A Review of Encryption Techniques in IoT Devices

Ahmet Furkan Aydogan *axa184@shsu.edu*
Computer Science
Sam Houston State University
Huntsville, 77340, United States

Cihan Varol *cvarol@shsu.edu*
Computer Science
Sam Houston State University
Huntsville, 77340, United States

Amar Rasheed *axr249@shsu.edu*
Computer Science
Sam Houston State University
Huntsville, 77340, United States

Narasimha Karpoor Shashidhar *karpoor@shsu.edu*
Computer Science
Sam Houston State University
Huntsville, 77340, United States

Abstract

IoT devices are now frequently used in living spaces, education systems, military, police surveillance mechanisms and critical government systems. At the same time, cyberattacks against IoT devices are on the rise. The main element of protecting IoT security is encryption methods. However, it is difficult to say that each of the encryption methods with dozens of different approaches can provide security for IoT devices. This study examines symmetric, asymmetric, hybrid, lightweight, Authenticated Encryption with Associated Data (AEAD) and post-quantum encryptions, which are among the encryption methods used to ensure IoT security. In addition, the study has a wide examination of the differences, advantages, disadvantages, complexity and costs between the mentioned methods. Finally, the study conveys the results of the examined encryption methods against popular attack methods. Our study reveals that while a small portion of current IoT encryption methods uses asymmetric or symmetric encryption methods, hybrid and lightweight encryption techniques make up most of the remaining work. Although lightweight methods have been getting popular in the IoT field, it does not create a balance between cost and security, unlike hybrid encryption methods.

Keywords: Symmetric Encryption, Asymmetric Encryption, Hybrid Encryption, Lightweight Encryption, Authenticated Encryption with Associated Data (AEAD),Post-quantum encryption, IoT Security, IoT Encryption.

1. INTRODUCTION

IoT has been defined as objects capable of internet connection to send and receive data from the United States Federal Trade Commission (Federal Trade Commission, 2015). Arguably, IoT devices have started to take place in daily life faster than innovations such as television, telephone, or the internet (Crane, 2021). IoT devices, which had a market share of 190 billion dollars in 2018, are expected to increase to 1.11 trillion dollars in 2026 while increasing the current market share by a 24.7% compound annual growth rate (CAGR) (Insights, 2019). According to Gartner's research, the number of IoT devices, which was 14.2 billion in 2019, will reach 25 billion in 2025 (Omale, 2018). Another study reflects that 152,200 IoT devices will be activated every minute in 2025 (Rosen, 2019).

Along with the usage rates, the security of IoT is also becoming an emerging issue. In particular, many data and privacy theft incidents have been encountered by exploiting unconscious users' vulnerabilities. Home appliances generally use IoT devices, which can open the door of our house to digital pirates. Hackers penetrating company-controlled IoT devices can yield catastrophic conclusions. For instance, at the end of 2016, a botnet attack called Mirai affected the whole world. Thousands of IoT devices were infected to conduct Distributed Danial of Service (DDOS) attacks on widely known websites such as Twitter, GitHub, and Netflix by exploiting IoT devices' default usernames and passwords (Fruhlinger, 2018).In 2017, the German Federal Network Agency decided to disintegrate all products on the charge of espionage on a doll called "My Friend Cayla," which can communicate with children through an IoT device. The mentioned application is also described in the Telecommunications Act. Under section 90 (Kinast, 2017). In 2018, the cybersecurity teams examining TVs revealed a problem with the background running applications that sent confidential information to companies (Graham, 2018). Another incident happened to Laura Lyons and her family in Chicago in 2019. Hackers controlled the IoT camera that they recently acquired. The hackers used the IoT device to report that three different ballistic nuclear missiles fired from North Korea were approaching (Hollister, 2019).

According to the NetScout report, future attacks will continue with great extent and impact. The NetScout claims that IoT devices will become the target of hacker attacks every five minutes (Netscout, 2019). The research conducted by Symantec Corporation has reached statistical data with honeypots imitating IoT devices. During the investigation, 5,200 attacks against imitation IoT devices were detected. According to the data obtained, most attacks (75%) were against IoT security cameras (Davis, 2019). Cyber Threat reports prepared by Avast shared that those attacks on IoT devices worldwide increased by 217.5% in 2019 compared to the previous year. Another striking data revealed by the research is that botnet attacks targeting IoT devices occur mainly against IoT device users in the United States. In other words, users in the USA were the main target in all 1794 attacks, with 46% of 3900 attacks directed to IoT devices. The Avast report also indicated that 40.8% of smart home systems are under attack (Avast, 2019a)(Avast, 2019b). These attacks can be life-threatening if they are aimed at Medical IoT devices, destroying patient records, and losing confidentiality (Gatlan et. al., 2019).

All of the mentioned security issues prove that precautions should be taken to improve IoT devices. Today, many companies are working on IoT security. In 2019, the IoT security budget increased by 28% compared to the previous year and reached 1.5 billion dollars. In addition, 373 million dollars for endpoint security, 186 million dollars for gateway security, and 946 million dollars for professional services were reported. In the coming years, it is expected that the budget allocated for industry 4.0 will reach much larger sizes. It is stated that the mentioned expenditure amounts may increase to higher levels in the coming years. (Bamiduro, 2018).

Alrawi et al. provided a detailed review of IoT security (2019). The article contains a great deal of information about examining the attacks on IoT devices and explains a version of the potentially affected devices. Author shares attack vectors such as Vulnerable Services, Weak Authentication, Default Configuration, Permissions, Programming, Data Protection, Encryption, MITM, Mitigation, and Stakeholders. The attack vectors’ striking feature is that the attackers are not very skilled in mitigations and framework attacks. Encryption, weak authentication for the cloud endpoint portion, and data protection for mobile devices pose a significant challenge (Alrawi et. al., 2019).

Farooq et al. (2015) examined the attacks by exploring the general architecture layers of IoT devices. IoT devices' general architecture consists of four layers: Perception Layer, Network Layer, Middle-ware Layer, and Application Layer. The perception layer is the part related to the sensors in IoT devices. It includes elements such as RFID or barcodes and provides contact with the outside world. The Network Layer allows the IoT to receive or send data via a network. The network configuration in IoT devices can be internet, local, or mobile data. The Middle-ware layer processes data for IoT and links processed data to databases. The application layer acts as a bridge between the user and the IoT device. That is why IoT can be easily used in smart home systems or industrial areas with the help of the user interface. Security problems accompany these layers. Attacks to the perception layer can include unauthorized tag access, tag cloning, eavesdropping, spoofing, and RF jamming. The network layer attacks are typically Sybil, sinkhole, sleep deprivation, denial of service (DoS), malicious code injection, and man-in-the-middle. The Middle-ware layer, which has an important place for IoT devices, is tried to be defeated by unauthorized access, DoS, and the malicious insider. Finally, the article describes attacks against the application layer. In particular, techniques such as malicious code injection, denial-of-service (DoS), spear-phishing, and sniffing attacks can be used to attack the application layer (Farooq et. al., 2015)

Open Web Application Security Project (OWASP) have reported the stages that threaten IoT security. According to the OWASP report, the least likely security problems were a lack of physical hardening. It mainly describes the presence and abuse of IoT devices because of easy-to-access points, such as security problems caused by using the existing settings of IoT devices and insufficient privacy protection (Owasp, 2018). As indicated previously, security is a primary concern for IoT devices. That is why there is a dire need to examine current encryption methods in the field. Therefore, in this work, we categorize encryption methods as asymmetric, symmetric, hybrid, and lightweight and provide an up-to-date literature review.

2. LITERATURE REVIEW

2.1 Lightweight Encryption Methods for IoT Devices

Lightweight encryption, which has gained popularity in recent years, provides excellent advantages, especially for IoT devices. Employing existing encryption methods to expect lower power consumption of IoT devices has been a challenge that made the researchers look closely at Lightweight ways. The encryption algorithms developed as lightweight methods are generally evaluated in three categories: reliability, cost, and efficiency, as shown in Figure 1.

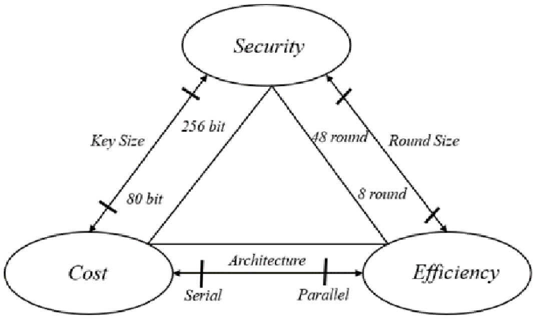

FIGURE 1: Interaction of reliability, performance, and price scheme (Khomlyak, 2017).

