and requirements. Some of the presented algorithms highlight the essential needs for LWC algorithms and the importance of making them compatible with the resources of IoT devices. This study also discussed the latest studies related to each of Lightweight Cryptography, Lightweight AES-based algorithms, and the most prominent evaluation criteria used to judge the suitability of an algorithm. Finally, this study presented the results of testing the AES algorithm according to the specified criteria. We believe that these results constitute a starting point for future work as promising results in the field of LWC algorithms and their suitability to the resources of IoT devices.

Toward Lightweight Cryptography: A Survey
DOI: http://dx.doi.org/10.5772/intechopen.109334

Author details

Mohammed Abujoodeh*, Liana Tamimi and Radwan Tahboub
College of IT and Computer Engineering, Palestine Polytechnic University, Hebron, Palestine

*Address all correspondence to: 131089@ppu.edu.ps

References

[1] Alfred Y. Network Security. Malaysia: Asia Pacific University; 2019. pp. 5-11. DOI: 10.13140/RG.2.2.19900.59526

[2] Manuj A. Network Security with pfSense: Architect, Deploy, and Operate Enterprise-Grade Firewalls. 1st ed. Birmingham: PACKT Publishing; 2018

[3] William S. Cryptography and Network Security: Principles and Practice. 8th ed. London: Pearson; 2017

[4] Mohammed Z, Ahmed E. Internet of things applications, challenges and related future technologies. Journal of World Scientific News. 2017;**67**(2):126-148

[5] Available from: https://internetofthingsagenda.techtarget.com/Ultimate-IoT-implementation-guide-for-businesses [Accessed: February 15, 2022]

[6] Mista S, Roy C, Mukherjee A. Introduction to Industrial Internet of Things and Industry 4.0. 1st ed. Florida: CRC Press; 2021

[7] Qabajeh L. A more secure and scalable routing protocol for mobile ad hoc networks. Security and Communication Networks. 2013;**6**:286-308

[8] Makhdoom I, Abolhasan M, Ni W. Blockchain for IoT: The challenges and a way forward. International Council for Evangelical Theological Education. 2018: 594-605

[9] Salhab O, Jweihan N, AbuJoodeh M, Abutaha M, Farajallah M. Survey paper: Pseudo-random number generators and security tests. Journal of Theoretical and Applied Information Technology. 2018;**96**: 1951-1970

[10] Abutaha M, Farajallah M, Tahboub RM. Survey paper: Cryptography is the science of information security. International Journal of Computer Science and Security. 2011;**5**:475

[11] Available from: https://geek-university.com/ccna-security/aaa-explained [Accessed: February 15, 2022]

[12] Elminaam DA, Abdual-Kader HM, Hadhoud MM. Evaluating the performance of symmetric encryption algorithms. IJ Network Security. 2010; **10**:216-222

[13] Guanrong C, Ybin M, Charles C. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons & Fractals. 2004;**21**:749-761

[14] Zhu S. Algorithm design of secure data message transmission based on Openssl and Vpn. Journal of Theoretical & Applied Information Technology. 2013;**48**:562-569

[15] Bellare M, Rogaway P. Optimal asymmetric encryption. In: Workshop on the Theory and Application of Cryptographic Techniques. Berlin, Heidelberg: Springer; 1994. pp. 92-111

[16] Simmons G. Symmetric and asymmetric Encryption. ACM Computing Surveys (CSUR). 1979;**1**:305-330

[17] Shamir A, Adleman L, Rivest R. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM. 1978;**21**: 120-126

[18] Alsadeh A, Karakra A. A-RSA: Augmented RSA. 40th conf of SAI Computing. 2016:1016-1023

[19] Gamal T. A public-key cryptosystem and a signature scheme based on discrete

logarithms. IEEE Transactions on Information Theory. 1985;**31**:469-472

[20] Available from: https://www.quest10.com/p/33937/el-gamal-cryptography-algorithm-1/ [Accessed: February 15, 2022]

[21] Maurer U, Wolf S. The Diffie–Hellman protocol. Designs Codes and Cryptography. 2000;**19**:147-171

