Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2021, 8(1):174-188

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Comprehensive Study of Attacks and Cryptographic Measures for Internet of Things Devices

Mahmoud A. Khattab

Basic Engineering Science Department, Benha Faculty of Engineering, Benha University, Egypt

Abstract Recently, there has been a great interest in connecting all things (objects) together anywhere and anytime on the Internet, which today is known as the Internet of Things (IOT). The IOT architecture is made up of different layers, and each layer is subject to different attacks / threats. On the other hand, there are several security measures in place to counter these attacks. Cryptography algorithms are one of the most important security measures to counter these attacks. In this paper, several potential attacks on IoT devices are described. A comprehensive study of the most important cryptography algorithms (symmetric-key, asymmetric-key and Lightweight) was made, with an explanation of the mathematical background of them.

Keywords Internet of things security, Internet of things attacks, Symmetric-key Cryptography, Public-key Cryptography, Lightweight Cryptography

1. Introduction

These days the Internet has become an integral part of our daily life and it affects human life in unimaginable ways. Starting in the 1960's, communication between two computers was possible via a computer network, and then the matter developed until it became possible to communicate between any two objects through the web and this is called the era of the Internet of Things (IOT) [1, 2]. The term Internet of things was coined in 1999 by Kevin Ashton in his presentation which means in its context connecting things, sensors, actuators, and other smart technologies with each other i.e., enable person-to-person and object to object communication.

The Internet of Things aims to make various tasks easier for users by connecting different objects to the web [3]. It is expected that the number of objects connected to the network will reach 75 billion in 2025 [4]. With this enabling the huge number of things to communicate over the network, there are many applications for IOT [5-13], some of which are listed in Table 1. Despite the many applications of IOT and its widespread benefits, there are many challenges hindering IoT including privacy and security problems, massive information problems, things communication problems, over-reliance on technology, loss of job, widespread of Malware and intrusion [14, 15]. The security sector is the most important challenge of IOT, as no one can trust a machine unless he is certain that it is fully secured [16].

The types of attacks that IoT devices are exposed to varies, some are physical (the devices can be physically tampered with) and others are imperceptible (for example an attack on transmitted data or software). There are many papers describing these attacks as well as suggested solutions for them [17-19].

One of the most important measures used to counter these attacks is cryptography which is a powerful and effective data protection tool that preserves and transmits data in a way that only the sender and receiver can process and understand. In the transmission stage, there is a term in cryptography called encryption, which is the conversion of plaintext into encrypted text (ciphertext). In the reception stage, the term decryption is the conversion of the ciphertext into plaintext. It is worth noting that data encryption includes large computational processes and resources, which of course require sufficient storage space and energy, and this is not available with most IoT devices with limited redundant resources that usually rely on batteries, therefore, for this reason, it will be difficult to guarantee the high security of IoT devices [20].

Table 1: Some applications of internet of things

No.	Application
1	Smart Home (Home Automation)
2	Smart City (Smart transport, Smart Water System)
3	Smart Factory
4	Health Care and Fitness
5	Military
6	Social Life and Entertainment
7	Retail
8	Agriculture
9	Supply Chain Management and Logistics
10	Emergency
12	User Interaction
13	Culture and Tourism
14	Smart Environment (home, office, plant)
15	Energy Conservation
16	Disaster Alerting and Recovery
17	Solid Waste Management
18	Smart Metering
20	Consumer asset Tracking
21	Smart Grid
22	Pilgrims Monitoring

In this article, we review the most important attacks against IoT devices, as well as the various encryption methods to address these attacks with a short description of them, and accordingly we present previous contributions in using these methods with IoT devices.

2. Definition and Architecture of IOT

Most of the technology giants as well as the research community in this field have dealt with the IOT and its architecture from different perspectives, so we will find different definitions and architectures between one institution and another, also between one researcher and another.

The Internet of Things has many different terms of use such as Internet of Everything, Networked Society, and Industrial Internet of Things. The Internet of Things is a very broad vision. Research into the Internet of Things is still in its infancy. Therefore, to this day, there is no standard definition of the Internet of Things, but the more realistic definition illustrated in [2, 21] is that "*the Internet of Things allows people and things to communicate anytime, anywhere with anything and anyone, ideally using any network and any service*" (see Figure 1).

For architecture, there is no single consensus on the architecture of IOT, but there are proposals from several researchers [1, 9, 21-23], Some suggested a three-layer architecture, while others suggested a four-layer and five-layer architectures.

The three-layer architecture was introduced in the early stages of research in this area. It has three layers, namely, the perception layer, network later, and application layer (see Figure 2(a)). The perception layer is the physical layer, which contains sensors for sensing and collecting information about the surrounding environment. It senses some physical parameters or identifies other smart objects in the environment. The network layer is the core of IOT systems, it is responsible for connecting to other smart things, network devices, and servers. Its features are also used for transmitting and processing sensor data. It transmits the information gathered by the perception layer. It contains the software and hardware instrumentations of internet network in addition to the management and information centers. The application layer is responsible for delivering application specific services to the user. It defines various applications in which IOT can be deployed, for example, smart homes, smart cities, and smart health. The application layer's target is to converge between the IoT social needs and industrial technology [9].

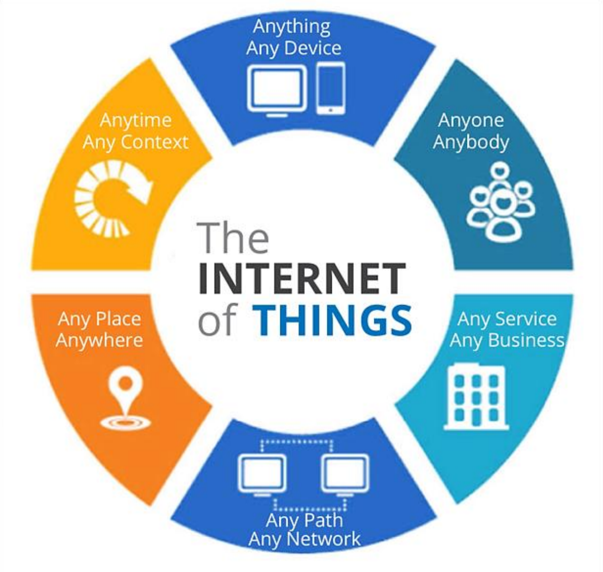

Figure 1: Definition of IOT [21]

The four-layer architecture is like the three-layer architecture but is increased by a four layer called the middleware layer, which is entrusted with providing management services by users or applications [24]. It contains the firmware and operating system code (see Figure 2(b)).