Developers are strictly obliged to keep all three features at the highest level. However, it is less likely to pull the highest standards in all three categories at the same time. For example, creating a safer Lightweight method on the IoT device will increase the cost since the algorithm contained in the encryption method must be complicated enough not to break in. However, it should be noted that Lightweight encryption methods are designed to provide a medium level of security and are highly employed in IoT devices (Journal, 2017). The Lightweight method is generally developed by eliminating the costly parts of existing encryption methods. Reducing power consumption is an essential concern for many IoT devices and can be called the most significant advantage of Lightweight algorithms. However, as a disadvantage feature of Lightweight algorithms, security issues are ignored when hardware encryption is performed. Thus, limiting the elements of an existing encryption method and integrating the restricted features into hardware components is a challenging and costly process (Munro, 2021).

Nowadays, cloud technology and IoT are intertwined. Cloud technology has many advantages for IoT devices. One of the advantages of cloud technology is definitely in the security component of IoT devices. Lightweight encryption methods are also used to improve the security levels of cloud systems. For example, Belguith et al. (2018) used a Lightweight encryption method to ensure the security of IoT devices. The main algorithm uses the attribute-based encryption (ABE) method with a policy update (PU-ABE), eliminating the steps required for attribute-based encryption using cloud technology, resulting in lower production costs. PU-ABE has three advantages: data can be easily re-encrypted with cloud technology, public keys can be hidden or shared with desired accounts, and data shared with end-users is a fixed size and low cost (Belguith et. al., 2018).

Muhammad et al. (2019) provide a new method for IoT security intertwined with cloud technology. Authors modified the ciphertext policy-attribute-based encryption (CP-ABE) method to ensure IoT devices' safety and established a Lightweight encryption method. Besides, it is mentioned that the decryption and ciphertext size of the CP-ABE process increases linearly. A modification is made because of the large size of the decoding data obtained when the number of attributes increases. To better integrate the CP-ABE method into IoT devices, the ciphertext part of the encryption method is defined as a constant and protected by cloud technology (Muhammad et. al., 2019).

Lightweight encryption algorithms can include chaotic systems. A chaotic system creates unique data sets. The unique groups are combined with encryption methods, enabling more complex results to strengthen the encryption method (Mohananthini et. al., 2020). For example, Naif et al. (2019) used an advanced encryption standard algorithm powered by the chaos method to secure IoT devices. The authors then transformed the resulting encryption algorithm into a Lightweight state to make it more effective. In the process, hashing algorithms called SHAKE and HMAC were used. First, a 5-dimensional chaos text generator was separated into 4 x 4 bits. The encrypted text generated by Advanced Encryption Standard (AES) is expanded by the specified 4x4 bits and called MLAES. Finally, 128-bit SHAKE and 256-bit HMAC algorithms were used in the hash comparison to check the data obtained. In the experiments, the proposed algorithm, compared to the AES algorithm, has an advantage in the CPU cycle (Naif et. al., 2019).

Another instance of a Lightweight algorithm with a chaotic method for ensuring IoT device security is conducted by Kumar et al. (2016). The authors used the 128-bit Lightweight encryption method to secure IoT devices. The authors stated that symmetric encryption methods enacted them. The first step produced a 128-bit ciphertext by the chaotic dynamic system. In the second step, the 128-bit ciphertext was divided into 8-bit portions and shuffled 1000 times. While the encryption was performed, the 8-bit blocks were multiplied by the mode 256 value and XORed. Logistic, Lozi, and intertwining logistic maps played a vital role in this work to create more complexity. As a result, the work protects against brute-force attacks (Kumar et. al., 2016).

Apart from Lightweight algorithms using cloud technology or chaotic systems, encryption methods solely focused on IoT security are also available. For instance, Katagi et al. (2008) evaluated encryption algorithms employed for IoT security. The study, which chooses symmetric encryption methods as a research area, contains information about Block Ciphers, Stream Ciphers, and Hash Functions. To reduce the cost of the AES encryption method, the authors stated that the EAS method was modified, the CLEFIA and PRESENT algorithms were created, and then applied the CLEFIA and PRESENT algorithms to the IoT devices under the Lightweight category. According to the article, when CLEFIA, AES, and CAMELLIA algorithms are used in IoT devices, the CLEFIA algorithm provides a significant advantage in the IoT device's gate efficiency headers. In the hardware performance schemes, the PRESENT algorithm, another Lightweight encryption method, gets ahead of the AES algorithm in terms of less energy consumption (Katagi et al., 2008).

Chaudhry (2018) measured the validity of the data to be transferred to IoT devices. The authors proposed a framework that can benefit from the 7-bit repository. The data to be transmitted is first processed with master encryption to be selected from a large encryption pool. Next, another 7-bit specific header is added to the encrypted data. In the decryption phase, after the decryption with the predetermined private key, the 7-bit portion with the ID feature is searched for matching in the bit pool. If both data are compared, the data is sent to the IoT device (Chaudhry, 2018).

Another research conducted by Saha et al. provides another method to provide IoT device security. Cipher Block Chaining (CBC) was added as an additional security provider in addition to White-Box Cryptography (WBC). The WBC method aims to hide encrypted data into randomly generated data blocks. The authors tried to prevent code removal and differential attacks with the mentioned method in the WBC method. The Cipher Block Chaining method is based on encrypting previously sent pure data and new data with the XOR method. WBC and CBC methods provide high data security when used together. In the final analysis, attacks directed at the WBC method, such as entropy, essential whitening, and code lifting, were repulsed (Saha et. al., 2019).

Gunathilake et al. (2019) combined 5G technology and Lightweight algorithms as an example of the methods to be used for IoT security in the future. The article focuses on IoT device security emerging with the spread of 5G technology. The report states that the lightweight encryption method would be suitable for 5G, especially considering the Random-Access Memory (RAM) capacity in IoT devices. The article, which includes technical analysis, proposes a careful examination of the Central Processing Unit (CPU) cycles and the capacities of RAM and Read-Only Memory (ROM). The authors suggest that the Lightweight algorithms' mathematical properties should be experimentally observed with hardware systems (Gunathilake et. al., 2019).

2.2 Authenticated Encryption with Associated Data Methods for IoT Devices

Authenticated Encryption with Associated Data (AEAD) is the name given to systems that guarantee the authenticity of inputs for data protection. AEAD is a variation of the Authenticated Encryption (AE) system. While AE focuses on the accuracy of encryption, AEAD focuses on the integrity of data and encryption. In summary, AEAD is responsible for controlling both encrypted and unencrypted data (Bellare et. al., 2003). AEAD mainly integrates with another system defined as a message authentication code (MAC). There are small bits of information in the MAC to control the entries. In addition, MAC is used to maintain the authenticity and stability of the data. There are three different approaches followed by systems working integrated with MAC. In the Encrypt-then-MAC (EtM) method, the data is integrated with the MAC after it is encrypted. This method can be described as the safest. In the Encrypt-and-MAC (E and M) method, MAC data is generated with data from plain text. The generated MAC and encrypted data are sent together. It is generally used in Secure Shell (SSH) connections. In the MAC-then-Encrypt (MtE) method, although the MAC is still generated from the data, both the MAC and the actual data are encrypted together. This system is used in Transport Layer Security (SSL) technology (Bellareet. al., 2004).

Grain stream cipher, published by Hell et al. (2007), emerged for devices with limited hardware capabilities. Grain has a key length of 80 bits. However, due to security vulnerabilities, a new version, Grain-128a, was released in 2011. Grain-128a has performed a more comprehensive study than its previous version using symmetric key encryption and MAC approaches. Also, Grain-128a needs additional hardware to keep security at a high level. In the encryption method, MAC and key consist of 32 bits. Also, there is a total of 256 bits of internal state size pre-output function (Agren et. al., 2011). The grain-128AEAD algorithm developed by Hell et al. (2019) is a new version of Grain-128a. Grain-128AEAD has a 128-bit key and a 96-bit nonce field. Also, the main difference between Grain-128a and Grain-128AEAD is that the new version complies with the NIST Lightweight Cryptography Standardization Process. So, Grain-128AEAD is a lightweight encryption method. The Grain-128AEAD MAC size has been increased to 64 bits. Grain-128AEAD also has an additional option that concentrates on encryption only, eliminating the authentication feature. However, no matter how much hardware power consumption there is, authentication of data, which is one of the working principles of the AEAD system, should always be activated. Unlike its older versions, Grain-128AEAD has set the keystream length as 2^{80} bits. This way, it focuses on eliminating linear approximation attacks. Released as a later update, Grain-128AEADv2 has clocked 320 times to initialization. This is 64 times more than the previous version (Hell et. al., 2019).

