Future Generation Computer Systems 129 (2022) 77–89

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Review article

Lightweight cryptography in IoT networks: A survey

Muhammad Rana [*], Quazi Mamun, Rafiqul Islam

School of Computing, Mathematics and Engineering, Charles Sturt University, Australia

ARTICLE INFO

Article history:
Received 11 May 2021
Received in revised form 6 October 2021
Accepted 13 November 2021
Available online 27 November 2021

Keywords:
Lightweight cryptography
Block cipher
Stream cipher
Elliptic curve cipher
Internet of Things (IoT)
Security

ABSTRACT

With the advent of advanced technology, the IoT has made possible the connection of numerous devices that can collect vast volumes of data. Hence, the demands of IoT security is paramount. Cryptography is being used to secure the authentication, confidentiality, data integrity and access control of networks. However, due to the many constraints of IoT devices, traditional cryptographic protocols are no longer suited to all IoT environments, such as the smart city. As a result, researchers have been proposing various lightweight cryptographic algorithms and protocols to secure data on IoT networks. This paper discusses state-of-the-art lightweight cryptographic protocols for IoT networks and presents a comparative analysis of popular contemporary ciphers. In doing so, it has classified the most current algorithms into two parts: symmetric and asymmetric lightweight cryptography. Additionally, we evaluate several recently developed block cipher and stream cipher algorithms in terms of their security. In the final section of this paper, we address the changes that need to be made and suggest future research topics.

Contents

1. Introduction

The Internet of Things (IoT) refers to everyday things that are readable, addressable, locatable, and identifiable via data sensing devices and manageable through the Internet. IoT devices are accessible by communication techniques such as RFID, wireless, wired, or other methods. Everyday objects not only include high-tech electronic devices such as mobile phones and vehicles, but things that we do not generally consider as electronic at all, like food items, animals, clothing, water, waste bins, trees, and so on. The IoT objective is to allow things to be communicated anywhere, anytime, with anything, preferably applying any network or service. In the last few years, the IoT has grown exponentially and now occupies our lives in areas as diverse as cities, agriculture, hospitals, the environment, homes, roads, etc. IoT end devices are usually equipped with different types of sensors and

[*] Corresponding author.
E-mail address: mrana@csu.edu.au (M. Rana).

https://doi.org/10.1016/j.future.2021.11.011
0167-739X/© 2021 Elsevier B.V. All rights reserved.

actuators, which collect numerous data and send the accumulated data through cyberspace to monitor, analyse, control, and reach various conclusions [1]. Most of these data are real-time data and help us make correct decisions in different service domains. However, this Internet-driven raw data needs to be transferred securely and switched to human-understandable information to gather knowledge and use this knowledge in various domains such as the smart city, agriculture, the environment, interactive transport, and electricity grids.

The smart city is one example of an IoT domain that has security problems that can be overcome by improving cryptography. Seventy percent of smart city services are currently provided in three fields: traffic, safety, and power [2]. Fig. 1 shows the leading smart city model around the globe. The United Nations Population Fund indicates that more than half of the world's population now resides in a city. By 2050, this is expected to increase to about 68% [3]. In China, more than 200 smart city developments are in progress [4]. However, smart city domains create numerous security and privacy challenges due to various weaknesses in each layer of a smart city's network architecture. For example, in 2015, approximately 230,000 Ukrainian residents experienced an extended period of power interruption when intruders hacked into the electricity grid [5]. The IoT plays a crucial role in predicting and managing natural disasters such as bushfires, earthquakes, hurricanes, and tsunamis. Various IoT sensors can help avoid and control damage to life and the environment from forest fires. The sensors can be installed around the edges of the forest and can continuously monitor the temperature and carbon content in the region. In cases of emergency, the IoT can aid an immediate response by preparing and distributing the environment report. The 2019–2020 bushfires in Australia burned 46 million acres and cost 2.9 billion USD [6]. According to United Nations Food and Agriculture, the world will be need 70% more food production in 2050 [7]. Smart agriculture will play a decisive factor in this growing market. For instance, in Chile, using remote sensors reduces 70% of the water requirement in blueberry production [8]. A possible security attack in an agricultural business could result in significant human and financial consequences. In June 2017, the 'NotPetya' malware attack in one agricultural-related organisation cost about half a billion U.S. dollars [9]. The enormous data shared in IoT-enabled environments can be exploited by malicious attackers, which creates a security challenge [10]. Hence, addressing and minimising these security and privacy risks by promoting efficient security solutions is crucial for the success of IoT domains.

Ensuring privacy in IoT end devices is challenging for several reasons. First, the CPU in IoT devices is minimal and cannot compute complex algorithms [11–18]. Second, the power consumption of the security algorithm should be low since most IoT devices are battery-powered [12,14–16,18–22]. Third, simple sensors are connected to cover a large physical network [18,20]. Finally, implementing the security algorithm needs to be cost-effective by deploying as few devices as possible [1,13,23–25]. Conventional cybersecurity cryptography such as AES (Advanced Encryption Standard), RSA (Rivest–Shamir–Adleman), DES (Data Encryption Standard), Blowfish, and RC6 cannot be used immediately in these smart domains because of the heterogeneity, scalability, and dynamic features of the IoT. Most of these algorithms consume more energy while operating. For example, AES uses 2.9 kB of flash and 1.2 kB of RAM [21]. Researchers have compared several WSN sensor motes and found that resource-constrained devices have as low as 2 kilobytes (kB) and 1 kB of Random Access Memory (RAM) and Electrically Erasable Programmable Read-Only Memory (EEPROM), respectively [21]. Such sensors cannot use the resource-consuming conventional security approaches [26,27]. Hence, secure communication is one of the most significant concerns in Low Power and Lossy Systems. This undoubtedly defines the necessity to develop Lightweight Cryptographic (LWC) algorithms for IoT security.

Table 1
IoT layer architecture and task.

Layer	Component	Tasks
Application layer	Third-party application, consoles, websites, touch panel.	Machine learning, business model, graphs and flowcharts.
Middleware layer	Vendor-specific third-party application.	Machine learning, processing, pre-processing, and real-time action.
Network layer	Nodes, gateways, firmware.	Transmit and process data, device management, process and secure routing.
Perception layer	Sensors (temperature and humidity), actuators (relays and motor).	Transfer data, identity, monitor, acquisition and action.

With growing interest in the IoT, the fundamental research question is: What lightweight cryptography has been developed to address the many IoT security issues?

This question must be addressed if researchers are to develop and execute secure and efficient IoT networks. A literature review on lightweight cryptography algorithms was deemed essential to ensure IoT communication security. Hence, this paper focuses on the following main research questions:

1. What lightweight cryptography has been developed to address the IoT security issues?
2. How can lightweight cryptography secure an IoT structure?
3. What consequences do the findings have on the future of IoT research?

This paper addresses the most current state state-of-the-art research in lightweight cryptography for the years 2019 and 2020. It also presents a comparative analysis of most current lightweight algorithms, such as LCC, LWHC, Modified PRESENT and SAT_Jo. The paper also evaluates the most recent protocols using a set of matrices like block size, key length, gate area, technology value, number of encryptions or decryptions, latency, and throughput. This comprehensive evaluation demonstrates the requirements of lightweight cryptography ciphers. This paper is organised into seven sections. Section 1 introduces the IoT and the need for the development of LWC in IoT systems. The IoT architecture and threats are presented in Section 2. Section 3 discusses IoT architecture and devices used according to the structure. Section 4 describes security mechanisms in IoT systems. The most recent lightweight cryptography developments in the IoT are discussed in Section 5. Section 6 presents a critical analysis of lightweight ciphers, identifies the research gaps and suggests further research. Finally, the conclusion is presented in Section 7.

2. IoT architecture and threats

This section examines the different layers of the IoT architecture, according to a device's functionality and possible exposure to various attacks. IoT domains show enormous possibility. However, IoT networks connect with heterogeneous devices with mixed operating systems and different communication protocols, such as wireless, Zigbee, and mobile technology, which can create considerable security and privacy threats [28]. In this section, we outline IoT architecture and discuss the different layers. We also present the various vulnerabilities to attacks on the IoT according

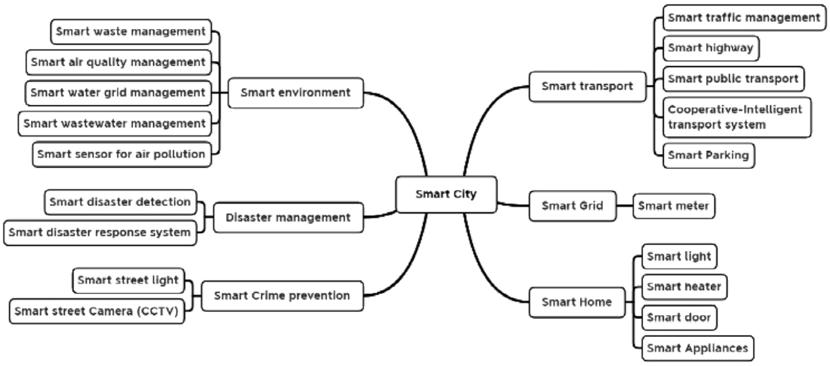

Fig. 1. Major services in a smart city network.

to the architectural layer. However, we discuss perception and network layers in detail as these two layers are more vulnerable due to resource constraints.

The IoT architecture contains four distinct critical layers: (i) perception layer, (ii) network layer, (iii) middleware layer, and (iv) application layer [12]. Table 1 shows the IoT layers with the components of each layer and their tasks in detail.