The five-layer architecture has been proposed due to the expected development of IOT. These five layers are the perception layer, the transfer layer, the processing layer, the application layer, and the business layer [9]. The role of the perception and application layer is the same as that of the three-layer architecture. The role of the transport layer is to transfer sensor data from the perception layer to the processing layer and vice versa through networks such as wireless, 3G, LAN, Bluetooth, RFID and NFC. The role of the processing layer, also known as the middleware layer, is to store, analyze, and process massive amounts of data that come from the transport layer. It can manage and provide a variety of services to the lower classes. It uses many technologies such as databases, cloud computing, and big data processing units. Finally, the role of the business layer is to manage the entire Internet of Things system, including applications, business models, profit, and user privacy (See Figure 2(c)).

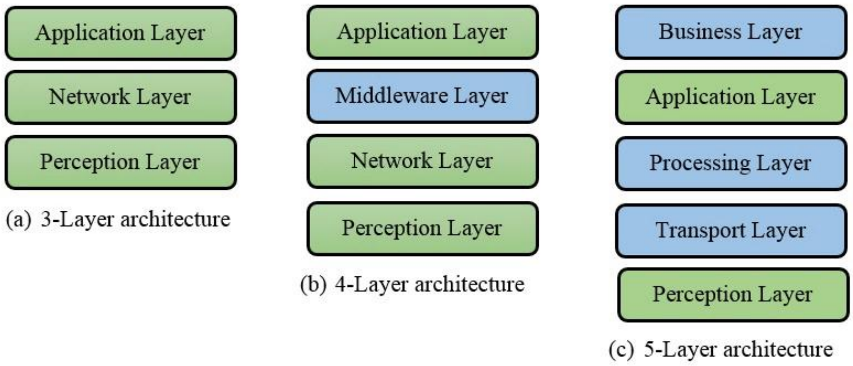

Figure 2: Different architectures of IOT

3. Attacks and Threats on IOT Devices

Internet of things devices are subject to many attacks, some of which are visible, and some are imperceptible. In this section we will present possible attacks on IoT systems from different perspectives and in the different layers of the IoT.

3.1. Perception Layer Attacks

Because the perception layer contains the sensors and actuators of the IOT systems, it is normal to be exposed to many attacks, especially physical attacks. These attacks include but are not limited to:

- **Unauthorized access to tags**: on the internet of things, RFID systems are used a lot, but due to the problem of collision of tags and ineffective matching in the authentication process, the adversary can access these tags without permission.
- **Node capture attacks**: also called a node response attack, it is an attempt by the opponent to capture and control the node by adding one or more nodes to the network that uses the same identifier as another node in the network, and thus may leak all information and thus threaten the security of the entire system.
- **Tag cloning**: The cybercriminal can access the tag, request a true copy of it, and then reproduce it and expose it to danger so that it cannot be distinguished from the original mark.
- **False data injection attacks**: also called Spoofing attack. In this type of attack, the impersonator broadcasts or injects false information into RFID systems. Then when received, IoT applications can then reverse the wrong notes or make incorrect offers.
- **Denial of service attack (DOS)**: This type of attack causes loss of network resources and renders the service unavailable by launching ineffective activities that consume resources.
- **Replay attack**: In this attack, the attacker sends some old messages (or resends and modifies the original messages), in which case it is easy to access previously sent data.
- **Side channel attack (SCA)**: The attacker attacks the encryption devices with side channel leakage information. There are different types of attacks that join under SCA such as timing attacks, power analysis attacks, fault analysis attacks, electromagnetic radiation attacks and environmental attacks.
- **Mass Node authentication problem**: Since there is no efficient authentication mechanism for the collective nodes, the attacker can exploit this flaw to attack the entire system.

3.2. Network layer attacks

The network layer is exposed to many attacks because the current Internet network architecture is specifically designed for interpersonal communication and does not necessarily apply to communication between devices. The network layer is not only exposed to traditional security problems (illegal access networks, eavesdropping on information, corruption of confidentiality and integrity, DOS attack, Man-in-the middle attack, virus invasion, exploit attacks) but it can also be exposed to compatibility problems, group security problems, authentication, and privacy issues [19, 25].

- **Sybil Attack**: In this attack, the attacker takes control of large parts of the network as the attacker uses multiple logical entities on the same physical node. This type of attack is like a clone-ID attack.
- **Buffer reservation attack**: An attacker blocks buffer space by sending incomplete packets and this inevitably results in denial of service as other fragment packages are discarded.
- **Malicious code injection**: An attacker can control the device and stop it from running by exploiting a vulnerability in the program where malicious code is injected.
- **Sinkhole Attack**: In this attack, the system is deceived that the data has been received from the other side, as the attacker makes the compromised node appear attractive to the nearby nodes, and this necessarily leads to the flow of data from any node to the compromised node and leads to reduced packets as well as consuming a large portion of energy.
- **Encryption attacks**:
  1) **Side Channel attack (SCA)**: The attacker obtains the encryption key by using the information emitted from the encryption devices, this information is not information about the massage or encrypted message, but information about the energy and time required to perform the encryption algorithm and others. There are different types of this attack such as timing attack, simple and differential power analysis, and differential fault analysis [17, 18].
  2) **Man-in-the-middle-attack**: The attacker listens secretly between two legal parties who believe they are communicating with each other directly without an intruder between them. This is of course with

the aim of deleting, changing, or delaying messages exchanged during communication between the two parties [17, 18].

3) **Cryptanalysis attack**: The cryptanalysis is the science concerned with breaking encryption algorithms by knowing the encryption key. This attack depends on the attacker's knowledge of one of the following (cipher text only, known plain text, chosen cipher text, chosen plain text, adaptive chosen plain text, adaptive chosen cipher text). Algebraic attacks are among the most powerful methods used by the attacker to carry out their attack, such as using the Gropner basis algorithm, F4 algorithm and F5 algorithm, for more details about cryptanalysis attack, we recommended [26].

3.3. Application layer attacks

The network layer is also vulnerable to many attacks. Some of these attacks were mentioned in [24, 25] as follows:

- **Identity Authentication and Data Access Permissions**: Due to the multiplicity of users for different applications, it is possible for illegal users to interfere, so there must be an effective authentication technique to face this type of attack.
- **Data Protection and Recovery**: Due to the lack of a perfect data processing algorithm, this may cause data loss and many damages as it is certain that the contact data includes user privacy.
- **The Ability of Dealing with Mass-data**: Due to the huge amount of data transferred and many nodes as well as the complex surrounding environment, this may lead to data loss or network interruption, so there must be the ability to adapt to the requirements and speed of data processing.
- **Software Vulnerabilities**: Due to the difference in the program codes among programmers, this causes security flaws in the programs, so the attacker can exploit the security holes and carry out his illegal purposes.
- **Forgery or counterfeiting**: The attacker copies the contents of the devices and then modifies them.
- **Spear-Phishing Attack**: In this attack, the adversary can gain access to the victim's credentials by luring him to open his email. This attack is also called an email spoofing attack.
- **Insecure software**: An attacker can exploit vulnerabilities in the unsafe firmware of the Internet of Things and can inject various types of malicious code into the system to steal data.
- **Sniffing Attack**: Sniffing applications, can be imposed on the system by the attacker and thus can obtain network information and thus damage the system.