The sponge function or construction in cryptography gives bit outputs of the desired length by changing the bit stream as input. The sponge function is used in authentication, random number generation, and stream ciphers (Bertoni et. al., 2012). The PHOTON family of hash functions is introduced in the study by Guo et al. (2011). The PHOTON hash output size ranges from 64 to 256 bits. It can have five different variations in total. These are PHOTON-80, PHOTON-128, PHOTON-160, PHOTON-224, and PHOTON-256. In addition, all PHOTON variants are integrated with MAC. The PHOTON-Beetle AEAD method proposed by Bao et al. (2019) works with PHOTON-256 and the Sponge function. The proposed method is among the NIST lightweight encryption methods. It is advantageous for IoT devices as it creates a small hardware footprint (Bao et. al., 2019).

The Sparkle by Beierle et al. (2020) is among the NIST lightweight symmetric cryptographic algorithms. The Sparkle is a family of cryptographic permutations. Sparkle can have block sizes of 256,384 or 512 bits. However, Sparkle can provide a security level between 120 and 250 bits. EAAD methods developed with Sparkle work in integration with MAC. For example, the encryption method called Schwaemm was created by the AEAD and Sparkle permutation. Providing low-cost processing capacity for IoT devices, Sparkle keeps IoT devices' ROM and RAM footprints in tiny proportions (Beierle et. al., 2020). Variations and bit values of Schwaemm algorithms are given in table 1.

Algorithm Name	Security Size in Bits	Permutations Size in Bits
Schwaemm-128-128	120	256
Schwaemm-256-128	120	384
Schwaemm-192-192	184	384
Schwaemm-256-256	246	512

TABLE 1: Security and permutation values of Schwaemmalgorithms in bits (Beierle et. al., 2020).

The Authenticated Encryption with Associated Data (AEAD) methods mentioned above are not included in the summary tables because their evaluation of NIST standards is ongoing.

2.3 Symmetric and Asymmetric Encryption Methods to Ensure Security of IoT Devices

Symmetric and asymmetric encryption methods are well-known protectors of IoT device security. In the symmetric encryption method, it is possible to encrypt and decrypt data using only one key. Users who do not have the key will see the data in encrypted mode. However, the user who has the key can decrypt the data. The secret key is usually a sequence of randomly generated letters or numbers. Most of the time, symmetric encryption's private keys use specific standards, such as FIPS 140-2. Two different methods can be followed when creating symmetric encryption. The first method is called Block Algorithms. In this method, data with a certain bit length begins to be encrypted. Previous data is stored in memory to transfer all data. The second method, Stream Algorithms, holds the algorithm's data instead of keeping it in memory. AES (Advanced Encryption Standard), DES (Data Encryption Standard), IDEA (International Data Encryption Algorithm), Blowfish (Drop-in replacement for DES or IDEA), and RC4 (Rivest Cipher 4) are types of symmetric encryption methods. While AES, DES, IDEA, and Blowfish algorithms work with the above-mentioned Block Algorithm method, the RC4 is one of the stream algorithms approaches. Symmetric encryption is fast and convenient. Many users prefer it because CPU consumption is not very high. Symmetric encryption is often used to encrypt large data sets. It is also widely used in security and hashing for payments. However, symmetric encryption presents several problems. The so-called key exhaustion problem can create vulnerabilities due to the abuse of old secret keys. In the attribution data problem, symmetric encryption keys do not contain metadata features such as expiration dates or access control lists, so this encryption can be abused over time (Turner, 2018).

Asymmetric encryption has two different keys, referred to as public and private. Anyone can use the public key to send data. However, the data sent can only be decoded by the private key. Besides, the public key's contents can be accessed by the private key, while the reverse is not possible. The CPU usage in asymmetric encryption is high, but the security portion is vital. DH (Diffie-Hellman), RSA (Rivest-Shamir-Adleman), Elgamal, and Merkle-Hellman are examples of asymmetric encryption algorithms. DH is considered the ancestor of asymmetric encryption. The most significant advantage is ensuring the secret key is generated safely and quickly. It provides critical exchanges between the parties by asymmetric encryption. RSA is an asymmetric encryption method that commercial organizations highly preferred. RSA is powerful because it generates large prime numbers and uses them in encryption. It performs encryption and decryption processes slowly. Therefore, internet bandwidth must be high for security. The Elgamal algorithm uses the fundamental exchange model of DF encryption. Elgamal creates the encryption algorithm with the data generated by a discrete logarithm. The Merkle-Hellman algorithm works unidirectional; the only point it separates from RSA. In the Merkle-Hellman method, the data is encrypted with the public key and decrypted only by the private key (Binance, 2018).

One example of using a symmetric encryption method to ensure IoT devices' security was conducted by Fischer et al. (2019). The article contains the steps to modify the CP-ABE plan to secure IoT devices. During the development phase of the new process, the key revocation problem in the ABE method was taken as the primary objective. Key revocation consists of revoking the privileges of users who have public keys. According to the article, the ABE method allocates too much time to the key revocation part, which increases production costs. The method presented as a solution is to remove the ABE method's key revocation and process it in a different section called proxy. ABE will see the key revocation values as a constant with the specified method before processing the text to be encrypted. At the same time, key revocation values will be created in the Proxy section. Besides, the values in the proxy content will not be enough to decrypt. The methodology performed better than the total ABE conversion and encryption phases (Fischer et. al., 2019).

Hussain et al. (2018) created a new encryption method inspired by asymmetric encryption to secure IoT devices' communication. The generated encryption method aims to obtain security by processing the user's data, along with a unique key code received from the user. The user can select a number between 0 and 255, then transfer the unencrypted data. Pure data received from the user is converted to ASCII codes. The cipher from 0 to 255 previously supplied by the user is inserted into the XOR process with plaintext and converted to ASCII. Then, the results are converted to decimal values using the n-bit and n/2-bit sequence path. The encryption becomes decrypted when both data are retrieved from the user (Hussain et. al., 2018).

Makwana (2017) used the homomorphic encryption (HE) method to ensure IoT devices' security using the cloud system. Homomorphic cryptography can be transmitted as the method where the ciphertext content is expected to have the same results when processed with the same mathematical content as the plaintext's value. In short, before a transaction is performed, the encrypted text will provide the same value as the pure text so that data confidentiality is ensured and decryption time is not wasted for the desired transaction. The public and private keys were again processed with homomorphic encryption and transferred to the cloud. In the last stage, before the requested data from the cloud system is sent to IoT devices, the data received from the cloud system by mobile devices or Raspberry Pi is transferred to the IoT device through mathematical operations and by extracting the command data (Makwana, 2017).

2.4 Hybrid Encryption Methods to Ensure Security of IoT Devices

It is undeniable that symmetric and asymmetric encryption methods are in a disadvantageous position in terms of power consumption. As seen in Figure 1, it was necessary to reduce the power consumption of the bit length used by encryption methods. Hybrid encryption combines the strengths of symmetric and asymmetric encryptions. However, in general, the bit lengths used by the encryption have been reduced. Since IoT devices are sensitive to power consumption, hybrid encryption methods can be a reasonable choice. The changes made in hybrid encryption mainly arise from the arrangement of public and secret keys (Denis, 2007).

Henriques et al. (2017) combined symmetric and asymmetric encryption to provide a more secure IoT environment. In the asymmetric encryption part, the RSA method and timestamps are used. The vigenere encryption method was modified and processed in the symmetric encryption section. The vigenere encryption method is an improved version of the caesar encryption method, which is considered primitive. The caesar encryption method is mainly based on converting data into an alphabet in a specific way. In vigenere encryption, several different alphabets are used at the same time. Specifically, a key is selected by the user and is processed on the alphabet used. In the article, data from the sensor is encrypted by the timestamp and modified vigenere encryption method. Then, the public key that the user will use for the IoT device is encrypted with RSA, followed by combining both encrypted data to make the security of the IoT's server-side. The critical point is that the timestamp has eliminated the relationship between the two encryption methods. By using a timestamp, encrypted data is provided to be unique. Otherwise, the data encrypted in the same time frame would yield the same results (Henriques et al., 2017).