2.1. Application layer and security attacks

The application layer is the top layer in the IoT infrastructure. The middleware layer passes information to this layer to process different applications. The application layer represents the IoT data as a business model, flowchart, and graph. Smart cities, smart homes and smart cars are some of the examples of application layer automation. Some application layer attacks include denial of service attacks [17], buffer overflow attacks [29], cross-site Scripting attacks [29], SQL injection attacks [30], phishing attacks [31], and data privacy issues [32].

2.2. Middleware layer and security attacks

The middleware layer operates the vendor-specific services for various IoT node information, which link the network and application layers. This link facilitates processing, pre-processing, and storing IoT node information based on the third party and node requirement [29]. An intruder can introduce various attacks in the middleware in a different way, such as an application security attack [33], unauthorised access attack [34], replay attack [35], sleep deprivation attack [36], data security attack [37] etc. The middleware and application layers both use resource-rich devices that can use traditional cryptography to secure IoT networks. Consequently, these two layers are not focused on in this paper. This paper discusses the security techniques of resource-constrained devices of the network and perception layers.

2.3. Network layer and security attacks

The network layer, also called the transit layer, processes and securely routes or transmits data throughout the IoT infrastructure. This layer uses different protocols like Zigbee, Bluetooth, IR, and 6LowPan for data transmission. For further processing and action, the network layer depends on the middleware layer. Following are some of the different possible attacks in this layer.

- Eavesdropping: Eavesdropping is a passive attack and extracts the message contents from network broadcastings. It snoops, captures, and sniffs out broadcasted data and initiates different attacks or steals different critical information [38].
- Device cloning attacks: As IoT devices are easily accessible from the network, an intruder can create a clone of the device and compromise IoT network infrastructure using these devices [39].
- Spoofing attacks: Devices are connected to the network either directly or through a gateway in the IoT structure. An attacker can physically capture the node or gateways and can replace or reprogram them with malicious code. Edge devices and gateways must be authenticated, and the data should be encrypted to prevent this kind of attack [40].
- DDoS attacks: IoT devices are exposed to these attacks as the IoT architecture uses heterogeneous and resource-constrained nodes. Firstly, an attacker captures the device's credentials and then gets access to the gateways/devices. A hacker uses network information to explore the IoT devices and can initiate a DoS attack by sending fake packets up and down the entire system [41].
- Key attack: Some devices have pre-shared keys that are hard-coded within the code. Hence, the intruder can easily capture this information [42].
- Traffic analysis: Traffic analysis is another passive attack that can retrieve valuable information from the network traffic, such as source and destination details from the header of the transmitted communications [15].
- Brute-force attack: This is known as an exhaustive search. This is a cryptographic hack that depends on predicting possible patterns of a targeted password to discover the correct password [43].
- Man-in-the-middle attacks: Due to the heterogeneous nature of IoT architecture, an intruder can secretly capture communications between two parties to eavesdrop or modify traffic travelling between them. Attackers may capture personal information, login credentials, disrupt communications, and corrupt data [8,17].
- Sinkhole attacks: In this type of attack, an adversary generates sinkholes to attract traffic flow from IoT devices. The attacker can later reroute the network traffic to devices other than the destination gateway. This attack compromises the IoT device's privacy and confidentiality [44].

2.4. Perception layer and security attacks

The crucial role of IoT systems is to gather and transmit information from the real world. Hence, the perception layer possesses different kinds of data gathering, processing and transmitting devices such as pressure and temperature sensors, Bluetooth, Zigbee, etc. The perception layer can be split into two parts: (a) the perception node (sensors or controllers, etc.) and (b) the perception network, which communicates to the upper layer of the IoT architecture [12]. Perception nodes, such as sensors and

Table 2
Comparison of IoT devices in terms of communication technology and interfaces.

Device	Interface provided	Communication
Raspberry Pi	HDMI, micro USB, USB 2.0, Ethernet, WLAN, Blueetooth4.2, CSI camera port, DSI display port	IEEE 802.11b/n/ac wireless LAN, Bluetooth 4.2, BLE and Gigabit Ethernet
Beagleboard	HDMI, 3.5 mm stereo in/out, I2X, UART, LCD	Ethernet, WLAN, Bluetooth
Netduino	UART, I2C, SPI	Ethernet, low power Wi-Fi
Arduino	USB, UART, ADC, I2C	Ethernet, IEEE 802.11ah, Wi-Fi support
OpenMote-CC2538	I2C, SPI	IEEE 802.15.4 with 2.4 GHz band
TELOSB	I2C, SPI	IEEE802.15.4 at 250 Kbps rate
Open Mote-B	I2C, SPI, USB 2.0	IEEE802.15.4 g
LSN50	I2C, ADC, DAC, USART, USB	Wireless chip

actuators, collect and control information. However, the perception network transmits the collected information to the gateway. The perception layer uses Zigbee, RFID, GPS and Long Range Wide Area Network (LoRaWAN) technology [16].

In the perception layer, the node could be attacked or intruded upon or compromised physically. Generally, this compromised device is called a faulty node. To ensure the quality of service, it is necessary to detect the defective devices and take action to avoid further degradation of the service. A localised fault detection algorithm was used to discover the faulty nodes in a WSN [12]. Da Silva et al. [45] suggested a decentralised intrusion discovery system paradigm for wireless sensor networks. Wang et al. [46] proposed that there was a probability of intrusion and discovery by hackers in both homogeneous and heterogeneous WSN.

Cryptography cipher algorithms and key management schemes are used to secure perception layer network communication. Device authentication uses a private key algorithm with greater scalability and can ensure the system's security without a complicated key management algorithm [47]. Key management involves secret key generation, distribution, storage, updating, and destruction. A key distribution system can be divided into four categories: (a) key broadcast distribution [48], (b) group key distribution [49], (c) master key pre-distribution; [50] and (d) pairwise key distribution [51,52]. However, the perception layer is vulnerable to the following attacks:

- Physical capture or damage: Generally, IoT devices are located in public or insecure places. Consequently, physical nodes could be captured, damaged, or compromised. An intruder can tamper with the device and use it to login to the IoT gateway to modify and capture network traffic and other secret information [53].
- Code injection attacks: As an intruder can access the physical IoT devices, the key can manipulate the devices by introducing malicious codes into the devices [39].
- Jamming attacks: This is a widespread method of attack for perception layer devices. IoT edge devices use the wireless protocol for network communication to transfer data to a different layer, program devices and receive instructions from the upper layer [22].
- Replay attack: The attacker sends back a previous message to the target device to gain network trust. So, the network can compromise the security as the previous message was authenticated [54].
- Battery draining: An intruder's objective is to drain the IoT device's battery. After capturing the device, an adversary continuously performs an energy-hungry operation. By doing this, an attacker can deteriorate the network by reducing the battery power of resource-constrained IoT sensors [55].

While there are various attacks mentioned above, we also suggest several other mitigation techniques such as cryptography, authentication, securing physical devices, etc. However, these attacks are still creating serious concerns for people using low resource devices in IoT domains.

3. Devices in different IoT layers

IoT devices are present in all architectural layers with limited proficiency due to low memory, internal storage, computational capability and power. The IoT environment comprises various service architectures, protocols, and network designs to deal with billions of IoT nodes that exchange information. IoT devices can be generally divided into three categories, Class 0, Class 1, and Class 2 [56,57].

Class 0 or low-end IoT devices often have constrained resources like memory, power, and computational capability, which are mainly present at the first or perception layer of the IoT architecture. These low-end devices sense data and communicate with lightweight communication protocols. The RAM varies from 1 to 50 kB, and flash memory ranges from 10 to 50 kB [58]. Security is the primary concern in these low-end nodes as they are particularly vulnerable to threats.

Class 1 or middle-end IoT devices have more resources compared to low-end nodes. These devices are basic microcontrollers and sit over low-end devices in the IoT architecture to improve the abilities of class 0 node devices [59]. These devices have a higher clock rate, from 100 MHz to 1.5 GHz, and their RAM varies from 100 kB to 100 MB. Their flash memory varies from 10 kB to 100 MB. These devices can use data encryption technology to secure data. Arduino and Netduino are two of the middle-end nodes, which also present at both the first and second layers of IoT design [60,61].

Single-board computers have a high number of resources in terms of CPU, RAM and flash memory and are in Class 2. These devices support traditional operating systems such as LINUX and UNIX [62] and growing technologies like artificial intelligence, machine learning, deep learning and neural networks. They have comparatively fewer security concerns due to their higher resources [63]. Table 2 shows a comparison of communication network technologies and the interfaces of the different IoT devices [58].

IoT devices have fundamental restrictions such as processing power, storage, memory, power consumption and connectivity [64,65]. Lightweight cryptography needs to consider these limitations while designing and implementing an IoT network. The effectiveness of a network depends on design complexity, power consumption, throughput, and CMOS technology.

Design complexity is determined by the gate value (GE). Power intake is crucial for active devices like wireless sensor devices. Nonetheless, energy consumption is the principal interest of passive devices, such as RFID tags and smart cards. Energy use is directly related to chip area [66]. A small area indicates low energy consumption. CMOS technology also affects performance qualities. Distinct technologies and standard-cell libraries generate different outcomes. For instance, the identical execution of PRESENT generates 1075 GE on 0.18 μm, 1169GE on 0.25 μm and 1000GE on 0.35 μm CMOS technology [67]. Table 3 presents the GE and power consumption in relation to CMOS technology [58]. Table 4 describes the frequency, energy, RAM and ROM features of different microcontrollers in the marketplace [68].

Table 3
Characteristics of different CMOS technology.

CMOS technology node (μm)	Gate density (kGEs/mm^2)	Power consumption (nW/MHz/GE)
0.35	6	18
0.18	125	15
0.13	206	10
0.09	404	7
0.065	800	6.68

Table 4
Features of various microcontroller platforms.