Most of the researchers offered their suggestions to counter attacks on IoT devices, for example, in [25], the researchers provided several security measures for each layer separately such as measures (Access Control, Data Encryption, The Based on IPSec Security Channel, Cryptography Technology Scheme, Physical Security Scheme, Key Management, Secret Key Algorithms, Security Routing Protocol, Intrusion Detection Technology, Authentication, Physical Security Design) for the perception layer, measures (specific authentication cohesive mechanism, the end-to-end authentication and key agreement mechanism, Public Key Infrastructure, WPKI for wireless, Security routing, Intrusion detection) for the network layer and finally measures (Across Heterogeneous Network Authentication and Key Agreement, Increasing the Awareness of Safety, Strengthen Information Security Management) for the application layer. Based on the suggestions submitted by many researchers, we find great interest in resolving data encryption in all layers of the Internet of Things.

4. Cryptography as a counter Measure

4.1. Cryptography algorithms

Due to the big data transferred between the different layers of IoT systems, and the restrictions on the devices used on IOT (such as RFID), and to maintain the confidentiality of this data, strong and fast cryptographic algorithms must be in place in the accounts to avoid the illegal attack on the data by the attackers, as cryptography deals with privacy, authentication, and integrity. This is what calls us to review the different cryptography algorithms. In fact, cryptography divides into two main types, symmetric-key cryptography, and asymmetric-key cryptography (public-key cryptography). However, given the special handling of IoT devices, it is possible to add a third type called lightweight cryptography.

A) Symmetric-key Cryptography

In this type, the sender and receiver use the same encryption key to perform the encryption process [27, 28] (see Figure 3). There are different algorithms based on this principle, classic symmetric-key cryptosystems (see Table 2) and modern symmetric-key cryptosystem (see Table 3). The modern cipher algorithms are more complex and secure than classical cipher algorithms [41].

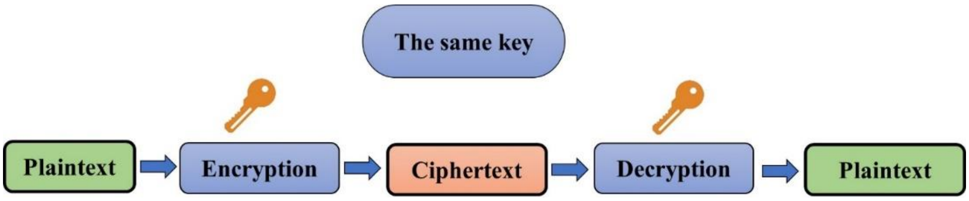

Figure 3: Symmetric-key cryptosystem process

Table 2: Examples of Classic Symmetric-key Cryptosystems

No	Algorithm Name
1	Caesar Cipher
2	Playfair
3	Hill Cipher
4	Vigenere Cipher
5	Vernam cipher
6	One-time Pad
7	Rail Fence
8	Root cipher
9	Columnar Transposition
10	Double Transposition
11	Myszkowski Transposition
12	Disrupted Transposition
13	Grills

Table 3: Examples of Modern Symmetric-key Cryptosystems

No.	Algorithm Name	Year
1	Data Encryption Standard (DES)	1979
2	Rivest Cipher 2 (RC2)	1987
3	Rivest Cipher 4 (RC4)	1987
4	International Data Encryption Algorithm (IDEA)	1991
5	Blowfish	1993
6	Rivest Cipher 5 (RC5)	1994
7	Triple Data Encryption Standard (3DES)	1995
8	Advanced Encryption Standard (AES)	1998
9	Serpent	1998
10	Twofish	1998
11	Rivest Cipher 6 (RC6)	1998
12	SEED	1998
13	Camellia	2000
14	Hierocrypt (Hierocrypt-1 and Hierocrypt-3)	2000

Therefore, we will pay attention to illustrating the most important modern and commonly used cryptosystems, as follows:

- **Data Encryption Standard (DES)**: DES is one of the oldest block cipher cryptosystems as it was published by IBM in 1977. 64-bit block data size is encrypted using a 56-bit key, where the first 8 bits are discarded. The cryptosystem process is done using 16 rounds in which it is initially split the 64-bit block into two halves of 32 bits each. This system uses both permutation boxes (P-boxes) and substitution boxes (S-Boxes).
- **Trible Data Encryption Standard (3DES)**: The use of a 56-bit key size in traditional DES is no longer appropriate to counteract modern cryptographic analysis techniques, so the same old version of

DES was used but repeated three times on each data block. It was first published in 1978 and uses 168-bit key size (3 times of DES key size).

- **International Data Encryption Algorithm (IDEA)**: It is also called Improved Proposed Encryption Standard (IPES) and it was first published in 1991. It encrypts 64-bit data blocks using a 128-bit key size, and this is in only 8.5 rounds. It is recommended to use it in real time communication systems or in wireless communications. It has many uses such as smart cards, emails, and more.
- **Blowfish**: It was introduced by Bruce Schneier in 1993 to replace the DES and IDEA systems, as it is characterized by its speed, simplicity, and security. It encrypts a block with a size of 64 bits and using different key sizes ranging from 32 bits to 448 bits as it relies on key-dependent S-boxes and uses 14 rounds to perform the encryption process.
- **Rivest Cipher 2 (RC2)**: It was designed by Ronald Rivest in 1987. It encrypts a data of block size 64 bits with different key sizes from 8 to 128 bits, but the default key size is 64 bits. It uses 18 rounds, 16 of type mixing and 2 of type mashing.
- **Rivest Cipher 4 (RC4)**: Designed by Ronald Rivest in 1987. It is stream cipher with key sizes ranges from 40 to 2048 bits. It is suitable for use in Wireless Equivalent Private (WEP) for wireless card and TLS.
- **Rivest Cipher 5 (RC5)**: It was designed in 1994 by Ronald Rivest. It is characterized by its simplicity and its difference from other block cipher as it encrypts data blocks of different sizes (32, 64 and 128 bits) and different key size (0 to 2040 bits) and different number of rounds (0 to 255). It is suggested in original case to use a block size of 64 bits, a 128-bit key and 12 rounds. It is suitable for smart card and other devices as it requires small memory for execution. But the disadvantage is that its strength depends on the size of the key used, the longest it is, the stronger it is, and vice versa.
- **Rivest Cipher 6 (RC6)**: It is, in fact, an improvement to RC5 algorithm by use an extra multiplication operation does not present in RC5 algorithm. It was introduced in 1997 by Ron Rivest, Matt Robshaw, Ray Sidney, and Yiqun Lisa Yin. It is a block cipher algorithm has a block size of 128 bits and supports key sizes of 128, 192, and 256 bits up to 2040-bits.
- **Twofish**: It is related to Blowfish algorithm, introduced by Bruce Schneier in 1998. It is a block cipher with block size 128 bits and key sizes 128, 192 or 256 bits. It is distinguished by its use of pre-computed key-dependent S-boxes, and a relatively complex key schedule.
- **SEED**: It is a block cipher algorithm introduced by Korea Internet & Security Agency (KISA) in 1998. It uses the same size for data block and the key which is 128 bits. It is rarely used outside of South Korea. The encryption process is done in 16 rounds and using 8×8 S-boxes.
- **Advanced Encryption Standard (AES)**: Also called Rijndael, developed by Vincent Rijmen and Joan Daemen in 1998. It was created as an alternative of 3DES cipher. In fact, Rijndael is a family of ciphers with different key and block sizes. Three members of Rijndael selected by National institute of standard & technology (NIST) in 2001 each with block size of 128 bits but with different key sizes 128, 192 and 256 bits where it was chosen from the top 5 encryption systems (MARS, RC6, Rijndael, Serpent and Twofish). The number of rounds is related to the key size used such that it uses 10 rounds for 128-bit key size, 12 rounds for 192-bits key size and 14 rounds for 256-bits key size. The strength of AES depends on the strength of the s-box used as it is the only nonlinear component of the entire system. The s-box composition is based on the equation $s(x) = Ax^{-1} + B\ (mod\ p)$ where x is an element of the used field (Finite field 2^8 for standard AES), x^{-1} is the multiplicative inverse of element x, A is the affine matrix, B is the additive constant and p is the irreducible polynomial. There are many researches that make different modifications on the standard s-box used in traditional AES in order to improve the performance of the whole system [29-35].
- **Skipjack**: It is a block cipher algorithm introduced by National Security Agency (NSA) in 1998 uses 80 bits key size for encryption and decryption of 64 bits data blocks. The number of rounds is 32 rounds.
- **Serpent**: It was designed by Ross Anderson, Eli Biham, and Lars Knudsen in 1998. It was ranked second in the selection after Rijndael by NIST. It has a block size 128 bits and supports a key size of