Jian et al. (2019) aimed to provide security on IoT devices by hybridizing asymmetric and symmetric encryption methods. During data transfer between IoT and other devices, it is first suggested that the IoT device should control the network to coordinate which and how many devices are connected. In the first step, called the self-identification procedure, the MAC addresses of each device connected to the network are recorded by IoT modules. In short, each device connected to the network shares its MAC addresses for later use in encryption and storing valuable data such as manufacturer information. Then those devices will register with the database. In the second step, public keys can be shared between the predefined devices. When transferring the data, IP addresses and MAC addresses are controlled via the server, and if the match is achieved, the data process is performed. Finally, 751 bits of a private key and 498 bits of a public key were obtained. This helped to achieve high data security, since the private key size starts with 256 bytes in asymmetric encryption (Jian et al., 2019).

In another example, the authors aimed to secure IoT devices in a hybrid way using RSA, AES, and e-mail verification methods. The changes on the Field Programmable Gate Array (FPGA) of the devices were applied using Xilinx SPARTAN-6, and the results were analyzed with the synthesis tool of Xilinx ISE-Design 14.5 software. Finally, methods such as brute force attacks on IoT devices have been experimented with using cloud technology, pointing out vulnerabilities (Chandu et. al., 2017).

A modified RSA encryption method is developed in another work to ensure IoT security by Thirumalai et al. (2017). Unlike the original RSA, the selected large prime numbers were assigned to X and Y values using Pell's coordinates. Then, the tuples' data belonging to the RSA method are used to calculate Pell's parameters. From the Pell parameters' output values, X is used as a public key, while Y outputs are used as a private key. The article has improved IoT device security using the Diophantine equation method. The authors showed a cost profit of 24% over the traditional RSA encryption (Thirumalai et al.,2017).

Yousefi et al. (2017) provided a hybrid method referred to as HAN. The so-called HAN method is based on asymmetric encryption and targets low computation. Also, the AES and NTRU encryption algorithms are used in the HAN algorithm. Each data that will send a command to IoT has a digital signature. First, a public key is created to solve the private key produced for the security of the digital signature, and AES first encrypts the key. In the second step, the AES, containing the public key, is encrypted by the NTRU. The NTRU algorithm encrypts the private key sent to the receiver to resolve the public key. Before AES's decryption step, the NTRU algorithm is applied for public and private keys. The HAN algorithm has a faster throughput time than AES and RSA encryption methods (Yousefi et al., 2017).

Ibrahim et al. (2019) presented a hybrid encryption method to protect data privacy on IoT devices. 64-bit data requested from the user was transformed into more secure and costless data with the help of KHAZAD and Feistel encryption methods. Specifically, KHAZAD and Feistel algorithms are based on substitution permutation networks (SPN). The displacement properties of the received data are used in both algorithms. The similar properties of both algorithms provide an advantage in terms of hybridization. Also, KHAZAD encryption was modified in the article to a 16-bit format. First, the 64-bit data received from the user is divided into 4-bit segments. Secondly, the modified KHAZAD algorithm and the data divided into 4-bit segments are encrypted with the XOR gate, a specification of the KHAZAD algorithm. Secondly, the modified KHAZAD algorithm and the data divided into 4-bit segments were in the process with linear and non-linear transformations. Following the above transformation, the data were taken into 4x4 and four different matrices called K (1, 2, 3, and 4). After all these procedures, K (1) and K (2) matrices were separated into one group, and K (3) and K (4) matrices were separated into another group by using the Feistel method. In the fourth step, the groups were made more secure by using the XOR gate between them first and then between the two groups themselves. A high rate of the execution time was obtained compared to the DES algorithm (Ibrahim et al., 2019).

In the article by Nagpal et al. (2019), the pure-form data was encrypted twice with the blowfish encryption method to ensure the security of IoT devices. The user was asked to enter two different data. The first data is as a string, and the second one is an image. After converting image and string data into binary form, both data were converted into 64x64. Then, the data was encrypted for the second time using the image encryption method, a feature of the Blowfish algorithm. In the last stage, the password that will allow the user to access the IoT device is processed with the blowfish algorithm, and both blowfish encryption is combined. The password was decoded by following a reverse path. This new encryption method yielded better entropy test results than the DES, 3-DES, and AES algorithms (Nagpal et. al., 2019).

Daddala et al. (2017) provided an IoT security solution by modifying the AES. Also, a more secure method was followed by creating authentication and secure communication layers. Specifically, layers formed for Man in the Middle (MITM) attacks have high success rates. Although data of various lengths are generated in the AES algorithm, the authors have chosen 128 bits. Following the 128-bit AES method, the 16x16 matrices called S-Box were made safer using multiplication and inversion methods. Separate ID keys have been created for the communication and authentication sections and are expected to match before data transfer (Daddala et. al., 2017).

In the article by Peshwe et al. (2017), the SIGMA-1 algorithm and Identity Based Encryption methods were used to achieve security in IoT devices. SIGMA-1 algorithm has been created and signed a certificate. Then, the signal strength calculation establishes a boundary between the server and the requesting device. With identity-based encryption, certificates have been tried to be made more secure. The proposed method has been shown to perform better with SIGMA-1 encryption. Besides, the article states that the SHA-3 method can make the certificates more confident with the hashing method (Peshwe et al., 2017).

Wei et al. (2017) used the identity-based encryption (IBE) based SM9, Chinese national cryptography standard, algorithm, and Elliptic Curve Diffie-Hellman (ECDH) methods as hybrids. The encryption method, called One-Time File Encryption Protocol (OTFEP), uses a timestamp measurement technique to ensure that attackers or email-based communication protocols can read data on IoT devices. First, the authors use the Curve Cryptography (ECC) feature of the SM9 algorithm, which generates a minor size key to provide pairing-based security. Then, the Elliptic Curve Diffie-Hellman (ECDH) method was used to control the session agreement stages (Wei et al., 2017).

2.5 Post Quantum Encryption Methods for Ensure Security of IoT Devices

There are two innovations that quantum mechanics brought to encryption methods. The first is Quantum Key Distribution (QKD), which tries to provide security using quantum mechanics. The second is quantum computers, which are created by using quantum mechanics and use an enormous processing advantage compared to today's computers to break existing encryption methods.

Unlike quantum key distribution, post-quantum encryption is the name given to encryption that is attacked but not broken by quantum computers. Post-quantum encryption methods generally focus on public key algorithms (Shor, 1997). Quantum computers that focus on solving multiplications of large-digit prime numbers that provide security in encryption have been successful (Shor, 1994). After the successful decryption processes performed by quantum computers, different methods were introduced, and security was aimed against quantum computers. Five different approaches are the core of post-quantum encryption methods: Lattice-based cryptography, Supersingular elliptic curve isogeny cryptography, Hash-based cryptography, Multivariate cryptography, and Code-based cryptography.

The concept of lattice in geometry and group theory is based on the fact that when two points on an absolute coordinate space are subject to mathematical processing, the results will give two different point groups (Zassenhaus, 2013). Lattice-based cryptography is based on the multiplicity of groups of two points and the repetition of mathematical operations. It revealed the first lattice-based cryptographic structure introduced by Ajtai (1996). The NTRU encryption algorithm introduced by Hoffstein et. al. (1998) turned out to be a successful method using the lattice-based structure. NTRU is based on the difficulty of possible combinations of points in a polynomial ring. The principles of NTRU on polynomials were moved to a different area with the Learning with errors (LWE) method. LWE is based on the difficulty of obtaining generated points on a finite ring. Ring learning with errors (RLWE) looks for points the size of the square root of the given points to process the LWE. The probabilities above are so large that approximately 49 million bits of key detection are required to be calculated by quantum computers (Lyubashevsky et. al., 2012).

Supersingular isogeny Diffie–Hellman key exchange (SIDH) is a method focused on points on finite fields with vertices. Relying on the mathematical power of calculating curves with finite vertices but not singular, SIDH used a 2688-bit public key, unlike methods such as RLWE and NTRU (Costello et. al., 2016). However, it was broken by a devastating key recovery technique, which was revealed in July 2022 (Castryck, 2022).

Multivariate cryptography is based on the difficulty of increasing the number of additional variables to a system of equations on a finite field and solving the resulting new multivariate equation. The Unbalanced Oil and Vinegar signature uses the multivariate quadratic equation to solve the difficulty of the asymmetric parts of the multivariate cipher based on polynomials (Garey et. al., 1979).