Microcontroller platform	Frequency (MHz)	RAM (kB)	ROM (kB)	Power (mA)
8-bit	4–8	0.064–4	1.4–128	2.2–8
16-bit	4–8	2–10	48–60	1.5–2
32-bit	13–180	256–512	4000–32,000	31–100

4. Securing the IoT system

Section 4 concisely discusses lightweight algorithms used to secure IoT network communications. Furthermore, this part classifies the latest developments in lightweight algorithms. Fig. 2 illustrates different types of the most recent lightweight cryptography, which is primarily split into two categories, symmetric and asymmetric algorithms. The symmetric lightweight algorithms are further divided into Lightweight Block Ciphers (LWBC) and Lightweight Stream Ciphers (LWSC). Elliptic curve cryptography (ECC) falls under asymmetric cryptography. The factors of the lightweight cryptographic primitives are evaluated by the key size, block size, number of rounds, and structures. We will discuss the recent development of three cipher technologies used to secure resource-constrained IoT networks — block cipher, stream cipher and elliptic curve cipher.

4.1. Lightweight block cipher

Block cipher is a symmetric cipher where a complete block of data is processed simultaneously. Lightweight block ciphers are used in two distinct styles of networks: Substitution–permutation networks (SPN) and Feistel networks (FN). A Feistel structure is designed to have the same circuit for encryption and decryption, which keeps the costs down. Feistel structures use the same program code for encryption and decryption procedures, which ensures low memory requirements [69]. The SPN is faster without a key schedule, but this makes the system vulnerable to attacks. The SPN structure is more suitable for security because of lesser execution requirements, and it has a lower power expenditure [28].

Key size, block size, structure type, and the encryption/decryption rounds are the primary considerations to evaluate a lightweight block cipher. Thangamani and Murugappan [70] recommended that a lightweight algorithm must focus on the three challenges: minimal memory, low power intake, and address insufficient security. Zhao, Yan, and Li [71] recommended that a lightweight algorithm should have a small block size of 32–64 bits compared to the conventional 64 and 128 bits block size.

4.2. Lightweight stream cipher

A stream cipher is another category of lightweight symmetric cryptographic algorithm, which encrypts and decrypts data bit by bit. A stream cipher is simpler and quicker compared to block ciphers. This cipher was developed by applying linear feedback shift registers (LFSRs) and nonlinear feedback shift registers (NLFSRs) [72]. It is extensively applied in WSNs, cell phones etc. [72,73]. Stream ciphers are used because they have fewer computation requirements and are quicker compared to other ciphers. Some common stream ciphers are RC4, Salsa20, Trivium and Chacha.

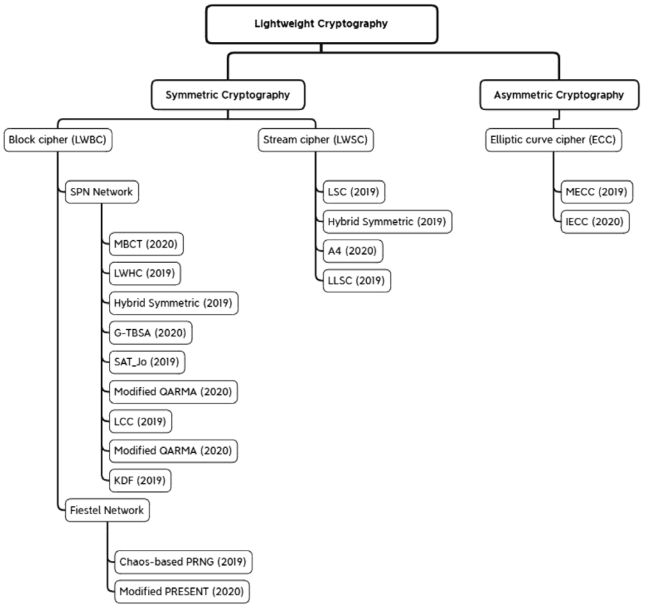

Fig. 2. Classification of recently developed lightweight cryptography algorithms.

4.3. Lightweight elliptic curve cipher

Asymmetric ciphers like ECC are also used to secure IoT networks. ECC ensures authentication and confidentiality [49]. Asymmetric ciphers use a larger key size and more memory consumption, which makes this cipher less popular in terms of IoT security. RSA and ECC are asymmetric cryptography devices that can be used in IoT network security. ECC uses a smaller key size to achieve a similar security level compared to RSA. El-Gamal and Diffie–Hellman key algorithms can secure IoT systems. AES [50] is 100–1000 times quicker than ECC on 8-bit microcontrollers. ECC is presently the most popular option chosen to provide security in IoT systems.

5. Recent lightweight cryptography for IoT security

This section briefly reviews the latest lightweight cryptographic protocols to secure IoT networks in resource-restricted systems.

Prakash, Singh, and Khatri [42] developed a new hybrid algorithm called lightweight hybrid cryptography (LWHC) that uses a combination of LED and PRESENT ciphers with a compact key scheduling algorithm SPECK. This system used RECTANGLE S-Box to make it faster and more robust. Encryption is done using LED, PRESENT and RECTANGLE S-Box. However, the SPECK algorithm is also used for key scheduling. The proposed system used a 64 bits key of plain data to encrypt and perform an XOR operation with a 128 bits schedule key. The LED cipher used the typical key algorithm, which is not that resistant to key attacks. However, the 128 bits SPECK key schedule algorithm is used in the proposed system, which gives it a lightweight nature and secures it from various key attacks. This advanced system uses a 64-bit block plain text XOR with 128 bits key that maintains its robustness.

The proposed algorithm is lightweight and secure from various key attacks but does not resist other security attacks.

Noura, Couturier, Pham, and Chehab [43] proposed a lightweight stream cipher scheme (LSC) to reach a high level of security. This is based on the dynamic key-dependent method. This system includes only a few simple operations that reduce costs. It involves cryptographic primitives, which change in a dynamic, lightweight manner for each input block. Security and performance studies confirm that the proposed cipher attains a high level of effectiveness and robustness, making it suitable for resource-restricted IoT devices. It uses 128, 192, and 256 bit secret keys with a nonce. This algorithm is a combination of CR4, Pseudo-random number generator (PRNG), and Linear feedback shift register (LFSR) with 12 to 14 rounds. This algorithm shows high periodicity, low energy consumption and is resistant to statistical, algebraic, and brute-force attacks. However, it is not adequate for disclosure and de-synchronisation attacks.

Shantha and Arockiam [72] designed the SAT_Jo system, which is based on the substitution–permutation network with a new lightweight block cipher that is appropriate for tag-based functions of the IoT. This system computes a 4 × 4 S-box by 2^4 orders of the Galois field. This design is based on the SPN of a block cipher with DES and PRESENT [74] and involves 31 rounds. It uses a 64 bit block and 80 bit key size. This system presents an adequate level of security to the resource-constrained nodes of tag-based applications. The researchers found that it offers a better balance between performance, resource requirements and security for resource-constrained IoT systems.

Kubba and Hoomod [75] considered a Hybrid symmetric model based on the PRESENT (block cipher) and Salsa20 (stream cipher) cryptography algorithms. This algorithm also used a 2D logistic map of a chaotic system to generate pseudo-random keys, which produced more complexity. This proposed algorithm aimed to enhance the complexity of the PRESENT algorithm and keep the performance of computational operations as minimal as possible. The proposed algorithm uses a 64-bit symmetric block cipher algorithm and a 128-bit length key. The block PRESENT algorithm is mainly considered for its fast performance [76]. Salsa20 was proposed because of its efficiency of use with constrained nodes [77]. Therefore, PRESENT and Salsa20 cipher algorithms meet the lightweight algorithm's speed and complexity requirements. Twenty rounds of PRESENT key algorithms were used instead of the 31 rounds of keys. The Salsa20 cipher algorithm was used to generate keystream. The suggested algorithm has a significant level of randomness and demonstrates efficient performance with rapid execution times. Hence, the executed time of the proposed algorithm is a satisfactory lightweight algorithm speed. However, although the proposed algorithm maintains a minimum computational time, it introduces more complexity.

Noura et al. [78] proposed One Round Cipher (ORC), which is a lightweight algorithm based on a dynamic structure with a single round roll. This model generates a dynamic key, which is then used to develop two robust substitution tables, a dynamic permutation table, and two pseudo-random matrices. This dynamic cipher structure uses a single round while providing a high level of randomness and security. The proposed cipher is resistant to statistical attacks that exhibit high randomness. The OCR cipher shows a high level of sensitivity, which protects it from key-related attacks.

The Generalised Triangle Based Security Algorithm (G-TBSA) [18], designed by Ahmed et al. is applied in wireless sensor networks (WSNs) with low power Wi-Fi. G-TBSA is a combination of resource-friendly data encryption and an efficient key generation mechanism. The key generation process is the heart of the algorithm since it uses fewer resources to generate the keys, which minimises the complexity and provides energy efficiency. The proposed mechanism used the non-right-angle triangle based method, and the output signal was used instead of time calculation. The proposed G-TBSA is more energy-efficient than other algorithms. However, this method works only for sensor devices.

Modified PRESENT [79] is a new lightweight PRESENT cipher designed by Chatterjee and Chakraborty. It has changed the original PRESENT cipher by reducing the encryption and modifying the key register. The key register is updated by encrypting its value by adding a delta value function of TEA (Tiny encryption algorithm), which is another lightweight cipher. The additional layer helps to reduce the PRESENT rounds from 31 to 25, which is the minimum required for security. The efficiency of the proposed algorithm is increased by encrypting the key register. The proposed algorithm proves its superiority by analysing different software parameters like N-gram, Non-Homogeneity, Frequency Distribution graph and Histogram. This algorithm shows better performance in terms of gate value. However, it has not been tested for battery consumption.