[22] Mihailescu M, Nita S. "Elliptic-curve cryptography", elliptic-curve cryptography. In: Pro Cryptography and Cryptanalysis. Berkeley, CA: Apress; 2021. DOI: 10.1007/978-1-4842-6367-9_1

[23] Al-Absi M, Abdullaev A, Absi A, Sain M, Lee H. Cryptography survey of DSS and DSA. In: Advances in Materials and Manufacturing Engineering, Lecture Notes in Mechanical Engineering. Singapore: Springer; 2020. pp. 661-669

[24] Agrawal M, Mishra P. A comparative survey on symmetric key encryption techniques. International Journal on Computer Science and Engineering. 2012;**4**:877-882

[25] Stallings W. The RC4 stream encryption algorithm. In: Cryptography and Network Security, Prentice Hall, 2005

[26] Mister S, Tavares S. Cryptanalysis of RC4-like ciphers. Selected Areas in Cryptography. 1998;**1556**:131-143

[27] Robshaw M, Billet O. New Stream Cipher Designs: The eSTREAM Finalists. New York: Springer; 2008

[28] Bernstein D. The Salsa20 family of stream ciphers. In: Robshaw M, Billet O, editors. New Stream Cipher Designs. Lecture Notes in Computer Science. Vol. 4986. Berlin, Heidelberg: Springer; 2008

[29] Berbain C, Billet O, Canteaut A, Courtois N, Gilbert H, Henri L, et al. SOSEMANUK: A Fast Software-Oriented Stream Cipher. New York: Springer; 2008. pp. 98-118

[30] Coppersmith D, Holloway C, Matyas S, Zunic N. The data encryption standard. Information Security Technical Report. 1997;**2**:22-24

[31] Daemen J, Joan, Rijmen V. The data encryption standard. In: The Design of Rijndael. Springer; 2002, 2002. pp. 81-87

[32] Available from: https://www.cryptomathic.com/news-events/blog/3des-is-officially-being-retired. [Accessed: February 15, 2022]

[33] Bhat P, Deepthi. Comparison of MD5 and blowfish algorithm. International Journal of Innovative Research in Science Engineering and Technology. 2016;**5**:506-511

[34] Kalaiselvi RC, Vennila M. An analysis of AES, RSA, and blowfish - A review. The International Journal of Analytical and Experimental Modal Analysis. 2020;**XII**:568-588

[35] Blumenthal U, Fabio M, Keith M. The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-Based Security Model. Vol. No. RFC 3826. USA: Bell Labs; 2004

[36] Dworkin M. Recommendation for Block Cipher Modes of Operation: Methods and Techniques. SP: NIST; 2001. pp. 800-38A

[37] Cruz-Cunha M, Portela I. Handbook of Research on Digital Crime, Cyberspace Security, and Information Assurance. 1st ed. Pennsylvania: IGI Global; 2014

[38] Thakor V, Razzaque MA, Khandaker M. Lightweight cryptography

algorithms for resource-constrained IoT devices: A review, comparison, and research opportunities. IEEE Access. 2021;**9**:28177-28193

[39] Muhammad R, Quazi M, Rafiqul I. Current lightweight cryptography protocols in Smart City IoT networks: A survey. ArXiv. 2010;**00852**: 2020

[40] Manifavas H, Hatzivasilis G, Fysarakis K, Papaefstathiou J. A survey of lightweight stream ciphers for embedded systems. Security and Communication Networks. 2015;**9**: 1226-1246

[41] Buchanan W, Li S, Asif R. Lightweight cryptography methods. Journal of Cyber Security Technology. 2018;**1**:187-201

[42] Sehrawat D, Gill N. Lightweight block ciphers for IoT based applications: A review. International Journal of Applied Engineering Research. 2018;**13**: 2258-2270

[43] Dutta I, Ghosh B, Bayoumi N. Lightweight Cryptography for Internet of Insecure Things: A Survey. 2019 IEEE 9th Annual Computing and Communication Workshop and Conference; 2019. pp. 0475-0481

[44] S. Rajesh, V. Paul, V. Menon, and M. Khosravi, Mohammad, "A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices", Symmetry. 2019;**11**(2):293

[45] Gunathilake N, Buchanan W, Asif R, Rameez. Next Generation Lightweight Cryptography for Smart IoT Devices: Implementation, Challenges and Applications. IEEE 5th World Forum on Internet of Things (WF-IoT); 2019. pp. 707-710