128, 192 and 256 which requires 32 rounds. Each round applies one of eight 4×4 S-boxes 32 times in parallel.

- **Camellia**: It is a type of block ciphers that was developed by Mitsubishi Electric and NTT of Japan in 2000 and was named for Camellia japonica. Camellia is a block cipher with 128 bits block size and key sizes of 128, 192 and 256 bits. It is used frequently low-cost smart cards and high-speed network systems. The key size is related to the number of rounds, as 18 rounds are used when using 128 bits as the key size, but 24 rounds are used when using the 192 bits or 256 bits key size. Camellia uses 4 S-boxes each S-box is 8×8.
- **Hierocrypt**: It is a block cipher introduced by Toshiba in 2000. There are two versions of Hierocrypt codes (Hierocrypt-1 and Hierocrypt-3). There is a clear similarity between both versions, but they differ mainly in block size, key size, and number of rounds. Hierocrypt-1 uses 64 bits for data block, 128 bits key size and 6.5 rounds. Hierocrypt-3 uses 128 bits for data blocks, different key sizes 128, 192 or 256 bits and different number of rounds depends on the key size such that it uses 6.5 rounds for 128 bits key size, 7.5 rounds for 192 bits key size and 8.5 rounds for 256 bits key size. In [42], the performance of Hierocrypt-3 was improved by changing the S-Box used in traditional Hierocrypt-3, and the results showed better performance, especially against the algebraic attacks.

B) Asymmetric-Key Cryptography

In this type there are two keys, one for encryption and this is the public key and another for decryption, which is the private key [27, 36] (see Figure 4). There are different algorithms based on this principle such as Diffie-Hellman key exchange algorithm, RSA (Rivest–Shamir–Adleman) algorithm, etc., (see Table 4).

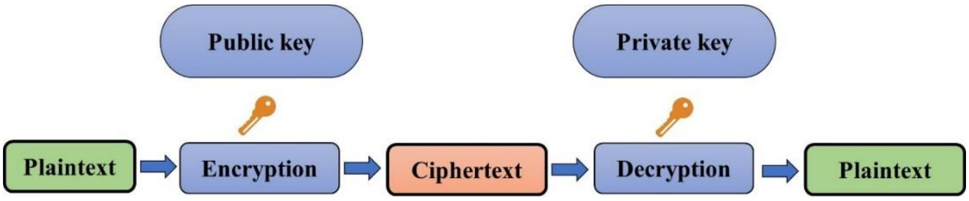

Figure 4: Asymmetric-key cryptosystem process

Table 4: Examples of Public-key cryptosystems

No	Related problem	Algorithm name
1	Integer Factorization	Rivest–Shamir–Adleman (RSA) Cryptosystem Rabin-Williams cryptosystem
2	Discrete Logarithm	Diffi-Hellman key Exchange (DHKE) Protocol ElGammal Cryptosystem Digital signature algorithm (DSA)
3	Elliptic Curve discrete logarithm	Elliptic Curve Diffi-Hellman (ECDHA) Protocol Elliptic Curve Digital Signature Algorithm (ECDSA) Diffi-Hellman Digital Signature Algorithm (DHDSA)
4	Lattice problems	Goldreich–Goldwasser–Halevi (GGH) NTRUEncrypt and Micciancio’s cryptosystem
5	Others	Short Range Natural Numbers Algorithm (SRNN) cryptosystem

The most important public-key cryptosystems and widely used are illustrated as follows:

- **Diffi-Hellman key Exchange (DHKE)**[27, 37]: DHKE algorithm is not an encryption algorithm, but it is a secure key exchange algorithm (key distribution), introduced in 1976. It is based on the difficulty of computing discrete logarithms. Suppose that there are two users A and B wished to exchange the key, then the Diffi-Hellman algorithm for this purpose is as follow:
  1) The two users A and B are agree upon a prime number p and a generator g.
  2) User A select a random integer $a < p$ and compute $g^a mod\, p$ and send it publicly to user B.
  3) User B select independently a random integer $b < p$ and compute $g^b mod\, p$ and send it to user A.
  4) User A compute the secret key as $K = (g^b mod\, p)^a (mod\, p)$
  5) User B compute the secret key as $K = (g^a mod\, p)^b (mod\, p)$