Hash functions are seemingly meaningless outputs when datasets of different lengths interact with datasets of fixed size. However, with the advent of quantum computers, hash functions exposed to high processing capacity became weak. The Lamport one-time signature scheme, which emerged as a precaution against the deactivation of hash functions, is based on one-time signing (OTS) data to ensure security. Lamport key pair includes the public and private keys. Since Alice needs to generate a 256-bit hash function, it should generate a total of 2x256 random number pairs. Since each generated number will be 256 bits, the final formulation will be 2x256x256. This value is 128 kbit in size and is called a privative key. The obtained private key is disposable because when another data is signed, the private key interacting with the public key can be decrypted (Lamport, 2018). However, the Merkle signature scheme improved the existing method and revealed that when the generated private key is incremented linearly, it can sign more than one data on tree topology. Namely, the Merkle signature scheme does not rehash each pair with 256 bits when generating the private key. Instead, it turned out that the data to be signed should be (n) x 256. In other words, the privative key will grow linearly for data to be signed more than once (Merkle, 1979). The study by Garcia (2005) showed that the Merkle signature scheme is safe when using one-way hash functions.

Error-correcting code (ECC) is a method introduced to fix problems in communication channels (Peterson et. al., 1972). When the redundancies added to the data to be sent reach the receiver, it can be revealed whether an intervention has occurred during the communication. Binary Goppa code emerged from the same strategy and was used to ensure data security (Berlekamp, 1973). The McEliece cryptosystem emerged as the first method to use randomization. The interaction of the data to be encrypted with the randomly generated asymmetric key naturally made it difficult to decrypt. When developing the private key, linear data derive the private key using a linear code. Encryption can be easily decrypted using binary Goppa codes added to the keys. However, decryption is considered secure when subjected to mathematical processing (McEliece, 1978).

Table 2 lists the advantages and disadvantages of post-quantum encryption approaches. Hash-based Cryptography has the advantage of being collision-resistant with others. Although Code-based Cryptography has been in use for many years compared to other approaches, it has large bit sizes.

Table 3 examines the encryption methods created with post-quantum encryption approaches. The table content shows the approaches used by the algorithms, key sizes, and signature sizes. While BLISS-II, GLP-Variant GLYPH Signature, SPHINCS, and SPHINCS+ algorithms contain signature features, other algorithms do not. The SPHINCS algorithm has the most significant bit size among the algorithms with the signature feature.

Approach	Advantages	Disadvantages
Code-based	Simple and fast implementation.	Elder. Large bit sizes.
Hash-based	Collision-resistant.	Large signature sizes, No key-exchange.
Isogeny-based	Based elliptic curves and bit sizes are small.	Operations are slow.
Lattice	Simple and fast implementation.	Key-exchange size is large.
Multivariate	Simple and fast implementation.	Operation bit sizes are large.

TABLE 2: Comparison of Post-Quantum Encryption Approaches.

Algorithm	Type	Private Key	Public Key	Signature
BLISS-II	Lattice	2 KB	7 KB	5 KB
GLP-Variant GLYPH Signature	Ring-LWE	0.4 KB	2 KB	1.8 KB
Goppa-based McEliece	Code-Based	11.5 KB	1 MB	Null
NewHope	Ring-LWE	2 KB	2 KB	Null
NTRU Encrypt	Lattice	842.875 B	766.25 B	Null
Quasi-cyclic MDPC-based McEliece	Code-Based	2,464 B	1,232 B	Null
Rainbow	Multivariate	95 KB	124 KB	Null
SIDH	Isogeny	48 B	564 B	Null
SIDH (Compressed Keys)	Isogeny	48 B	330 B	Null
SPHINCS	Hash-Signature	1 KB	1 KB	41 KB
SPHINCS+	Hash-Signature	64 B	32 B	8 KB
Streamlined NTRU Prime	Lattice	Null	154 B	Null

TABLE 3: Created Encryption Methods with Post-Quantum Approaches.

Post-quantum encryption methods are still under development. Existing approaches show that the processing capacity of the encryption methods to be obtained will have significant bit rates. Improvements are needed for post-quantum encryption methods that are reduced to levels that can be used in IoT devices. Due to the situation above, post-quantum encryption methods will not be included in the tables where all encryption methods will be compared.

3. COMPARISON OF IOT ENCRYPTION METHODS

As discussed previously, the encryption methods for IoT devices are categorized into four: asymmetric, symmetric, hybrid, and lightweight. Three different tables are prepared to quickly summarize the recent literature review on IoT device encryption methods. Table 4 categorizes all the discussed encryption methods. "✓" sign indicates the type of the implemented security solution. Note that since symmetric and asymmetric encryption are used in hybrid methods, these individual techniques are identified with the "•" sign.

Table 5 contains the complexity and cost ratios of the mentioned encryption methods. The "+" sign is used in the table to indicate the differences between the methods. Simultaneously, the method with the most cost and complexity is specified as "+++"; the lowest rated is marked with a single plus sign. As shown in Table 3, both Fischer et al. (2019) and Muhammad et al. (2019) used a method based on the development of the ABE encryption method. However, the cost of implementing Fischer's algorithm (2019) is higher than Muhammad et al. (2019) because it have equalized the text portion of the ABE method to a fixed level. Most of the time, the cost factor is linearly associated with the complexity level. However, while Ibrahim et al.'s (2019) solution is considered a complex algorithms, the cost factor is low. The reason for the mentioned situation is that Ibrahim et al. (2019) modified the KHAZAD and Feistel encryption methods. As a result, the authors improved the specified procedures, divided 16-bit data into 4-bit segments, and reduced processing costs (2019). Although Makwana et al.'s (2017) method is determined as high in cost, the complexity is moderate because the technique is based on applying the encryption (HE) algorithm twice. However, a vulnerability that may occur in the HE method will become readily available for the encrypted data. Also, please note that we cannot provide cost/complexity implications on Gunathilake et al.'s (2019) method, which is based on 5G technology and includes a theoretical explanation of a new encryption method that can be produced in the future. Table 6 uses the "Schema" header to determine which field the encryption methods work on the IoT device. Then, under the heading "Theory," it specifies the methods used or referenced by encryption methods. The "Summary" header briefly contains information about the operation of the encryption method.

Reference	Symmetric	Asymmetric	Hybrid	Lightweight
Belguith et al.	x	x	x	✓
Chandu et al.	•	•	✓	x
Chaudhry	x	x	x	✓
Daddala et al.	•	•	✓	x
Fischer et al.	✓	x	x	x
Gunathilake et al.	x	x	x	✓
Henriques et al.	•	•	✓	x
Hussain et al.	x	✓	x	x
Ibrahim et al.	•	•	✓	x
Jian et al.	•	•	✓	x
Katagi et al.	x	x	x	✓
Kumar et al.	x	x	x	✓
Makwana et al.	x	✓	x	x
Muhammad et al.	x	x	x	✓
Nagpal et al.	•	•	✓	x
Naif et al.	x	x	x	✓
Peshwe et al.	•	•	✓	x
Saha et al.	x	x	x	✓
Thirumala et al.	•	•	✓	x
Wei et al.	•	•	✓	x
Yousefi et al.	•	•	✓	x

TABLE 4: Categorical Distribution of Encryption Methods.

Reference	Cost	Complexity
Belguith et al.	+	+
Chandu et al.	++	++
Chaudhry	+	+
Daddala et al.	++	+++
Fischer et al.	++	+++
Gunathilake et al.	Null	Null
Henriques et al.	+++	+++
Hussain et al.	++	++
Ibrahim et al.	+	+++
Jian et al.	++	++
Katagi et al.	++	++
Kumar et al.	++	+
Makwana et al.	+++	++
Muhammad et al.	+	+
Nagpal et al.	+	++
Naif et al.	++	+
Peshwe et al.	+	++
Saha et al.	++	++
Thirumala et al.	++	++
Wei et al.	+++	+++
Yousefi et al.	++	++

TABLE 5: Cost and Complexity Distribution of Encryption Methods.