The Key-Dependent and Flexible (KDF) [80] scheme supports the logic and delivers a new lightweight cipher with a simple round event and dynamic key for every message. Subsequently, the proposed cipher is used for real-time multimedia applications and uses limited resources. This algorithm generates dynamic cryptographic primitives and performs the mixing of selected blocks in a dynamic pseudo-random manner. Accordingly, different plaintext messages are encrypted differently, thus avoiding the avalanche effect. This proposal shows a high level of immunity to attacks such as statistical, differential and brute-force attacks. KDF needs less computational complexity (less delay) than AES.

Roy, Rawat, and Karjee [15] proposed Lightweight CA (LCC), which is a lightweight cellular automaton (CA)-based cipher. This method encrypts information at the perception layer, which proves more efficient than some of the existing ciphers like DES and 3DES. This algorithm passes the randomness tests according to the National Institute of Standards and Technology (NIST). It also passes all the DIEHARD tests, which shows the strong security features of LCC. Modified QARMA [71] proposed by Zhao, Yan, and Li, a part-iterative architecture for QARMA, integrates encryption and decryption operations. Systems use ASIC in CMOS 55 nm technology, where a maximum frequency of 666.67 MHz is achieved. The results show that this model has achieved a 54% reduced gate area and is concurrently enhanced by 25 times its maximum frequency. The throughput increased by 1.56 times when contrasted with the unrolled implementation. However, further modification was needed to optimise the resource intake.

Chaudhary and Chatterjee [81] designed the Modified Block Cipher Technique (MBCT), which is a combination of one Matrix Rotation, XOR and the Expansion function. The encryption process primarily changes the Expansion and Round key creation function. The key length is 256 bits, and 256 bit block plain text was used in this process. This algorithm uses 32 rounds and needs less encryption and decryption time compared to AES, DES and SIMON. Modified MBCT also used less memory than AES, DES and SIMON.

Thangamani and Murugappan [44] propose lightweight cryptography primitives by combining the differential logical pattern (DLP), S-box pattern generation, and random key generation, and decreasing the time memory complexity by decreasing the number of pattern structure elements. It uses 16 × 16 blocks of the input message. This LWC technique provides higher complexity to the scheme, which makes it difficult for a hacker to capture information. DLP uses less memory and time. In S-box, the random key and input text or image generate the patterns. The DLP method is applied to encrypt the input blocks before transmitting the message. The receiver generates the key for decryption by inversing the random key and S-box. Then, the DLP decryption technique is used to rebuild the initial information. The benefits of this method are reduced memory consumption and time complexity.

Hamzaab et al. developed Chaos-based PRNG [27] encryption to maintain patient data confidentiality. This algorithm is used a keyframe immediately after extracting them using a video summarisation technique from video information. The symmetric block encryption system operates the proposed chaos-based algorithm with one set of confusion and diffusion processes. It uses a new Pseudo-Random Number Generator (PRNG) based on Zaslavsky's chaotic and 2D logistic map. Statistical probabilistic performance makes this system more appropriate for real-time. This method is effective in resisting various attacks such as differential, statistical, and exhaustive because the hackers cannot find the secret keys. This technique is fast and safer than other current algorithms.

Gyamf, Ansere, and Xu [82] proposed an algorithm to improve ECC — it contains a lightweight ECC based on the Diffie–Hellman key exchange technique and Advanced Encryption Standard (AES). Two sets of public and private keys are generated from ECC Standard curves constructed on key sizes of 256-bits and 512-bits. Key K, generated by the cloud server, is used for encryption and decryption by applying AES with 10 different sets of data. Sensor information was encrypted at the IoT devices with a low key before it was transmitted to the upper layer, which confirms primary security as the Internet service is used. The IoT Edge obtains the generated public keys from the remote server, then extracts and updates the different IoT devices. It ensures the highest security level as it performs the higher standard of encryption and decryption. This proposed Modified ECC (MECC) has decreased the complexity of conventional algorithms and considerably reduced the run time for heavy encryption. Therefore, this solution is suitable for resource-constrained IoT nodes.

Khan et al. proposed IECC [83] as a secure framework for authentication and encryption in IoT-based medical sensor information. The IECC is a curve-based system that has a specific base point derived from functions of a prime number. This system mixes biometric parameters and user credentials. An additional secret key is created to improve network security compared to typical ECC. A public key is generated to encrypt the information, and a private key is used to decrypt the data. The secret key is generated from the private key, public key and elliptic curve point. The system possesses security requirements such as low encryption and decryption times and low communication overheads. The mean encryption and decryption time of IECC is 1.032 and 1.004 μs, respectively. This value is less than the ECC and RSA values. Statistical analysis also shows the strength of this scheme.

Mohandas et al. [84] introduced an A4 stream cipher using a Linear Feedback Shift Register (LFSR) and a Feedback with Carry Shift Register (FCSR). A4 shows higher security and is easy to implement in different applications to secure data communication. A seed box containing 256 hexadecimal numbers each of 128 bits is established at both ends: sender and receiver. After receiving the seed value, the LFSR primarily clocks to set a number of times. This clocking of LFSR ensures the second level of security as an intruder is completely unaware of this calculation. A4 is entirely protected from algebraic attacks due to the arrangement of LFSR and FCSR. This algorithm also shows resistance to brute-force and differential attacks.

A new lightweight stream cipher (NLSC) [85] on the Chaos stream cipher algorithm is a combination of a chaotic system and two Nonlinear Feedback Shift Registers (NFSRs) designed by Ding et al. An 80 bit secret key has been used in this algorithm, which is a combination of a Logistic chaotic system, two 40-level NFSRs, and three multiplexers. The test results show that the stream cipher has good cryptographic characteristics and resists statistical attacks. This algorithm is also suitable for resource-constrained devices.

6. Discussion and limitations of existing lightweight cryptography

Recently developed cryptography can be split into two types, symmetric and asymmetric. Block cipher and stream cipher represent a symmetric algorithm, whereas ECC represents the asymmetric cipher. Symmetric ciphers use reduced key length compared to the asymmetric algorithm. Hence, they are vulnerable to security attacks because of their less complex nature. Asymmetric ciphers use more complexity to secure IoT network communications, but the larger key length makes them slower. Studying these crucial considerations is necessary to create an algorithm that will use less power, decrease complexity, take less time, and adequate security to low-end IoT devices [86].

Hash functions denote another cipher besides symmetric and asymmetric. The function, also called message digests, is a keyless cipher. Hashing supports integrity instead of confidentiality services. This is a one-way cipher and cannot decrypt to plain text. It has two parts – the message and the message digest. Any alterations to the message would create a different message digest. Some of the lightweight hash functions are LNHASH [87], PHOTON [88], HVH [89] and SPONGENT [90]. Symmetric cipher Message authentication codes (MAC) provide authenticity and integrity in the data transfer process. It generates a tag from the message and a secret key. This can be used to verify the message integrity, which has been done intentionally or accidentally during transmission. LightMAC [91] and CHASKEY [92] are examples of the lightweight MAC cipher. We are not discussing the hash function and MAC in this chapter as we will focus more on other symmetric ciphers like block, stream and ECC ciphers.

Table 5 shows a comparative analysis of some of the most recent (2019–2020) proposed protocols. The following protocols have been reviewed in this paper: LWHC, SAT_Jo, Modified PRESENT, LCC, Modified QARMA, DLP and MBCT. They are block cipher algorithms that are suitable for resource-constrained devices in an IoT environment. Modified PRESENT and LWHC both used 25 rounds of the algorithm; thus, they need less computational power. Although they are not vulnerable to key attacks, they are vulnerable to other attacks. On the other hand, MBCT applied 256 bits of key and 32 rounds of the algorithm, which required low memory but needed to verify differential and linear cryptoanalysis attacks. While Modified ECC and IECC are suitable for resource-constrained devices, IECC is only suitable for authentication purposes. One Round ORC is a stream cipher algorithm that is resistant to statistical analysis, but latency is comparatively high. Hybrid symmetric ciphers with PRESENT and Salsa20 need less computational power. However, more time (computational time) is required to calculate the algorithm. G-TBSA requires low energy consumption but is only suitable for wireless sensor networks. LSC and KDF are suitable for stream cipher and use less computational power and less power to generate the algorithm, but they showed less resistance to disclosure and de-synchronisation.

Table 6 compares various lightweight ciphers that could be used for resource-restricted devices. Block size, key size, gate area, latency and throughput are the main parameters for lightweight primitives. Lightweight cryptography primitives' performance is calculated by several matrices: key size, rounds, latency, throughput and gate area. The strength of the cipher also depends on the strength of the Substitution box (S-box) and Permutation box (P-box).

Table 5
A comparative analysis of the most recent ciphers.