- **Rivest–Shamir–Adleman (RSA) Cryptosystem**[27, 37, 38]: It is one of the most important asymmetric-key cryptosystems which introduced in 1977. RSA is based on the Factoring problem i.e., difficulty of factoring the product of two large prime numbers. For some plaintext block message m, the following is the algorithm of RSA cryptosystem for encryption and decryption:
  1) Choose two large prime numbers p and q.
  2) Compute $n = pq$.
  3) Compute $\emptyset(n) = (p-1)(q-1)$.
  4) Choose an integer e to be a public key such that $1 < e < \emptyset(n)$ and $\gcd\left(e, \emptyset(n)\right) = 1$. Where $\gcd\left(e, \emptyset(n)\right)$ is the greatest common divisor of both e and $\emptyset(n)$.
  5) Choose the private key d such that $ed = 1(mod\left(\emptyset(n)\right)$.
  6) The public key is (n, e) while the private key is (n, d).
  7) For encryption, compute $c = m^e (mod\ n)$, where c is the ciphertext.
  8) For decryption, compute $m = c^d mod\ n = (m^e)^d mod\ n = m^{ed} mod\ n$, where m is the original message.
- **Rabin-Williams Cryptosystem**[39]: This cryptosystem is very similar to the RSA cryptosystem, published in 1979 and depends on the difficulty of factoring integers. To encrypt a plaintext m, the following procedures are followed:
  1) Choose two large prime numbers p and q such that $p \equiv q \equiv 3(mod\ 4)$.
  2) Compute $n = pq$. The public key is then n and the private key is (p, q).
  3) For encryption, compute $c = m^2 (mod\ n)$, where c is the ciphertext.
  4) For decryption, we follow this procedure: compute m_p and m_q where

     $$m_p = C^{\frac{p+1}{4}}(mod\ p), \qquad m_q = C^{\frac{q+1}{4}}(mod\ q)$$

     Test that $m_p^2 \equiv C\ (mod\ p)$ and $m_q^2 \equiv C\ (mod\ q)$ then use the Chinese remainder theorem to obtain four possibilities for $m\ (mod\ n)$ such that $m = \pm m_p (mod\ p)$ and $m = \pm m_q (mod\ q)$.
- **ElGammal Cryptosystem**[27, 38]: It is a public-key cryptosystem introduced in 1986. It can be defined over any cyclic group G with order q and generator g. The ElGammal cryptosystem difficulty is based on computing discrete logarithms in G. First part of ElGammal algorithm is to generate the key which is as follows:
  1) Choose and integer x randomly from $\{1, \ldots, q-1\}$.
  2) Compute $h = g^x$.
  3) User B publish the public key which is consists of the values (G, q, g, h).
  4) User A encrypt a message M as follows: user A maps the message M to an element $m \in G$, then choose an integer y randomly from $\{1, \ldots, q-1\}$, compute $s = h^y$ then the ciphertext consists of (C_1, C_2) where $C_1 = g^y$, $C_2 = m.s$
  5) User B decrypt the ciphertext as follow: Compute $s = c_1^x$, where $C_1^x = g^{xy} = h^y$, then compute the inverse of s in the group G denoted s^{-1}, and finally compute $m = C_2 s^{-1}$ then map m back to the plaintext message M.
- **Digital Signature Algorithm (DSA)**[40, 41]: DSA was proposed by National institute of standards and technology in 1991. The digital signature works within the framework of public-key encryption systems based on the algebraic properties of modular exponentiation and the discrete logarithm problem. The algorithm uses two keys, a public key, and a private key. The private key is used to generate a digital signature for a message, and such a signature can be verified by using the corresponding public key. The digital signature provides message authentication, integrity, and non-repudiation.
- **Elliptic Curve Cryptography (ECC)**[27, 37]: The use of elliptic curves in cryptography was suggested in 1985 by Neal Koblitz and Victor S. Miller. The cryptography algorithms based on elliptic curves were used extensively from 2004 to 2005 as it is used in many applications such as key agreement and digital signatures. The elliptic curve over $Z_p, p > 3$, is the set of all pairs $(x, y) \in Z_p$ which fulfill $y^2 \equiv x^3 + ax + b\ (mod\ p)$ together with imaginary point of infinity O, where $a, b \in Z$ and Z_p is a prime field and

fulfill the condition $4a^3 + 27b^2 \neq 0\ (mod\ p)$. In Figure 5, the elliptic curve $y^2 = x^3 - 3x + 3$ over the set of real numbers R.

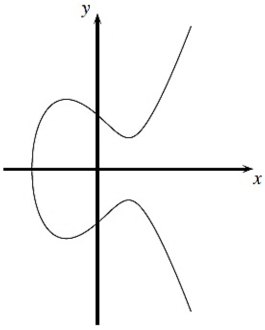

Figure 5: The Elliptic Curve $y^2 = x^3 - 3x + 3$ over real numbers R

- **Elliptic Curve-Diffi-Hellman (ECDH) [**27, 37**]:** It is a protocol used to exchange keys (public and private) using elliptic curves. The advantage of using elliptic curves in key exchange is the creation of small-sized keys rather than the use of large prime numbers in thetraditional key exchange of Diffi-Hellman. Both the sender and the receiver agree on an elliptic curve $E: y^2 \equiv x^3 + ax + b\ (mod\ p)$ with a primitive element $B = (x_B, y_B) \in E$ where a, b, p and B are the domain parameters. Each of the sender (say T) and receiver (say S) has a key pair that consists of a private key $d \in [1, \#E - 1]$where $\#E$ is the number of points on the chosen elliptic curve, and a public key $Q = (x, y)$, which is a point on the chosen elliptic curve. The protocol to be followed is as follows:
  1) The sender T and receiver S choose two integers d_T and d_S from $[1, \#E - 1]$
  2) The public key of sender is $d_T B = d_T(x_B, y_B) = (x_T, y_T) = Q_T$ and the public key of receiver is $d_S B = d_S(x_B, y_B) = (x_S, y_S) = Q_S$
  3) The joint secret key between the sender and receiver is then $d_T Q_S = d_T(x_S, y_s) = d_S(x_T, y_T) = d_S Q_T$
- **Goldreich–Goldwasser–Halevi (GGH) [**27, 42**]:** GGH is asymmetric-key cryptosystem that uses lattices i.e., it is a lattice-based cryptosystem. It was proposed and published in 1997 and is based on the problem of lattice reduction. Given a lattice L and a good basis B for L, it is easy to generate a vector which is close to a lattice point i.e., taking a lattice point and adding a small error vector e. But to return from this erroneous vector to the original lattice point a special basis is needed. The procedures for this cryptosystem are as follows:
  1) For a lattice L, choose a good basis B for L (i.e., short, and nearly orthogonal vectors) and a unimodular matrix U (a square matrix having determinant $+1$ or -1. The private key is then B and U.
  2) The public key is the basis $B^{'} = U.B$
  3) Given a message $M = (m_1, \dots, m_n)$ and an error vector $e = (e_1, e_2)$, compute the vector $v = M.B^{'}$, the ciphertext is then $c = v + e$
  4) To decrypt the message M, compute

$$\begin{aligned} c.B^{-1} &= (M.B^{'} + e)B^{-1} \\ &= M.B^{'}.B^{-1} + e.B^{-1} \\ &= M.U.B.B^{-1} + e.B^{-1} \\ &= M.U + e.B^{-1} \end{aligned}$$

  the term $e.B^{-1}$ is small enough to be removed, then the message M is then $M = M.U.U^{-1}$
- **NTRU cryptosystem**[43]: The NTRU cryptosystem introduced by Hostein, Pipher and Silverman in 1996 is a lattice-based cryptosystem. All arithmetic operations (addition and multiplication) in NTRU are done in polynomial rings:

$$\mathcal{R} = \frac{Z[x]}{x^N - 1}, \qquad \mathcal{R}_p = \frac{(Z/pZ)[x]}{x^N - 1}, \quad \mathcal{R}_q = \frac{(Z/qZ)[x]}{x^N - 1}$$

Where Z is the set of integers, Z/qZ is the set of integers modulo q and $N-1$ is the maximum degree of polynomials in the mentioned polynomial rings (for more details about quotient rings and convolution polynomial rings we recommended [44]). This cryptosystem is depending on choosing some parameters and keys as described in Table 5.