<table>
<tr><th>Reference</th><th>Scheme</th><th>Theory</th></tr>
<tr><th colspan="3">Summary</th></tr>
<tr><td>Belguith et al.</td><td>Provide IoT security with the Cloud.</td><td>Lightweight, ABE, PU-ABE.</td></tr>
<tr><td colspan="3"><b><i>The method modifies the ABE algorithm and acquires the performance criteria of PU-ABE. Finally, it provides password resolution using Cloud.</i></b></td></tr>
<tr><td>Chandu et al.</td><td>Provide security of applications of IoT devices.</td><td>Hybrid, RSA, AES.</td></tr>
<tr><td colspan="3"><b><i>As a result of Field Programmable Gate Array (FPGA) module modification, brute force attacks that may be exposed were tried to be prevented.</i></b></td></tr>
<tr><td>Chaudhry</td><td>Provide secure data sharing of IoT devices.</td><td>Lightweight.</td></tr>
<tr><td colspan="3"><b><i>Encryption was made by creating two different 7-bit values.</i></b></td></tr>
<tr><td>Daddala et al.</td><td>Provide secure data sharing of IoT devices.</td><td>Hybrid, AES.</td></tr>
<tr><td colspan="3"><b><i>Provided IoT security by modifying the AES. In addition, a more secure process was followed by creating authentication and secure communication layers. Against to MITM attacks.</i></b></td></tr>
<tr><td>Fischer et al.</td><td>Provide secure data transferring of IoT devices.</td><td>Symmetric, CP-ABE.</td></tr>
<tr><td colspan="3"><b><i>High-performance symmetric encryption with the CP-ABE method's cancellation of the key generation process.</i></b></td></tr>
<tr><td>Gunathilake et al.</td><td>Proposal Lightweight for IoT devices.</td><td>Lightweight, 5G.</td></tr>
<tr><td colspan="3"><b><i>How lightweight algorithms will gain importance with 5G technology in the future.</i></b></td></tr>
<tr><td>Henriques et al.</td><td>Provide secure data transferring of IoT devices.</td><td>Hybrid, RSA, and Vigenere.</td></tr>
<tr><td colspan="3"><b><i>Asymmetric and symmetric encryption methods are used together. For the asymmetric domain, the properties of the RSA method are used. For the symmetric domain, the properties of the Vigenere method are used.</i></b></td></tr>
<tr><td>Hussain et al.</td><td>Provide secure data transferring of IoT devices.</td><td>Asymmetric.</td></tr>
<tr><td colspan="3"><b><i>The generated encryption method aims to obtain security by processing the data received from the user along with a unique key code received from the user.</i></b></td></tr>
<tr><td>Ibrahim et al.</td><td>Provide secure data transferring of IoT devices.</td><td>Hybrid Encryption, KHAZAD, Feistel.</td></tr>
<tr><td colspan="3"><b><i>The 64-bit data requested from the user was converted into more secure and cost-effective data with the help of KHAZAD and Feistel encryption methods.</i></b></td></tr>
<tr><td>Jian et al.</td><td>Provide secure data transferring between IoT.</td><td>Hybrid.</td></tr>
<tr><td colspan="3"><b><i>An encryption method has been developed for secure data transfer between IoT devices. As a result, a secure IoT network has been achieved.</i></b></td></tr>
<tr><td>Katagi et al.</td><td>Provide secure data sharing of IoT devices.</td><td>Lightweight, CLEFIA, PRESENT, AES, CAMELLIA.</td></tr>
<tr><td colspan="3"><b><i>Comparison values of CLEFIA CAMELLIA and PRESENT algorithms derived from AES encryption.</i></b></td></tr>
<tr><td>Kumar et al.</td><td>Provide framework security of IoT devices.</td><td>Lightweight, Asymmetric, Chaotic Method.</td></tr>
<tr><td colspan="3"><b><i>First, 128-bit data entered asymmetric encryption. Secondly, it was strengthened with a chaotic system, and complexity was increased with the logistic map method.</i></b></td></tr>
</table>

Makwana et al.	Provide IoT security with the Cloud.	Asymmetric, HE.
The homomorphic encryption (HE) method is used to ensure the security of IoT devices using the cloud system.		
Muhammad et al.	Provide IoT securitywith the Cloud.	Lightweight, CP-ABE, Chaotic, SHAKE, HMAC, AES.
CP-ABE has been changed to avoid high costs, and encrypted data has been stored in the Cloud for more secure communication.		
Nagpal et al.	Provide secure data sharing of IoT devices.	Hybrid, Blowfish.
Provide IoT security with a technique that tries data in the pure form two times encrypted with blowfish encryption methods.		
Naif et al.	Provide secure data transferring of IoT devices.	Lightweight, Chaotic Method, SHAKE, HMAC, AES.
Data created using the chaos text generator is encrypted with AES. Also, it made data powered by hash algorithms called SHAKE and HMAC.		
Peshwe et al.	Provide secure data sharing of IoT devices.	Hybrid, IBE, SIGMA-1.
SIGMA-1 algorithm and Identity Based Encryption methods are used to achieve security in IoT devices.		
Saha et al.	Provide secure data transferring of IoT devices.	Lightweight, CBC, WBC.
Cipher Block Chaining (CBC) is added as an additional security provider in addition to the White-Box cryptography (WBC) method used to secure IoT devices.		
Thirumala et al.	Provide secure data sharing of IoT devices.	Hybrid, RSA, Diophantine Equation.
The primes are chosen using Pell parameters. Then, the Diophantine equation is used to strengthen the encryption method in terms of cost.		
Wei et al.	Provide secure data sharing of IoT devices.	Hybrid, IBE, ECDH, OTFEP, ECC, SM9.
The authors use the identity-based encryption (IBE) based SM9, the Chinese national cryptography standard, algorithm, and Elliptic Curve Diffie-Hellman (ECDH) methods as hybrids.		
Yousefi et al.	Provide secure data transferring of IoT devices.	Hybrid, HES, NTRU, HAN.
In practice, the digital signature is intended to be encrypted to secure data transfer.		

TABLE 6: Brief Summary of Encryption Methods for IoT Devices.

4. DISCUSSION AND CONCLUSION

IoT devices are among the first preferred devices in smart home systems, education, military, industry, and business. However, IoT devices, which have a large volume in terms of economic and usage rates, are increasingly exposed to cyber-attacks. Many companies with an important place in cybersecurity consistently reveal that attacks against IoT devices are increasing (Watters, 2022).

As a remarkable piece of information, even laws to improve the security levels of IoT devices support the development of encryption methods. According to the draft called the Internet of Things Cybersecurity Improvement Act of 2019, the most crucial stage for the security of IoT devices is related to manufacturers' ability to provide private encryption to users (Library of Congress, 2020). Another bill named information privacy: connected devices also mentions the importance of password security (California Legislative Information, 2018). Finally, California's IoT Security Law draft will ensure that each device user has a unique password to protect devices (Sysman, 2019).

IoT devices, where missing encryption methods, are regularly exposed to attacks, and significant economic damages occur (Vuldb, 2022); the D-Link DCS-5009L caused $100,000 in damage due to the lack of encryption methods. According to research conducted by Constantin, unencrypted data could easily be obtained in the Insteon Wink Hub 2 model smart home system device. The weakness did not expose itself for about 60 days, and many users were affected by the deficiency of Wink Hub 2 (Constantin, 2017). The security vulnerability in Belkin WeMo devices was likewise based on weak encryption methods. Breaking the fixed-encryption-protected data contained in the firmware in Belkin WeMo Motion, Belkin WeMo Link, Belkin WeMo Switch, and Belkin WeMo Crockpot caused significant economic damage (Constantin, 2016). Bose SoundTouch 10 devices had semi-protected encryption. The device was weakened due to the lack of encryption power and was subjected to remote hijacking (Brown, 2017). The LIFX Virtual Bulb has fallen into a situation that was hacked within hours of coming to the market. The device, which has shallow encryption capabilities, was exposed to brute force due to an unsafe transfer of hash values. The inevitably hacked device has put users in a difficult situation (Latest Hacking News, 2019). Logitech Harmony was subjected to a direct attack, which was rarely seen before.

Examining the device, experts found that the root password is entirely blank. It was not difficult to discover the mentioned problem by hackers (Williams, 2018). As a result of Philips HUE products using weak encryption, it was effortless to obtain the data transferred to servers or management panels. The data obtained included mac addresses and usernames (Lodge, 2020). In (Consumer Reports, 2018), information about Roku TVs revealed that there were encryption problems on the devices due to the research made by the users. During the transfer of users' privacy data, thousands of users have survived the risk of being compromised by hackers due to the lack of encryption. Although the common point of the attacks, as mentioned earlier, is the abuse of encryption methods, there are different methods to disable the encryption methods of IoT devices. Remote and software-based attacks on IoT devices are generally listed in four main categories: Privilege escalation, Eavesdropping, Brute-force password attacks, and DDoS (Nedbal, 2018). Privilege escalation may be examined more comprehensively than other methods because it is usually possible to fully capture the security parameters of the IoT device after disabling it. For example, IoT devices that are security neutralized after the buffer-overflow attack can be easily exposed to privilege escalation attacks. Chang showed the encryption systems as the part with the highest risk potential while transferring the privilege escalation attacks carried out with buffer overflow (Chang, 2020).