Proposed work	Algorithm, tools and techniques	Key size, block size and rounds	Cipher and network type	Features
LSC (2019) [43]	CR4, PRNG, Dynamic key, XorShift64, LFSR, XOR	Secret keys 128, 192, 256 nonce and dynamic key 512 bits, 12 to 14 rounds	Stream	High periodicity Low energy and computational power are needed Resistant to statistical, algebraic, and brute-force attacks
Hybrid Symmetric (2019) [75]	PRESENT and Salsa20, XOR, Chaotic system, Pseudo-random keys	64 bits block, 128 bits key	Block	Works efficiently with fast execution time. Less computational power is needed
LWHC (2019) [42]	LED, PRESENT RECTANGLE S-Box, XOR, SPECK key generation	64 bits block, 128 keys	Block	Robust to key attacks
ORC (2019) [78]	KSA, RC4, SHA-512	One round	SPN, FN	Resistant to statistical analysis, visual degradation and sensitivity test
SAT_Jo (2019) [72]	PRESENT, DES, S-Box	64 bits block, 80-bit key, 31 rounds	Block cipher, SPN	This algorithm offers a better balance between performance, resource requirements and security for resource-constrained IoT systems
G-TBSA (2020) [18]	TBSA, Non-right-angle triangle		WSN	Low energy consumption Suitable only for wireless sensor networks
Modified PRESENT (2020) [79]	PRESENT, TEA, S-Box, P-Layer	64 bits plain text, 80 bits key, 25 rounds	SPN, FN	This algorithm shows better performance in terms of gate value
KDF (2019) [80]	Dynamic key generation, SHA-512	64 bits Key		The effect of error propagation is limited to the byte
LCC (2019) [15]	CA, PRN (pseudo-random number), RV512, GCA, non-linear		Block chipper	Resistant to brute-force, linear cryptanalysis, differential cryptanalysis attacks
Modified QARMA (2020) [71]	QARMA, ASIC, CMOS 55 nm, S- Box, Boolean, Permutation, MixColumms	64 blocks, 27 rounds	Block cipher, SPN	This algorithm reduced 54% of the area, and simultaneously, the frequency increased by 25x
MBCT (2020) [81]	Matrix location, XOR, Expansion function	256 bits Key. 256 block plain text. 32 rounds	Block cipher	Less encryption and decryption time are required compared to AES, DES, and SIMON
Chaos-based PRNG (2019) [27]	PRING, confusion and diffusion operation. 2-D chaotic system		Block cipher, Symmetric	Resistant to various attacks: differential, statistical, and exhaustive attacks because of secret keys
MECC (2019) [82]	ECC, AES, Diffie–Hellman	256 bits and 512 bits	Asymmetric, ECC	It reduced the complexity of the conventional algorithms, which made them suitable for resource-constrained IoT nodes
IECC (2020) [83]	ECC, SHA-512, XOR, Secret key	512 bits	Asymmetric, ECC	Statistical analysis shows the strength of this scheme
LNHASH (2019) [87]	Linear and nonlinear CA designed based on sponge construction	96 bits, 128 bits and 160 bits	Hash function	LNHASH extends the balance between speed, security, and implementation cost
A4 (2020) [84]	LFSR, FCSR, XOR, Boolean	126 bits Key	Stream cipher	A4 is entirely protected from algebraic attacks and shows resistance to brute-force and differential attacks
NLSC (2019) [85]	NFSR	80 bits Key	Stream cipher	This algorithm resists statistical attacks
DLP (2019) [45]	S-Box, Random Key		Block cipher	Reduces memory consumption, reduces time complexity

Key size: Key length or key size indicates the number of bits in a key that are used in cryptographic cipher techniques. The encryption strength depends on the complexity of the method, which prevents the discovery of the key. Encryption intensity is defined by the key size used in the encryption process. Longer keys deliver a more robust cipher, but on the other hand, they require more power and complexity. AES, LED, RECTANGLE and PRINCE use 128 bits in a key, which makes them unsuitable for resource-restricted devices. However, the Modified QARMA algorithm applies only a 64 bit key size for encryption purposes. SAT_Jo, Modified PRESENT, Piccolo and KTANTAN are suitable for IoT devices as they use an 80 bit key size. However, more power consumption tests need to be done for the modified PRESENT cipher before it is evaluated.

Table 6
Comparison of the different lightweight algorithms in terms of key length, Blocksize, Technology value, GE, Latency, Throughput, and the number of rounds.

Algorithm	Key size (bits)	Block size (bits)	Gate area (GE)	Technology value (μm)	No. of rounds	Latency (Cycle)	Throughput (Mbps)
AES [93,94]	128/192/256	128	2400	0.13	10/12/14	226	56
PRESENT [75,95]	80, 128	64	2195	0.13	31	31	206
HIGHT [93]	128	64	3048	0.13	32	34	188
KTANTAN [96]	80	64	688	0.13	12/16/20	255	25.1
LED [93,96]	128	64	1265	0.13	32	48	133.33
RECTANGLE [28]	128	64	1787	0.13	25	26	246
PRINCE [93,96]	128	64	3491	0.13	12	12	533
SIMON [97]	96	48	763	0.13	32/52/72	304	15.8
Piccolo [93]	80	64	1260	0.13	25	237.04	237
SFN [98]	96	64	1876	0.18	32	1876.04	200
SAT_Jo [72]	80	64	1167	0.13	31	1270	14.9
Modified PRESENT [79]	80	64	1884	0.13	25		
QARMA [71]	64	64	17 109	0.13	27	1	1705
Modified QARMA [71]	64	64	7844	0.13	27	27	2667

Block size: A block cipher operates on a constant sequence of bits. Block cipher cryptography uses the same size input and output block. Bigger block sizes require more CPU and battery power to secure the IoT network. Hence, a smaller block size algorithm is suitable for IoT end devices. According to the comparisons displayed in the algorithm table, AES uses the largest block size, which is 128 bits (16 bytes), to secure the network, and SIMON applies the lowest – a 46 bit block. However, most of the block ciphers consume 64 bits (8 bytes) in their block range.

Gate area and technology value: Lightweight cryptography is measured by the Gate Equivalent (GE), which indicates the physical area essential to execute the algorithm. A suitable lightweight primitive requires less gate area. According to the ISO/IEC standard [99], lightweight cryptographic cryptography should have a GE value from 1000 to 2000. Table 6 compares the lightweight ciphers in the IoT environment: KATANTAN, LED, RECTANGLE, SIMON, Piccolo, SFN, SAT-Jo, and Modified PRESENT. Power consumption is a critical consideration in deciding what cipher to use. Energy can be assessed on Gate value (GE) and subsequent CMOS technology value [96] when CMOS technology shifts from μm to nanometre (nm) gate intensity rises. Table 6 represents the corresponding technology value of 0.13 μm.

A number of rounds: A part of the key size, ciphers use round-based execution, which makes the cipher extra secure. A larger key size and more rounds make the system safer [100]. However, these systems use more energy and computational power. Cryptographic designing requires a decrease in the necessary iterations of rounds. As a result, this diminishes the required resources and latency, which are necessary to preserve the major functionality of IoT nodes [78]. The currently used typical ciphers are not fitted to these IoT nodes because many round repetitions are essential to achieve the preferred security [43]. AES, KTANTAN and PRINCE require fewer rounds, but the gate area is higher than the lightweight algorithms. SFN and Sat_Jo are lightweight algorithms though they use more rounds than AES. Piccolo is a suitable lightweight in terms of key length, number of rounds, gate area and throughput.

Latency: Latency is calculated as the time needed between the initial approach of encryption and when the encrypted output is generated [101]. Latency represents the number of cycles. Latency can be calculated as $L = K \times t_{cycle}$. L stands for latency; K stands for the number of clock cycles necessary to compute one block of cipher text, and t_{cycle} indicates the time in one cycle. Latency is critical for real-time applications, such as those for smart cities, smart transport, and smart grids, etc. IoT devices are resource-restricted heterogeneous devices with high latency, low energy, minimal computation capability, and low throughput [15]; therefore, the minimum achievable latency is required. Low computational complexity is an essential prerequisite for an effective cipher system to ensure low latency, which leads to lower power and resource overheads [78]. Consequently, it is necessary to design new cryptographic algorithms with low latency and low resource consumption. PRINCE, QARMA, and modified QARMA have the lowest latency, which indicates they are suitable for resource-constrained devices. However, according to Thangamani and Murugappan [70], further modification is needed for the QARMA algorithm. AFN, SAT_Jo, AES and Piccolo are some of the block ciphers with higher latency, making them vulnerable to several IoT attacks. KTANTAN and Piccolo show more latency, which makes these algorithms unsuitable for IoT devices.

Throughput: Throughput is measured by the number of bits transformed per second at a specific frequency throughout the encrypting and decrypting processes of the cipher algorithm. Throughput is expressed as the specified frequency. The following equation calculates throughput:

$$T = \frac{B \times F}{N}$$

where T stands for throughput, B is data size in bits, called block size, and F is frequency. N is the number of cycles per block. Throughput is higher in traditional cryptography; in contrast, several IoT applications expect moderate throughput [69]. SAT_Jo cipher facilitates high throughput and low latency. This algorithm needs less energy and less area compared to the PRESENT block cipher, which makes it suitable for the lightweight block cipher [73]. However, LED, SIMON, SFN, and SAT_Jo use more algorithms, which makes them expensive in terms of computational power. Most of the algorithms show vulnerability to various attacks.

Confusion and diffusion properties: Shanon's confusion and diffusion are the two fundamental properties to intensify the cipher [96]. Substitution box (S-box) provides confusion properties that facilitate keeping secret the relationship between ciphertext, key, and plaintext. Diffusion uses P-box and scatters the statistical structure of plaintext over the ciphertext, which hides the relationship between plaintext and ciphertext [102]. Static S-box stored in ROM. Hardware efficient algorithm PRESENT uses a single 4-bit S-box, although it uses larger cycles in software [93]. GIFT suggests a simpler version of S-Box with reduced physical space. It also uses a smaller number of rounds which offers higher throughput and a faster key schedule. GOST, a lightweight cipher developed by the Soviet Union government, adopted an S-box form PRESENT which requires 651 GE [103]. LBlock uses ten S-boxes; on the contrary, TWINE uses a single S-Box for its cipher [104]. An ultra-lightweight algorithm HIGHT does not use S-box with simple computational functions. The multiplicative inverse-based Halaka uses an 8-bit S-box with LFSR, which gives more security than PRESENT [93].