[46] Usman M. Lightweight encryption for the low powered IOT devices. arXiv. 2020;**2012**:00193

[47] Abu-tair M, Djahel S, Perry P, Scotney B, Zia U, Carracedo J, et al. Towards secure and privacy-preserving IoT enabled smart home: Architecture and experimental study. Sensor. 2020; **20**:6131

[48] Ramadan R, Aboshosha B, Yadav K, Alseadoon I, Kashout M, Elhoseny M. LBC-IoT: Lightweight block cipher for IoT constraint devices. CMC-computers Materials Continua. 2021;**67**:3563-3579

[49] Prakasam P, Madheswaran M, Sujith KP, Shohel S. An enhanced energy efficient lightweight cryptography method for various IoT devices. ICT Express. 2021;**7**:487-492

[50] Thabit F, Alhomdy S, Al-ahdal A, Abdulrazzaq, Jagtap P. A new lightweight cryptographic algorithm for enhancing data security In cloud computing. Global Transitions. 2021;**2**: 91-99

[51] Javed A. Fast Implementation of AES on Mobile Devices. Proc. 8th Int. Netw. Conf.; 2010. pp. 133-142

[52] Abhijith P, Goswami M, Tadi S, Pandey K. Optimized architecture for AES. Cryptology ePrint Archive: Report. 2014;**1**:540

[53] Bui D, Puschini D, Bacles-Min S, Beigné E, Tran X-T. AES Datapath optimization strategies for low-power low-energy multisecurity-level internet-of-thing applications. IEEE Transactions on Very Large-Scale Integration (VLSI) Systems. 2017;**25**:3281-3290

[54] Mamun A, Rahman S, Shaon T, Hossain A. Security analysis of AES and enhancing its security by modifying

S-box with an additional byte. International Journal of Computer Networks & Communications. 2017;**9**: 69-88

[55] Farooq U, Aslam F. Comparative analysis of different AES implementation techniques for efficient resource usage and better performance of an FPGA. Journal of King Saud University - Computer and Information Sciences. 2017;**29**:295-302

[56] Daoud L, Hussein F, Rafla N. Optimization of Advanced Encryption Standard (AES) Using Vivado High Level Synthesis (HLS). Proceedings of 34th International Conference on Computers and Their Applications. 2019: 36-44

[57] Naif R, Abdul-Majeed GH, Farhan AK. Secure IOT system based on chaos-modified lightweight AES. 2019 International Conference on Advanced Science and Engineering (ICOASE). 2019:1-6

[58] Farooq U, Mushtaq M, Bhatti M. Efficient AES implementation for better resource usage and performance of IoTs. In: CYBER 2020 - 5th International Conference on Cyber-Technologies and Cyber-Systems. France: Nice; 2020

[59] Nagalakshmi E, Mohan V, Kumar D. AES datapath optimization strategies for low-power low-energy multi security-level internet-of-thing applications. International Journal of Advanced Research in Science, Engineering and Technology. 2020;**2**:347-355

[60] Salim K, Alalak S, Jawad M. Improved image security in internet of thing (IoT) using multiple key AES. Baghdad Science Journal. 2021;**18**: 417-429

[61] Jagtap S, Thabit F, Alhomdy S. Security analysis and performance evaluation of a new lightweight cryptographic algorithm for cloud computing environment. Global Transitions Proceedings. 2021;**2**:100-110

[62] Coskun B, Memon N. Confusion/diffusion capabilities of some robust hash functions. 40th Conference Information Sciences and Systems. 2006; **CISS'6**:1188-1193

[63] Farajallah M, Abutaha M, Abujoodeh M, Salhab O, Jweihan N. Pseudo-random number generator based on look-up table and chaotic maps. Journal of Theoretical and Applied Information Technology. 2020;**98**:3130

[64] Abujoodeh M, Tamimi L, Tahboub R. Exploring and Adapting AES Algorithm for Optimal Use as a Lightweight IoT Crypto Algorithm, [master thesis]. Palestine: Palestine Polytechnic University; 2022. Available from: https://scholar.ppu.edu/handle/123456789/8635