From this point of view, an IoT security scheme using the cloud system will be more successful in preventing the attack mentioned above because it has followed a more institutional method against buffer-overflow style vulnerabilities. Also, using a more corporate network structure, IoT device data routing provides success against DDoS attacks. On the other hand, symmetric-asymmetric and hybrid encryptions seem more suitable against eavesdropping attacks because of the success of symmetric encryption methods and privacy and public-key systems in network-based spoofing controls such as MITM attacks have been proven (Nedbal, 2018). Likewise, the symmetric-asymmetric and hybrid methods mentioned for Brute-force password attacks are much more suitable encryption methods due to their bit lengths.

Table 7 examines the strength levels of previously reviewed encryption methods compared to popular attack methods. The "✓" sign reflects a possible level of security. Inferences arise from the explanations given above. Table7 shows that lightweight encryption methods become more advantageous against Privilege Escalation and DDoS attacks due to their susceptibility to cloud systems. However, symmetric-asymmetric and hybrid systems are more successful against Brute-Force and Eavesdropping attacks.

Reference	Privilege Escalation	Brute-Force	Eavesdropping	DDoS
Belguith et al.	✓	✗	✗	✓
Chandu et al.	✓	✓	✓	✓
Chaudhry	Null	Null	Null	Null
Daddala et al.	✗	✓	✓	✗
Fischer et al.	✗	✓	✓	✗
Gunathilake et al.	Null	Null	Null	Null
Henriques et al.	✗	✓	✓	✗
Hussain et al.	✗	✓	✓	✗
Ibrahim et al.	✗	✓	✓	✗
Jian et al.	✗	✓	✓	✗
Katagi et al.	Null	Null	Null	Null
Kumar et al.	Null	Null	Null	Null
Makwana et al.	✓	✓	✓	✓
Muhammad et al.	✓	✗	✗	✓
Nagpal et al.	✗	✓	✓	✗
Naif et al.	Null	Null	Null	Null
Peshwe et al.	✗	✓	✓	✗
Saha et al.	Null	Null	Null	Null
Thirumala et al.	✗	✓	✓	✗

Wei et al.	✗	✓	✓	✗
Yousefi et al.	✗	✓	✓	✗

TABLE 7: Encryption Methods Against Popular Attack Methods.

Overall, each of the four categories mentioned is capable of securing IoT devices, but there are differences between them. Asymmetric and symmetric encryption methods are successful in terms of security but have a disadvantage in energy consumption. To avoid IoT devices' energy and processing capacity problems while creating hybrid methods, security levels, which are the most prominent feature of the methods they reference, must be reduced. Lightweight encryption methods seem to be the most suitable system for IoT devices at first glance because energy and CPU expenditures are very low during the production phase of the encryption method. However, lightweight encryption methods may be the most disadvantageous method of security methods to other categories. Encryption methods on IoT devices are still inadequate and are doomed to be improved. In the future, a new type of encryption method for those devices will be an innovative and appropriate step for securing IoT devices.

5. REFERENCES

Agren, M., & al., E. (2011). Grain-128a: A New Version of Grain-128 With Optional Authentication. *International Journal of Wireless and Mobile Computing*, *5*(1), 48. doi:10.1504/ijwmc.2011.044106

Ajtai, M. (1996). Generating Hard Instances of Lattice Problems (Extended Abstract). *Symposium on the Theory of Computing*. doi:10.1145/237814.237838

Alrawi, O., & Others. (2019, May). SoK: Security Evaluation of Home-Based IoT Deployments. *2019 IEEE Symposium on Security and Privacy (SP)*. doi:10.1109/sp.2019.00013

Avast, P. (2019b). *Avast Highlights the Threat Landscape for 2019*.

Avast. (2019a). *Avast Smart Home Security Report 2019*.

Bamiduro, W. (2018). Worldwide IoT Security Spending Will Reach $1.5 Billion in 2018. *Gartner*.

Bao, Z., Wang, L., Guo, J., Wang, L., & Wu, W. (2019). *PHOTON-Beetle Authenticated Encryption and Hash Family, Submission to the NIST Lightweight Cryptography Standardization Process*.

Beierle, C. (2020). Lightweight AEAD and Hashing Using the Sparkle Permutation Family. *IACR Transactions on Symmetric Cryptology*. Retrieved from https://tosc.iacr.org/index.php/ToSC/article/view/8467

Belguith, S., & Others. (2018). PU-ABE: Lightweight Attribute-Based Encryption Supporting Access Policy Update for Cloud Assisted IoT. *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*. doi:10.1109/cloud.2018.00137

Bellare, M., Rogaway, P., & Wagner, D. (2003). A conventional authenticated-encryption mode. *manuscript, April*.

Bellare, M., Rogaway, P., & Wagner, D. (2004). Breaking and provably repairing the SSH authenticated encryption scheme. *ACM Transactions on Information and System Security (TISSEC)*, *7*(2), 206–241. doi:10.1145/996943.996945.

Berlekamp, E. (1973). Goppa codes. *IEEE Transactions on Information Theory*, *19*(5), 590–592.

Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2012). Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. *Selected Areas in Cryptography*, 320–337. Springer.

Binance Academy, (2018). Symmetric Vs. Asymmetric Encryption. Retrieved from https://academy.binance.com/en/articles/symmetric-vs-asymmetric-encryption

Brown, A. (2017). Sonos and Bose Speakers Can Be Remotely Hijacked, Is YOUR Speaker Safe? *Express. Co. Uk*.

California Legislative Information (2018). SB-327 Information Privacy: Connected Devices. Retrieved from https://legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB327

Castryck, W., & Decru, T. (2022). An efficient key recovery attack on SIDH (preliminary version). *Cryptology EPrint Archive*, *2022*, 1–13. Retrieved from https://eprint.iacr.org/2022/073.pdf

Chandu, Y., & Others. (2017). Design and Implementation of Hybrid Encryption for Security of IOT Data. *2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon)*. doi:10.1109/smarttechcon.2017.8358562

Chang, Z. (2020). IoT Device Security Locking Out Risks and Threats to Smart Homes. *Trend Micro Research*.

Chaudhry, S. (2018). An Encryption-based Secure Framework for Data Transmission in IoT. *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. doi:10.1109/icrito.2018.8748523

Constantin, L. (2016). Update Your Belkin WeMo Devices Before They Become Botnet Zombies. *Computerworld*.

Constantin, L. (2017). Researchers Find Vulnerability in Smart Home Control Apps. Retrieved from https://www.pcworld.com/article/3223737/researchers-find-vulnerability-in-smart-home-control-apps.html

Consumer Reports (2018). Samsung and Roku Smart TVs Vulnerable to Hacking, Consumer Reports Finds. *Consumer Reports*.

Costello, C., Longa, P., & Naehrig, M. (2016). *Efficient Compression of SIDH Public Keys*. Retrieved from https://eprint.iacr.org/2016/963.pdf

Crane, C. (2021). *Block Cipher Vs Stream Cipher: What They Are and How They Work*.

Daddala, B., & Others. (6 2017). Design and Implementation of a Customized Encryption Algorithm for Authentication and Secure Communication Between Devices. *2017 IEEE National Aerospace and Electronics Conference (NAECON)*. doi:10.1109/naecon.2017.8268781

Davis, D. B. (2019). ISTR 2019: Internet of Things Cyber Attacks Grow More Diverse. *Symantec*.

Denis, S. T. (2007). *Cryptography for Developers*. Syngress.

Farooq, M., & Others. (2015). A Critical Analysis on the Security Concerns of Internet of Things (IoT). *International Journal of Computer Applications*, *111*(7), 1–6. doi:10.5120/19547-1280

Federal Trade Commission. (2015). *Internet of Things Privacy and Security in a Connected World*.

Fischer, M., & Others. (3 2019). Using Attribute-Based Encryption on IoT Devices With Instant Key Revocation. *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. doi:10.1109/percomw.2019.8730784

Fruhlinger, J. (2018). *The Mirai Botnet Explained: How Teen Scammers and CCTV Cameras Almost Brought Down the Internet*.

García, L. C. C. (2005). *On the security and the efficiency of the Merkle signature scheme*. Retrieved from Cryptology ePrint Archive website: https://eprint.iacr.org/2005/192.pdf

Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. W.H. Freeman.