The trade-off between performance, cost, and security in the resource-constrained environment is depicted in Fig. 3. Kong

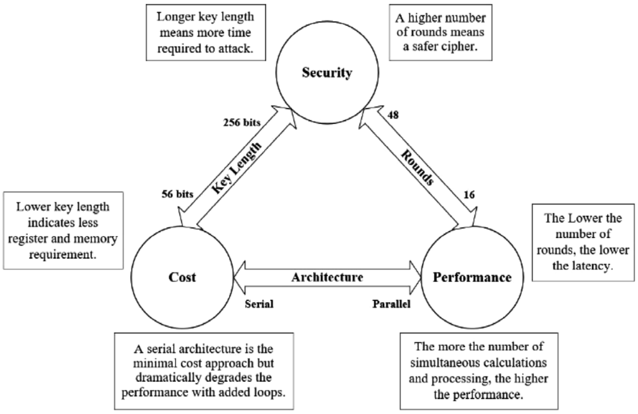
Longer key length means more time required to attack.
A higher number of rounds means a safer cipher.
Security
256 bits
Key Length
56 bits
48
Rounds
16
Lower key length indicates less register and memory requirement.
The Lower the number of rounds, the lower the latency.
Cost
Architecture
Performance
Serial
Parallel
A serial architecture is the minimal cost approach but dramatically degrades the performance with added loops.
The more the number of simultaneous calculations and processing, the higher the performance.

Fig. 3. Trade-off between cost, performance, and security.

[100] found a significant link between IoT devices' cost, performance, and security. The system speed is dynamically affected by the process platform. Compared to serial architecture, parallel arrangement enhances performance and decreases latency. Consequently, parallel structure and a lower number of cryptographic rounds increase performance. Moreover, the system cost is directly associated with the selection of algorithms and performance. More cost is necessary to develop a more robust and faster cipher [69].

System performance is primarily computed by the number of rounds that a cipher can execute. More rounds involve more computational requirements; hence latency increases [100]. Conventional block ciphers like AES use a multi-round composition with several round repetitions, which can be based on SPN or FN [78]. Feistel networks have a comparative advantage over SP networks, considering the encryption and decryption activities. SPN consumes more resources than the Feistel network structure. However, the security of SPN is comparatively higher because the round function can modify all block messages in a reiterative round [98].

The key length is directly related to the security of lightweight cryptographic network communication. The larger the key, the more secure the network; however, larger keys require more memory and CPU. Consequently, it makes the cipher unsuitable for resource-restricted devices. Table 6 shows the largest key has 256 bits, and the smallest has 64 bits. QARMA uses the smallest key size, which makes it a lightweight cipher. However, its gate area is higher than the standard lightweight cipher. On the other hand, SIMON and KTANTAN gates are in the ultra-low lightweight range, but the latency of KTANTAN is higher than the other ciphers.

Many new lightweight cipher algorithms have been proposed. Nevertheless, it is necessary to improve in the areas of security enhancement, decreasing latency, reducing energy consumption, lowering power consumption and chip area reduction. Different types of ciphers are facing various challenges. For example, the LCC shows resistance to various attacks. However, its key management strategies could be further developed. In contrast, G-TBSA consumes low energy but is only suitable for wireless sensor networks. None of the modern lightweight algorithms is secure enough for both block cipher and stream cipher.

Choosing an efficient and adequate number of S-boxes is essential to achieve an acceptable balance between performance and security [105]. Therefore, planning fast and straightforward but robust confusion and diffusion properties are critically important to balance cost, performance, and security. A smaller number of S-boxes need to be used to reduce memory consumption and computing power while providing adequate protection. For instance, PRESENT was motivated to design from AES by reducing the number of S-boxes from eight to one. However, creating a robust S-box using different confusion techniques while maintaining acceptable security and less overhead is still an interesting research question. This study has identified various issues that need to be addressed to develop a lightweight block and stream cipher algorithm.

Challenges in lightweight block cipher:

- Design a robust S-box and P-box.
- Implement a shorter key length.
- Develop a simple key structure.
- Generate simpler and fewer rounds of an algorithm.
- Use more frequent dynamic keys.
- Apply a smaller block of data.

Issues in lightweight stream cipher:

- Apply a smaller key size.
- Decrease the internal state.
- Reduce the chip area.

At present, we are working on substitution and permutation techniques and mainly focusing on the design of the S-box for lightweight cipher, which considers the balance of three main features: cost, performance, and security.

7. Conclusion

We have analysed contemporary research on lightweight cryptographic techniques used in IoT networks to keep data communication secure. Each algorithm has merits and demerits in terms of ensuring security while exchanging information in the IoT environment. Some algorithms demand more storage space but have fewer computational requirements and vice versa. Several algorithms are lightweight in terms of energy, computational power, and cost-effectiveness; however, they do not demonstrate resistance to various attacks. We found two types of algorithms according to the key arrangement: symmetric and asymmetric cryptography. Popular recently devised symmetric algorithms used in IoT security are block ciphers and stream ciphers; however, neither of these is ideal for securing resource-constrained communications in IoT systems. The security problem is a critical issue of the IoT, which has not been appropriately addressed in contemporary research on network protection. A lightweight cryptographic algorithm needs to be developed to secure resource-constrained IoT architecture. Growing attack patterns of IoT networks demand research on the improvement of lightweight ciphers. Future research could benefit by focusing on reducing key size, using a more frequent dynamic key, decreasing block size, introducing more straightforward rounds and designing simple key schedules for lightweight block cipher development. Internal state, minimising key size, and initialising vector are some of the prime objectives to develop future lightweight stream ciphers.

CRediT authorship contribution statement

Muhammad Rana: Conceived the model and the conceptual framework, Analysed the data, Developed the theory and investigation, Contributed to the interpretation of the results, Provided critical feedback and helped shape the research, analysis, and manuscript. **Quazi Mamun:** Analysed the data, Verified the analytical methods, Supervised the findings of this work, Provided critical feedback and helped shape the research, analysis, and manuscript. **Rafiqul Islam:** Verified the analytical methods, Supervised the findings of this work, Provided critical feedback and helped shape the research, analysis, and manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Hameed, A. alomary, Security issues in IoT: A survey, in: 2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), IEEE, 2019, http://dx.doi.org/10.1109/3ICT.2019.8910320.

[2] J. Lee, J. Kim, J. Seo, Cyber attack scenarios on smart city and their ripple effects, in: International Conference on Platform Technology and Service (PlatCon), IEEE, 2019, http://dx.doi.org/10.1109/PlatCon.2019.8669431.

[3] World Urbanization Prospects: The 2018 Revision, Department of Economic and Social Affairs, The United Nations, United Nations, New York, USA, 2018.

[4] Y. Li, Y. Lin, S. Geertman, The development of smart cities in China, in: 14th International Conference of Computer, Urban Planning and Urban Management, 2015, pp. 7–10.

[5] L. Cui, G. Xie, Y. Qu, L. Gao, Y. Yang, Security and privacy in smart cities: Challenges and opportunities, IEEE Access 6 (2018) 46134–46145, http://dx.doi.org/10.1109/access.2018.2853985, IEEE Access.

[6] A. Gissinga, M. Timmsa, S. Browninga, R. Cromptona, J. McAneney, Compound natural disasters in Australia: a historical analysis, Environ. Hazards (2021) http://dx.doi.org/10.1080/17477891.2021.1932405, Taylor & Francis.

[7] K. Demestichas, N. Peppes, T. Alexakis, Survey on security threats in agricultural IoT and smart farming, Sensors (2020) http://dx.doi.org/10.3390/s20226458, MDPI.

[8] M. Gupta, M. Abdelsalam, S. Khorsandroo, S. Mittal, Security and privacy in smart farming: Challenges and opportunities, IEEE Access 8 (2020) 34564–34584, http://dx.doi.org/10.1109/ACCESS.2020.2975142, Art no. 9003290.

[9] M.M. Jahna, et al., Cyber Risk and Security Implications in Smart Agriculture and Food Systems, White Paper, Jahn Research Group, University of Wisconsin–Madison, College of Agriculture and Life Sciences, 2019.

[10] J. Laufs, E. Borrion, B. Bradford, Security and the smart city: A systematic review, Sustainable Cities Soc. 55 (2020) http://dx.doi.org/10.1016/j.scs.2020.102023, Elsevier Scince Direct.

[11] X. Jiang, M. Lora, S. Chattopadhyay, An experimental analysis of security vulnerabilities in industrial IoT devices, ACM Trans. Internet Technol. (2020) http://dx.doi.org/10.1145/3379542, ACM Digital Library.

[12] Y. Yang, L. Wu, G. Yin, L. Li, H. Zhao, A survey on security and privacy issues in internet-of-things, IEEE Internet Things J. 4 (5) (2017) 1250–1258, http://dx.doi.org/10.1109/JIOT.2017.2694844, IEEE.

[13] M.B.M. Noor, W.H. Hassan, Current research on Internet of Things (IoT) security: A survey, Comput. Netw. 148 (2019) 283–294, http://dx.doi.org/10.1016/j.comnet.2018.11.025, Elsevier.

[14] V. Rao, K.V. Prema, Comparative study of lightweight hashing functions for resource constrained devices of IoT, in: 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), IEEE, 2019, http://dx.doi.org/10.1109/CSITSS47250.2019.9031038.

[15] S. Roy, U. Rawat, J. Karjee, A lightweight cellular automata based encryption technique for IoT applications, IEEE Access 7 (2019) 39782–39793, http://dx.doi.org/10.1109/ACCESS.2019.2906326, IEEE Access.

[16] R. Yugha, S. Chithra, A survey on technologies and security protocols: Reference for future generation IoT, J. Netw. Comput. Appl. 169 (2020) http://dx.doi.org/10.1016/j.jnca.2020.102763, Elsevier.