The NTRU cryptosystem is carried out in three stages, key generation, encryption, and decryption considering the parameters (N, p, q, d) as follows:

1) Choose two randomly polynomials $f(x)$ and $g(x)$ from the ring $\mathcal{R}$ that satisfy the conditions illustrated in Table 5.
2) Compute the inverse of $f(x)$ modulo q denoted $f_q(x)$ and the inverse of $f(x)$ modulo p denoted $f_p(x)$. If either $f_q(x)$ or $f_p(x)$ does not exist, a new $f(x)$ must be chosen.
3) Compute the public key $h(x) = pf_q(x) * g(x)\ mod\ q$, where $*$ is the convolution ring product operation. The private key is the pair $\left(f(x), f_p(x)\right)$ and the public key is $h(x)$.
4) To encrypt a plaintext $m(x) \in \mathcal{R}$ whose coefficients are between $-\frac{1}{2}p$ and $\frac{1}{2}p$, choose a random polynomial $r(x) \in \mathcal{R}$ that satisfy the conditions illustrated in Table 5, then computes $e(x) = ph(x) * r(x) + m(x)\ mod\ q$. The polynomial $e(x)$ is the ciphertext.
5) In the decryption stage, first compute $a(x) = f(x) * e(x)\ mod\ q$, then compute $b(x) = f_p(x) * a(x)\ mod\ p$. The polynomial $b(x)$ is the plaintext $m(x)$.

Note that, to avoid decryption error, the parameters p, q and d must be chosen probably and satisfy that $q > (6d+1)p$. For example, the choice (7,3,41,2) is guarantee that the decryption will work probably such that $41 = q > (6d+1)p = (6(2)+1)3 = 39$. In [45, 46], NTRU-like cryptosystem have been introduced to extend the traditional NTRU cryptosystem. In [47-50], contributions were made to improve performance of the standard NTRU cryptosystem.

Table 5: Parameters of NTRU cryptosystem

Parameter	Description
N	All polynomials in the convolution ring have degree $N-1$
q	Large integer prime number used in polynomials coefficient reduction (i.e., coefficients of polynomials in the convolution ring are reduced modulo q), satisfy that $\gcd(N, q) = 1$
p	Small integer prime number used in message coefficient reduction (i.e., coefficients of message polynomial is reduced modulo p), satisfy that $\gcd(p, q) = 1$ and $p < q$
d	An integer used to define the coefficients of both $f(x)$ and $g(x)$
$f(x)$	A polynomial that is belong to the set $T(d+1, d)$ which is the set of all polynomials in the convolution ring $\mathcal{R}$ having $(d+1)$ 1s, $(d) - 1s$ and the rest equal to 0. For example, $f(x) = x^6 - x^4 + x^3 + x^2 - 1 \in T(3,2)$. The polynomial $f(x)$ is the private key.
$g(x)$	A polynomial that is belong to the set $T(d, d)$. For example, $g(x) = x^6 + x^4 - x^2 - x \in T(2,2)$. The polynomial $g(x)$ is used with $f(x)$ to generate the public key $h(x)$.
$h(x)$	A polynomial $h(x) \in \mathcal{R}$ is the public key
$r(x)$	A polynomial $r(x) \in \mathcal{R}$ is a blinding polynomial used in encryption stage satisfy $r(x) \in T(d, d)$

- **Short Range Natural Numbers Algorithm (SRNN) [51]:** SRNN cryptosystem is in fact like RSA cryptosystem, but with some modifications, this modification increases the security of the method used. SPNN cryptosystem uses extremely large number that has two prime factors and two short range natural numbers. The encryption and decryption processes of this cryptosystem are as follow:
  1) Choose two large prime numbers p and q.
  2) Compute $n = pq$.
  3) Compute $\emptyset(n) = (p-1)(q-1)$.
  4) Choose an integer e to be a public key such that $1 < e < \emptyset(n)$ and $\gcd\left(e, \emptyset(n)\right) = 1$.
  5) Choose d such that $1 < d < \emptyset(n)$ and $ed = 1\left(mod\left(\emptyset(n)\right)\right)$.
  6) Choose short range natural number u randomly such that $u < \emptyset(n) - 1$

7) Choose another short natural number a such that $u < a < \emptyset(n)$, then compute u^a
8) The public key is then (n, e, u^a) and the private key is (d, a, u)
9) To encrypt a message m (represented in integer form), compute $c = (mu^a)^e(mod\ n)$, where C is the ciphertext.
10) For decryption, compute $m = (v^e c)^d(mod\ n)$, where m is the original message.

C) Lightweight cryptography

Because IoT devices are resource-limited and have many restrictions on the memory size, speed, power consumption, etc. This is what leads us to use cryptography algorithms other than the conventional cryptography algorithms due to our need for less memory, less computing resource, and less power supply to provide security solution that can work over resource-limited devices. So, we need to develop the Lightweight Cryptography (LWC). The lightweight cryptography is be simpler and faster than the conventional cryptography but is less secured. Block cipher is preferred over stream ciphering in an IoT environment since block cipher is uses confusion and diffusion properties whereas the stream cipher uses only confusion property.

In [52], the most important LWC cryptosystems are surveyed based on its structure (Substitution-Permutation network (SPN), Feistel network (FN), generalized Feistel network (GFN), Add-Rotate-XOR (ARX), Nonlinear feedback shift register (NLFSR), Hybrid) sorted in ascending order in years of introduction for each structure.

In [53], Researchers discussed several lightweight cryptographic algorithms that are supposed to replace the conventional cryptographic algorithms given the strict needs in the Internet of Things environment.

4.2. Cryptography and internet of things

There are many previous research literatures that used and compared the performance between different cryptographic algorithms in general and particularly in IoT devices. In the following some of these contributions

In [54], the performance, especially in power consumption, was compared between DES, AES and RSA when they were used to secure electronic medical records, as AES surpassed DES and RSA in power consumption.