Gatlan, S. (2019). Medical IoT Devices With Outdated Operating Systems Exposed to Hacking. *BleepingComputer*.

Graham, J. (2018). *Your Smart TV May Be Prey for Hackers and Collecting More Info Than You Realize, 'Consumer Reports' Warns*.

Gunathilake, N. A., & Others. (2019). Next Generation Lightweight Cryptography for Smart IoT Devices: : Implementation, Challenges and Applications. *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. doi:10.1109/wf-iot.2019.8767250

Guo, J., Peyrin, T., & Poschmann, A. (2011). The PHOTON Family of Lightweight Hash Functions. *Advances in Cryptology--CRYPTO 2011*, 222–239. Springer.

Hell, M., & al., E. (2019). Grain-128AEADv2-A lightweight AEAD stream cipher. *Information Technology*.

Hell, M., Johansson, T., Maximov, A., Meier, W., Müller, F., & Robshaw, M. (2007). Grain: A Stream Cipher for Constrained Environments. *International Journal of Wireless and Mobile Computing*, *2*(1), 86. doi:10.1504/ijwmc.2007.013798

Henriques, M. S., & Vernekar, N. K. (5 2017). Using Symmetric and Asymmetric Cryptography to Secure Communication Between Devices in IoT. *2017 International Conference on IoT and Application (ICIOT)*. doi:10.1109/iciota.2017.8073643

Hoffstein, J., Pipher, J., & Silverman, J. H. (1998). NTRU: A Ring-based Public Key Cryptosystem. *Lecture Notes in Computer Science*, *1423*, 267–288. doi:10.1007/bfb0054868

Hollister, S. (2019). *No, Nest Cams Are Not Being Hacked to Issue Fake Nuclear Bomb Threats*.

Hussain, I., & Others. (8 2018). Proposing an Encryption/ Decryption Scheme for IoT Communications Using Binary-bit Sequence and Multistage Encryption. *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. doi:10.1109/icrito.2018.8748293

Ibrahim, N., & Others. (2019). Hybrid Cryptosystem for Preserving Data Privacy in IoT Application. *IOSR Journal of Mobile Computing & Application (IOSR-JMCA)*, *6*(3), 1–8. doi:10.9790/0050-06030108

Insights, F. B. (2019). *Internet of Things Market Size, Growth IoT Industry Report 2026*.

Jian, M.-S., & Others. (2 2019). Internet of Things (IoT) Cybersecurity Based on the Hybrid Cryptosystem. *2019 21st International Conference on Advanced Communication Technology (ICACT)*. doi:10.23919/icact.2019.8701957

Journal, T. (2017). Lightweight Cryptography Applicable to Various IoT Devices. *NEC*.

Katagi, M., & Moriai, S. (2008). Lightweight cryptography for the internet of things. *Sony Corporation 2008*, 7–10.

Khomlyak, O. (2017). An Investigation of Lightweight Cryptography and Using the Key Derivation Function for a Hybrid Scheme for Security in IoT. *Blekinge Institute of Technology*.

Kinast. (2017). *Talking Doll Deemed to Be 'Concealed Listening Device'*.

Kumar, M., & Others. (2016). Lightweight Data Security Model for IoT Applications: A Dynamic Key Approach. *2016 IEEE International Conference on Internet of Things (IThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. doi:10.1109/ithings-greencom-cpscom-smartdata.2016.100

Lamport, L. (2018). Constructing Digital Signatures From a One Way Function. *Microsoft Research*. Retrieved from https://www.microsoft.com/en-us/research/publication/constructing-digital-signatures-from-a-one-way-function/

Latest Hacking News (2019). LIFX IoT Smart Light Bulb Hacked in Under an Hour. *Cyber Security News, Hacking Tools and Penetration Testing Courses*.

Library of Congress, (2020). H.R.1668 - 116th Congress (2019-2020): IoT Cybersecurity Improvement Act of 2020. Retrieved from https://www.congress.gov/bill/116th-congress/house-bill/1668

Lodge, D. (2020). Hijacking Philips Hue. *Pen Test Partners*.

Lyubashevsky, V., Peikert, C., & Regev, O. (2012). On ideal lattices and learning with errors over rings. *Advances in Cryptology--EUROCRYPT 2010*, 1–23. Retrieved from https://eprint.iacr.org/2010/119.pdf

Makwana, S. (8 2017). An Application of Homomorphic Encryption on IoT Based Green House. *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. doi:10.1109/icecds.2017.8389949

McEliece, R. J. (1978). A public-key cryptosystem based on algebraic coding theory. *Coding Thv 4244*, 114–116.

Merkle, R. C. (1979). *Secrecy, authentication, and public key systems*.

Mohananthini, N., & Others. (2020). Lightweight Image Encryption: A Chaotic ARX Block Cipher. *Journal of Circuits, Systems and Computers*, *30*(02), 2150026. doi:10.1142/s0218126621500262

Muhammad, N., & Others. (2019). Conceptual Framework for Lightweight Ciphertext Policy-Attribute Based Encryption Scheme for Internet Of Things Devices. *Malaysian Journal of Computing*, *4*(1), 237. doi:10.24191/mjoc.v4i1.6107

Munro, K. (2022). IoT Encryption: The Challenge of Missing Entropy. *Pen Test Partners*.

Nagpal, S., & Others. (2019). A New Method for Modifying Blowfish Algorithm for IoT. *International Journal of Innovative Technology and Exploring Engineering*, *8*(9S), 331–334. doi:10.35940/ijitee.i1053.0789s19

Naif, J. R., & Others. (2019). Secure IoT System Based on Chaos-Modified Lightweight AES. *2019 International Conference on Advanced Science and Engineering (ICOASE)*. doi:10.1109/icoase.2019.8723807

Nedbal, M. (2018). IoT Insecurity: 6 Common Attacks and How to Protect Customers. *Channel Futures*.

Netscout. (2019). *Highlights Dawn of the Terrorbit Era*.

Omale, G. (2018). *Gartner Identifies Top 10 Strategic IoT Technologies and Trends*.

Owasp. (2018). Internet of Things (IoT) Project. Retrieved from https://owasp.org/www-project-internet-of-things-security/

Peshwe, N., & Das, D. (10 2017). Algorithm for Trust Based Policy Hidden Communication in the Internet of Things. *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*. doi:10.1109/lcn.workshops.2017.77

Peterson, W. W., & Weldon, E. J., Jr. (1972). *Error-correcting codes*. MIT press.

Rosen, M. (2019). *Driving the Digital Agenda Requires Strategic Architecture*.

Saha, A., & Srinivasan, C. (2019). White-Box Cryptography Based Data Encryption-decryption Scheme for IoT Environment. *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. doi:10.1109/icaccs.2019.8728331

Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 124–134. doi:10.1109/SFCS.1994.365700

Shor, P. W. (1997). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, *26*(5), 1484–1509. doi:10.1137/s0097539795293172

Sysman, D. (2019). California's IoT Security Law: Why It Matters and the Meaning of 'Reasonable Cybersecurity'. *Forbes*. Retrieved from https://www.forbes.com/sites/forbestechcouncil/2019/11/20/californias-iot-security-law-why-it-matters-and-the-meaning-of-reasonable-cybersecurity/?sh=6c90b0d85f6c

Thirumalai, C., & Shanmugam, S. (4 2017). Multi Key Distribution Scheme by Diophantine Form for Secure IoT Communications. *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*. doi:10.1109/ipact.2017.8245059

Turner, P. S. D. (2018). *Symmetric Key Encryption - Why, Where and How It's Used in Banking*.

Vuldb. (2022). CVE-2019-10999. "D-Link DCS-5009L Alphapd wireless.htm Memory Corruption. Retrieved from https://vuldb.com/?id.134434

Watters, A. (2022). 30 Internet of Things Stats and Facts for 2022. *Default*. Retrieved from https://www.default.com/statistics/iot-stats-facts

Wei, B., & Others. (7 2017). A Practical One-Time File Encryption Protocol for IoT Devices. *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*. doi:10.1109/cse-euc.2017.206

Williams, A. (2018). Harmony Hub Hacked and Patched. *Hackaday*.

Yousefi, A., & Jameii, S. M. (5 2017). Improving the Security of Internet of Things Using Encryption Algorithms. *2017 International Conference on IoT and Application (ICIOT)*. doi:10.1109/iciota.2017.8073627

Zassenhaus, H. J. (2013). *The theory of groups*. Courier Corporation.

Zinevych, M. (2021). Does Encryption Protect Data Against Man-in-the-Middle Attacks? *Apriorit*.