[17] F.A. Alabaa, M. Othmana, I.A.T. Hashema, F. Alotaibi, Internet of Things security: A survey, J. Netw. Comput. Appl. 88 (2017) 10–28, http://dx.doi.org/10.1016/j.jnca.2017.04.002, Elsevier.

[18] S.F. Ahmed, M.R. Islam, T.D. Nath, B.J. Ferdosi, A.S.M.T. Hasan, G-TBSA: A generalized lightweight security algorithm for IoT, in: 2019 4th International Conference on Electrical Information and Communication Technology (EICT), IEEE, 2020, http://dx.doi.org/10.1109/EICT48899.2019.9068848.

[19] Q. Mamun, A qualitative comparison of different logical topologies for wireless sensor networks, Sensors (2012) http://dx.doi.org/10.3390/s121114887, Sensors.

[20] A. Lepekhin, A. Borremans, I. Ilin, S. Jantunen, A systematic mapping study on internet of things challenges, in: IEEE/ACM 1st International Workshop on Software Engineering Research & Practices for the Internet of Things (SERP4IoT), IEEE Digital Library, 2019, http://dx.doi.org/10.1109/SERP4IoT.2019.00009.

[21] N.A. Gunathilake, W.J. Buchanan, R. Asif, Next generation lightweight cryptography for smart IoT devices: Implementation, challenges and applications, in: IEEE 5th World Forum on Internet of Things (WF-IoT), IEEE, 2019, http://dx.doi.org/10.1109/WF-IoT.2019.8767250.

[22] V. Adat, B.B. Gupta, Security in Internet of Things: issues, challenges, taxonomy, and architecture, Telecommun. Syst. 67 (3) (2018) 423–441, http://dx.doi.org/10.1007/s11235-017-0345-9.

[23] K.-M. Chew, S.C.-W. Tan, G.C.-W. Loh, N. Bundan, S.-P. Yiiong, IoT soil moisture monitoring and irrigation system development, in: ICSCA 2020: Proceedings of the 2020 9th International Conference on Software and Computer Applications, ACM Digital Library, 2020, 247–252.

[24] S. Zeadallya, A.K. Das, N. Sklavos, Cryptographic technologies and protocol standards for Internet of Things, Internet Things (2019) http://dx.doi.org/10.1016/j.iot.2019.100075, Elsevier.

[25] M.A. Philip, Vaithiyanathan, A survey on lightweight ciphers for IoT devices, in: Presented at the International Conference on Technological Advancements in Power and Energy (TAP Energy), 2017.

[26] A.R. Sfar, E. Natalizio, Y. Challal, Z. Chtourou, A roadmap for security challenges in the Internet of Things, Digit. Commun. Netw. 4 (2) (2018) 118–137, http://dx.doi.org/10.1016/j.dcan.2017.04.003, Science Direct.

[27] R. Hamzaab, Z. Yancd, K. Muhammade, P. Bellavistaf, F. Titouna, A privacy-preserving cryptosystem for IoT E-healthcare, Inform. Sci. 527 (2020) 493–510, http://dx.doi.org/10.1016/j.ins.2019.01.070, Elsevier.

[28] B.S. Sumit Singh Dhanda, Poonam Jindal, Lightweight cryptography: A solution to secure IoT, Wirel. Pers. Commun. (2020) http://dx.doi.org/10.1007/s11277-020-07134-3, Springer.

[29] V. Varadharajan, U. Tupakula, K. Karmakar, Study of Security Attacks Against IoT Infrastructures, Technical Report TR1: ISIF ASIA Funded Project, 2018.

[30] M. Mahbub, Progressive researches on IoT security: An exhaustive analysis from the perspective of protocols, vulnerabilities, and preemptive architectonics, J. Netw. Comput. Appl. 168 (2020) http://dx.doi.org/10.1016/j.jnca.2020.102761, Elsiever.

[31] S.N. Swamy, D. Jadhav, N. Kulkarni, Security threats in the application layer in IOT applications, in: International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), IEEE, 2017, http://dx.doi.org/10.1109/I-SMAC.2017.8058395.

[32] A. Aggarwal, W. Asif, H. Azam, M. Markovic, M. Rajarajan, P. Edwards, User privacy risk analysis for the internet of things, in: Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), IEEE, 2019, http://dx.doi.org/10.1109/IOTSMS48152.2019.8939265.

[33] I. Makhdoom, M. Abolhasan, H. Abbas, W. Ni, Blockchain's adoption in IoT: The challenges, and a way forward, J. Netw. Comput. Appl. 88 (2018) 10–28, http://dx.doi.org/10.1016/j.jnca.2018.10.019, Elsevier.

[34] B.-C. Chifor, I. Bica, V.-V. Patriciu, Mitigating DoS attacks in publish–subscribe IoT networks, in: 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), IEEE, 2017, http://dx.doi.org/10.1109/ECAI.2017.8166463.

[35] M. Saadeh, A. Sleit, K.E. Sabri, W. Almobaideen, Hierarchical architecture and protocol for mobile object authentication in the context of IoT smart cities, J. Netw. Comput. Appl. 121 (2018) 1–19, http://dx.doi.org/10.1016/j.jnca.2018.07.009, Elsevier.

[36] H.P. Alahari, S.B. Yelavarthi, Performance analysis of denial of service DoS and distributed DoS attack of application and network layer of IoT, in: Third International Conference on Inventive Systems and Control (ICISC), IEEE, 2019, http://dx.doi.org/10.1109/ICISC44355.2019.9036403.

[37] F.A. Bakhtiar, E.S. Pramukantoro, H. Nihri, A lightweight IDS based on J48 algorithm for detecting DoS attacks on IoT middleware, in: IEEE 1st Global Conference on Life Sciences and Technologies (LifeTech), IEEE, 2019, http://dx.doi.org/10.1109/LifeTech.2019.8884057.

[38] H.A. Khattak, M.A. Shah, S. Khan, I. Ali, M. Imran, Perception layer security in Internet of Things, Future Gener. Comput. Syst. 100 (2019) 144–164, http://dx.doi.org/10.1016/j.future.2019.04.038, Elsiver.

[39] M.M. Nasralla, I. Garcia-Magarino, J. Lloret, Defenses against perception-layer attacks on IoT smart furniture for impaired people, IEEE Access 8 (2020) 119795–119805, http://dx.doi.org/10.1109/ACCESS.2020.3004814, IEEE.

[40] Y.M. Tukur, Y.S. Ali, Demonstrating the effect of insider attacks on perception layer of internet of things (IoT) systems, in: 15th International Conference on Electronics, Computer and Computation (ICECCO), IEEE, 2019, http://dx.doi.org/10.1109/ICECCO48375.2019.9043248.

[41] R. Kanagavelu, K.M.M. Aung, A survey on SDN based security in internet of things, Adv. Intell. Syst. Comput. 887 (2019) 563–577.

[42] V. Prakash, A.V. Singh, S.K. Khatri, A new model of light weight hybrid cryptography for internet of things, in: 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE, 2019, http://dx.doi.org/10.1109/ICECA.2019.8821924.

[43] H. Noura, R. Couturier, C. Pham, A. Chehab, Lightweight stream cipher scheme for resource-constrained IoT devices, in: 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2019, http://dx.doi.org/10.1109/WiMOB.2019.8923144.

[44] A.K. Mishra, A.K. Tripathy, D. Puthal, L.T. Yang, Analytical model for sybil attack phases in internet of things, IEEE Internet Things J. 6 (1) (2019) 379–387, http://dx.doi.org/10.1109/JIOT.2018.2843769, IEEE.

[45] A.P.R.d. Silva, M.H.T. Martins, B.P.S. Rocha, A.A.F. Loureiro, L.B. Ruiz, H.C. Wong, Decentralized intrusion detection in wireless sensor networks, in: 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks, ACM Digital Library, 2005, pp. 16–23, http://dx.doi.org/10.1145/1089761.1089765.

[46] W. Yun, W. Xiaodong, X. Bin, W. Demin, D.P. Agrawal, Intrusion detection in homogeneous and heterogeneous wireless sensor networks, IEEE Trans. Mob. Comput. 7 (6) (2008) 698–711, http://dx.doi.org/10.1109/tmc.2008.19, IEEE.

[47] Q. Jing, A.V. Vasilakos, J. Wan, J. Lu, D. Qiu, Security of the Internet of Things: Perspectives and challenges, Wirel. Netw. 20 (8) (2014) 2481–2501, http://dx.doi.org/10.1007/s11276-014-0761-7, Springer.

[48] S.C.-H. Huang, D.-Z. Du, New constructions on broadcast encryption key pre-distribution schemes, in: IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Vol. 1, IEEE Xplore, 2005, http://dx.doi.org/10.1109/INFCOM.2005.1497919.

[49] F.M. Al-Turjmana, A.E. Al-Fagihac, W.M. Alsalihb, H.S. Hassanein, A delay-tolerant framework for integrated RSNs in IoT, Comput. Commun. 36 (9) (2013) 998–1010, http://dx.doi.org/10.1016/j.comcom.2012.07.001, Elsevier.

[50] M. Rana, Q. Mamun, A robust and lightweight key management protocol for WSNs in distributed IoT applications, Int. J. Syst. Softw. Secur. Prot. (IJSSSP) 9 (4) (2018) http://dx.doi.org/10.4018/IJSSSP.2018100101, IGI Global.

[51] H. Chan, A. Perrig, PIKE: peer intermediaries for key establishment in sensor networks, in: IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE Xplore, 2005, http://dx.doi.org/10.1109/INFCOM.2005.1497920.

[52] S.M. Tahsien, H. Karimipour, P. Spachos, Machine learning based solutions for security of Internet of Things (IoT): A survey, J. Netw. Comput. Appl. 161 (2020) http://dx.doi.org/10.1016/j.jnca.2020.102630, Elsevier.