In [55], the comparison was between AES and DES of different versions, BLOWFISH and RSA when applied in smart grid, but from the point of view execution time, the result was that AES is the best. Comparisons similar to previous research were made in [56, 57]. The result was a victory for AES.

In [58], a comparison was made between RC4, AES, DES, and RSA algorithms was measured on a wireless sensor network in terms of power consumption, and the result was that RC4 consumed less power.

In [59], a comparison was made between RC6, Twofish, Serpent, and Mars was measured on an Android smartphone device. The result was the lowest power consumption in favor of Twofish and RC6.

In [60], analyzation of execution time and memory used were made between RC6, AES, 3-DES, and RSA algorithms. The RC6 algorithm obtained the best results.

In [61], the encryption/decryption speed was measured for different block ciphers include AES, Twofish, Serpent, Camellia, BLOWFISH, SEED, IDEA, DES, and 3-DES using different C/C++ cryptography libraries. It is pointed out that AES is surpassed other algorithms.

In [62], a study was conducted on measuring encryption speed and battery consumption in laptops connected to a wireless network by encrypting text files, images and audio by using BLOWFISH, DES, 3-DES, RC2, RC6, and AES algorithms, where DES outperformed all algorithms.

In [63], the ECC was used as a means of encrypt the collected data from sensors to suggest the security of the communications in the WSN. The effectiveness of the use of the ECC has been demonstrated as an anti-attack method to various types of attacks such as brute force attack, replay attack, and sinkhole attack.

In [64], ECC and RSA were compared in data security in the sensor network for remote health monitoring of patients, and the results showed that ECC is faster than RSA in key generation and signature.

5. Conclusions

In this article, we made a comprehensive study on attacks and threats on the Internet of Things devices in its various layers in the three-layer architecture (perception, network, and application layers), as well as the various cryptographic algorithms (symmetric-key, asymmetric-key, and lightweight) and their use in securing data in Internet of things environment.

Developing a cryptographic system suitable for working in an IoT environment is bound by some restrictions such as less storage space, less energy consumption, speed, cost, and more. Therefore, researchers in this field have tended to develop lightweight encryption systems. However, until now there is no lightweight encryption system that fulfills all these requirements at once, as if the focus is on efficiency and speed, this comes at the expense of the storage space used, and so on. Therefore, we are looking forward to finding solutions to problems related to data encryption within the Internet of Things environment.

References

[1]. Sethi, Pallavi, and Smruti R. Sarangi. (2017). Internet of things: architectures, protocols, and applications. Journal of Electrical and Computer Engineering.

[2]. Perera, C., et al. (2013). Context aware computing for the internet of things: A survey.IEEE communications surveys & tutorials, 16(1): 414-454.

[3]. Chaouchi, H. (2013). The internet of things: Connecting objects to the web, John Wiley & Sons.

[4]. Statista, I. (2018). Internet of Things (iot) connected devices installed base worldwide from 2015 to 2025 (in billions).

[5]. Dimitrakopoulos, G. (2011). Intelligent transportation systems based on internet-connected vehicles: Fundamental research areas and challenges. In 2011 11th International Conference on ITS Telecommunications. IEEE.

[6]. Huynh, T., Y. Tan, and K. Tseng. (2011). Energy-aware wireless sensor network with ambient intelligence for smart LED lighting system control. In IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society. IEEE.

[7]. Zhao, W., C. Wang, and Y. Nakahira. (2011), Medical application on internet of things. IET.

[8]. Khan, R., et al. (2012). Future internet: the internet of things architecture, possible applications and key challenges. In 2012 10th international conference on frontiers of information technology. IEEE.

[9]. Ara, T., et al. (2016), Internet of Things architecture and applications: a survey. Indian Journal of Science and Technology, 9(45).

[10]. Kyas, O. (2017). How To Smart Home: A Step by Step Guide for Smart Homes & Building Automation, Key Concept Press.

[11]. Mehmood, Y., et al. (2017). Internet-of-things-based smart cities: Recent advances and challenges. IEEE Communications Magazine, 55(9): 16-24.

[12]. El-Wakeel, A.S., et al. (2018), Towards a practical crowd sensing system for road surface conditions monitoring. IEEE Internet of Things Journal, 5(6):4672-4685.

[13]. Ramson, S.J., S. Vishnu, and M. Shanmugam. (2020). Applications of Internet of Things (IoT)–An Overview. In 2020 5th International Conference on Devices, Circuits and Systems (ICDCS). IEEE.

[14]. Said, O. asnd M.J.I.J.o.C.N. Masud. (2013). Towards internet of things: Survey and future vision. International Journal of Computer Networks, 5(1):1-17.

[15]. Kuyoro, S., F. Osisanwo, and O. Akinsowon. (2015). Internet of things (iot): an overview. In 3rd international conference on advances in engineering sciences & applied mathematics.

[16]. Sadeeq, M.A., et al. (2018). Internet of Things security: a survey. In 2018 International Conference on Advanced Science and Engineering (ICOASE). IEEE.

[17]. Andrea, I., C. Chrysostomou, and G. Hadjichristofi. (2015). Internet of Things: Security vulnerabilities and challenges. In 2015 IEEE Symposium on Computers and Communication (ISCC). IEEE.

[18]. Deogirikar, J. and A. Vidhate. (2017). Security attacks in iot: A survey. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). IEEE.

[19]. Khan, M.A. and K.J.F.G.C.S. Salah, (2018), IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems, 82: 395-411.

[20]. Fernández-Caramés, T.M.J.I.I.o.T.J. (2019). From pre-quantum to post-quantum iot security: A survey on quantum-resistant cryptosystems for the Internet of Things.IEEE Internet of Things Journal, 7(7): 6457-6480.

[21]. Vermesan, O., et al. (2011). Internet of things strategic research roadmap. Internet of things-global technological and societal trends, 1(2011): 9-52.

[22]. Ning, H. and Z.J.I.C.L. Wang, (2011), Future internet of things architecture: like mankind neural system or social organization framework. IEEE Communications Letters, 15(4): 461-463.

[23]. Weyrich, M. and C.J.I.S. Ebert. (2015). Reference architectures for the internet of things. IEEE Software, 33(1): 112-116.

[24]. Dazine, J., A. Maizate, and L. Hassouni. (2018). Internet of things security. In 2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD). IEEE.

[25]. Zhao, K. And L. Ge. (2013). A survey on the internet of things security. In 2013 Ninth international conference on computational intelligence and security. IEEE.

[26]. Reda-Elbarkouky Wageda I. El-Sobky, Ahmed A. Abdel-Hafez, Comparative Study of Algebraic Attacks. International Advanced Research Journal in Science, Engineering and Technology, 3(5): 85-90.

[27]. Paar, C. and J. Pelzl. (2009). Understanding cryptography: a textbook for students and practitioners, Springer Science & Business Media.

[28]. Prakash, D.J.J.o.t.G.R.S. (2019).Introduction to cryptography and network security.Journal of the Gujarat Research Society, 21(17): 417-423.

[29]. Wang, D. And S.-L. Sun. (2008). Replacement and Structure of S-boxes in Rijndael. In 2008 International Conference on Computer Science and Software Engineering. IEEE.

[30]. Kazlauskas, K. and J.J.I. Kazlauskas. (2009). Key-dependent S-box generation in AES block cipher system.Informatica, 20(1): 23-34.

[31]. Tesař, P.J.R. (2010). A new method for generating high non-linearity s-boxes. Radioengineering, 19(1): 23-26.

[32]. Cui, J., et al. (2011). An improved AES S-box and its performance analysis. International Journal of Innovative Computing, Information and Control, 7(5): 2291-2302.

[33]. Peng, J., et al. (2012). Construction and analysis of dynamic S-boxes based on spatiotemporal chaos. In 2012 IEEE 11th International Conference on Cognitive Informatics and Cognitive Computing. IEEE.

[34]. Das, S., J.U. Zaman, and R.J.P.T. Ghosh. (2013). Generation of AES S-Boxes with various modulus and additive constant polynomials and testing their randomization. Procedia Technology, 10: 957-962.

[35]. Waqas, U., et al. (2014). Generation of AES-like S-boxes by replacing affine matrix. In 2014 12th International Conference on Frontiers of Information Technology. IEEE.

[36]. Athena, J., V.J.C. Sumathy, and Systems, (2017), Survey on public key cryptography scheme for securing data in cloud computing. Circuits and Systems, 8(3): 77-92.

[37]. Sklavos, N. (2014). Book Review: Stallings, W. Cryptography and Network Security: Principles and Practice. Information Security Journal: A Global Perspective. 23(1-2): 49-50.

[38]. Hellman, M.E.J.I.C.M. (2002). An overview of public key cryptography. IEEE Communications Magazine, 40(5): 42-49.

[39]. Rabin, Michael O. (1979). Digitalized Signatures and public key functions as intractable as intractable as factorization. Massachusetts Inst of Tech Cambridge Lab for Computer Science.

[40]. Pornin, Thomas. (2013). Deterministic usage of the digital signature algorithm (DSA) and elliptic curve digital signature algorithm (ECDSA).Internet Engineering Task Force RFC, 6979:1-79.

[41]. Poulakis, D. And R. Rolland. (2015). A Digital Signature Scheme based on two hard problems, springer, in Computation, Cryptography, and Network Security. 441-450.

[42]. Singh, Vikram. (2015). A Practical Key Exchange for the Internet using Lattice Cryptography. IACR Cryptol. ePrint Arch.2015: 138.

[43]. Hoffstein, Jeffrey, Jill Pipher, and Joseph H. Silverman. (1998). NTRU: A ring-based public key cryptosystem. In International Algorithmic Number Theory Symposium. Springer.

[44]. Hoffstein, J., et al. (2008). An introduction to mathematical cryptography, Springer. Vol. 1.

[45]. Nevins, M., et al. (2010). NTRU over rings beyond Z, Designs, Codes and Cryptography, 56(1): 65-78.

[46]. Jarvis, K., M.J.D. Nevins, (2015). ETRU: NTRU over the Eisenstein integers. Designs, Codes and Cryptography, 74(1):219-242.

[47]. Jun, Yao, and Zeng Guihua. (2006). Enhanced NTRU cryptosystem eliminating decryption failures. Journal of Systems Engineering and Electronics, 17(4): 890-895.

[48]. Bu, S. And H. Zhang. (2009). Research on the Method of Choosing Parameters for NTRU. In 2009 International Conference on Multimedia Information Networking and Security. IEEE.

[49]. Shuai, L., et al., (2019), A Group-Based NTRU-Like Public-Key Cryptosystem for iot. IEEE Access, 7: 75732-75740.

[50]. Sever, Mehmet, and Ahmet Şükrü Özdemir. (2020). A new offer of NTRU cryptosystem with two new key pairs.Numerical Methods for Partial Differential Equations.

[51]. Sharma, S., J.S. Yadav, and P.J.i.J. Sharma. (2012). Modified RSA public key cryptosystem using short range natural number algorithm. International Journal, 2(8).

[52]. Thakor, Vishal A., M. A. Razzaque, and Muhammad RA Khandaker. (2020). Lightweight Cryptography for iot: A State-of-the-Art.arXiv preprint arXiv:2006.13813.

[53]. Buchanan, William J., Shancang Li, and Rameez Asif. (2017). Lightweight cryptography methods. Journal of Cyber Security Technology, 1(3-4): 187-201.

[54]. Pry, J.C. and R.K. Lomotey. (2016). Energy consumption cost analysis of mobile data encryption and decryption. In 2016 IEEE International Conference on Mobile Services (MS). IEEE.

[55]. Abood, O.G., M.A. Elsadd, and S.K. Guirguis. (2017). Investigation of cryptography algorithms used for security and privacy protection in smart grid. In 2017 Nineteenth International Middle East Power Systems Conference (MEPCON). IEEE.

[56]. Panda, M. (2016). Performance analysis of encryption algorithms for security. In 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES). IEEE.

[57]. Vyakaranal, S. And S. Kengond. (2018). Performance analysis of symmetric key cryptographic algorithms. In 2018 International Conference on Communication and Signal Processing (ICCSP). IEEE.

[58]. Al Sibahee, M.A., et al. (2017). The Best Performance Evaluation of Encryption Algorithms to Reduce Power Consumption in WSN. In 2017 International Conference on Computing Intelligence and Information System (CIIS). IEEE.

[59]. Soewito, B., F.E. Gunawan, and A. Antonyová. (2016). Power consumption for security on mobile devices. In 2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS). IEEE.

[60]. Ochôa, I.S., et al. (2018). Data Transmission Performance Analysis with Smart Grid Protocol and Cryptography Algorithms. In 2018 13th IEEE International Conference on Industry Applications (INDUSCON). IEEE.

[61]. Alrowaithy, M. and N. Thomas. (2019). Investigating the Performance of C and C++ Cryptographic Libraries. In Proceedings of the 12th EAI International Conference on Performance Evaluation Methodologies and Tools.

[62]. Surendran, S., A. Nassef, and B.D. Beheshti. (2018). A survey of cryptographic algorithms for iot devices. In 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT). IEEE.

[63]. Harbi, Y., et al. (2018). Secure data transmission scheme based on elliptic curve cryptography for internet of things. In International Symposium on Modelling and Implementation of Complex Systems. Springer.

[64]. Sharma, C. (2018). Performance Analysis of ECC and RSA for Securing coap-Based Remote Health Monitoring System. Springer, in Ambient Communications and Computer Systems. 615-628.