[53] K. Gafurov, T.M. Chung, Comprehensive survey on internet of things, architecture, security aspects, applications, related technologies, economic perspective, and future directions, J. Inf. Process. Syst. 15 (4) (2019) 797–819, http://dx.doi.org/10.3745/JIPS.03.0125.

[54] A. Tewari, B.B. Gupta, Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework, Future Gener. Comput. Syst. 108 (2020) 909–920, http://dx.doi.org/10.1016/j.future.2018.04.027, Elsevier.

[55] V.-L. Nguyen, P.-C. Lin, R.-H. Hwang, Energy depletion attacks in low power wireless networks, IEEE Access 7 (2019) 51915–51932, http://dx.doi.org/10.1109/ACCESS.2019.2911424, IEEE Access.

[56] V. Vujović, M. Maksimović, Raspberry Pi as a Sensor Web node for home automation, Comput. Electr. Eng. 44 (2015) 153–171, http://dx.doi.org/10.1016/j.compeleceng.2015.01.019, ACM Digital Library.

[57] T. Kafer, S.R. Bader, L. Heling, R. Manke, A. Harth, Exposing internet of things devices via REST and linked data interfaces, in: 2nd Workshop Semantic Web Technologies for the Internet of Things, Semantic Scholar, 2017.

[58] S. Bansal, D. Kumar, IoT ecosystem: A survey on devices, gateways, operating systems, middleware and communication, Int. J. Wirel. Inf. Netw. (2020) http://dx.doi.org/10.1007/s10776-020-00483-7, Springer.

[59] S. Huh, S. Cho, S. Kim, Managing IoT devices using blockchain platform, in: 19th International Conference on Advanced Communication Technology (ICACT), IEEE, 2017, http://dx.doi.org/10.23919/ICACT.2017.7890132.

[60] E. Baccelli, et al., RIOT: an open-source operating system for low-end embedded devices in the IoT, IEEE Internet Things J. 5 (6) (2018) 4428–4440, http://dx.doi.org/10.1109/JIOT.2018.2815038, IEEE.

[61] D. Zhai, R. Zhang, L. Cai, B. Li, Y. Jiang, Energy-efficient user scheduling and power allocation for NOMA-based wireless networks with massive IoT devices, 2018.

[62] F. Shaikh, E. Bou-Harb, N. Neshenko, A.P. Wright, N. Ghani, Internet of malicious things: Correlating active and passive measurements for inferring and characterizing internet-scale unsolicited IoT devices, IEEE Commun. Mag. 56 (9) (2018) 170–177, http://dx.doi.org/10.1109/MCOM.2018.1700685, IEEE.

[63] M.A.R. Shuman, et al., Establishing groups of internet of things (IOT) devices and enabling communication among the groups of IOT devices, 2017.

[64] K. Fysarakis, G. Hatzivasilis, K. Rantos, A. Papanikolaou, C. Manifavas, Embedded systems security challenges, in: Measurable Security for Embedded Computing and Communication Systems (MeSeCCS 2014), Research Gate, 2014, http://dx.doi.org/10.5220/0004901602550266.

[65] C. Manifavas, G. Hatzivasilis, K. Fysarakis, Y. Papaefstathiou, A survey of lightweight stream ciphers for embedded systems, Secur. Commun. Netw. 9 (2015) 1226–1246, http://dx.doi.org/10.1002/sec.1399, Wiley Online Library.

[66] A. Poschmann, Lightweight Cryptography - Cryptographic Engineering for a Pervasive World, Ruhr-University Bochum, 2009.

[67] C. Rolfes, A. Poschmann, G. Leander, C. Paar, Ultra-lightweight implementations for smart devices – security for 1000 gate equivalents, in: International Conference on Smart Card Research and Advanced Applications, 2008, pp. 89–103.

[68] R. Roman, C. Alcaraz, J. Lopez, A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes, in: Mobile Networks and Applications, Vol. 12, Springer Link, 2007, pp. 231–244, http://dx.doi.org/10.1007/s11036-007-0024-2.

[69] R. Kousalya, G.A.S. Kumar, A survey of light-weight cryptographic algorithm for information security and hardware efficiency in resource constrained devices, in: International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), IEEE, 2019, http://dx.doi.org/10.1109/ViTECoN.2019.8899376.

[70] N. Thangamani, M. Murugappan, A lightweight cryptography technique with random pattern generation, Wirel. Pers. Commun. (2019) 1409–1432, http://dx.doi.org/10.1007/s11277-018-6092-8, Springer.

[71] C. Zhao, Y. Yan, W. Li, An efficient ASIC implementation of QARMA lightweight algorithm, in: 2019 IEEE 13th International Conference on ASIC (ASICON), IEEE, 2020, http://dx.doi.org/10.1109/ASICON47005.2019.8983618.

[72] M.J.R. Shantha, L. Arockiam, SAT_Jo: An enhanced lightweight block cipher for the internet of things, in: 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, 2019, http://dx.doi.org/10.1109/ICCONS.2018.8663068.

[73] S.M.J. R, A. L, S.K. Malarchelvi, Security analysis of SAT_Jo lightweight block cipher for data security in healthcare IoT, in: ICCBDC 2019: Proceedings of the 2019 3rd International Conference on Cloud and Big Data Computing, 2019, pp. 111–116, http://dx.doi.org/10.1145/3358505.3358527.

[74] T.T.K. Hue, T.M. Hoang, D. Tran, Chaos-based S-box for lightweight block cipher, in: IEEE Fifth International Conference on Communications and Electronics (ICCE), IEEE, 2014, http://dx.doi.org/10.1109/CCE.2014.6916765.

[75] Z.M.J. Kubba, H.K. Hoomod, A hybrid modified lightweight algorithm combined of two cryptography algorithms PRESENT and salsa20 using chaotic system, in: 2019 International Conference of Computer and Applied Sciences (1st CAS2019), IEEE, 2019, http://dx.doi.org/10.1109/CAS47993.2019.9075488.

[76] W.-L. Cho, K.-B. Kim, K.-W. Shin, A hardware design of ultra-lightweight block cipher algorithm PRESENT for IoT applications, J. Korea Inst. Inf. Commun. Eng. 20 (7) (2016) http://dx.doi.org/10.6109/jkiice.2016.20.7.1296.

[77] E. Lara, L. Aguilar, J.A. García, M.A. Sanchez, A lightweight cipher based on salsa20 for resource-constrained IoT devices, Sensors (2018) http://dx.doi.org/10.3390/s18103326.

[78] H. Noura, A. Chehab, L. Sleem, M. Noura, R. Couturier, M.M. Mansour, One round cipher algorithm for multimedia IoT devices, Multimedia Tools Appl. (2018) http://dx.doi.org/10.1007/s11042-018-5660-y, Springer Link.

[79] R. Chatterjee, R. Chakraborty, A modified lightweight PRESENT cipher for IoT security, in: 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), IEEE, 2020, http://dx.doi.org/10.1109/ICCSEA49143.2020.9132950.

[80] H. Noura, A. Chehab, R. Couturier, Lightweight dynamic key-dependent and flexible cipher scheme for IoT devices, in: 2019 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2019, http://dx.doi.org/10.1109/WCNC.2019.8885976.

[81] R.R.K. Chaudhary, K. Chatterjee, An efficient lightweight cryptographic technique for IoT based E-healthcare system, in: 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, 2020, http://dx.doi.org/10.1109/SPIN48934.2020.9071421.

[82] E. Gyamf, J.A. Ansere, L. Xu, ECC based lightweight cybersecurity solution for IoT networks utilising multi-access mobile edge computing, in: Fourth International Conference on Fog and Mobile Edge Computing (FMEC), IEEE, 2019, http://dx.doi.org/10.1109/FMEC.2019.8795315.

[83] M.A. Khan, M.T. Quasim, N.S. Alghamdi, M.Y. Khan, A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data, IEEE Access 8 (2020) 52018–52027, http://dx.doi.org/10.1109/ACCESS.2020.2980739.

[84] N.A. Mohandas, A.A.R. Swathi, A. Nazar, G. Sharath, A4: A lightweight stream cipher, in: 5th International Conference on Communication and Electronics Systems (ICCES), IEEE, 2020, http://dx.doi.org/10.1109/ICCES48766.2020.9138048.

[85] L. Ding, C. Liu, Y. Zhang, Q. Ding, A new lightweight stream cipher based on chaos, Symmetry 11 (7) (2019) http://dx.doi.org/10.3390/sym11070853, MDPI.

[86] S. Thapliyal, H. Gupta, S.K. Khatri, An innovative model for the enhancement of IoT device using lightweight cryptography, in: 2019 Amity International Conference on Artificial Intelligence (AICAI), IEEE, 2019, http://dx.doi.org/10.1109/AICAI.2019.8701377.

[87] X. Zhang, Q. Xu, X. Li, C. Wang, A lightweight hash function based on cellular automata for mobile network, in: Presented at the 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN), 2019.

[88] J. Guo, T. Peyrin, A. Poschmann, The PHOTON Family of Lightweight Hash Functions, in: Advances in Cryptology – CRYPTO 2011, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 222–239.

[89] Y. Huang, S. Li, W. Sun, X. Dai, W. Zhu, HVH: A Lightweight Hash Function Based on Dual Pseudo-Random Transformation, in: Security, Privacy, and Anonymity in Computation, Communication, and Storage, Springer International Publishing, Cham, 2021, pp. 492–505.

[90] BahramRashidi, Efficient full data-path width and serialized hardware structures of SPONGENT lightweight hash function, Microelectron. J. 115 (2021) http://dx.doi.org/10.1016/j.mejo.2021.105167, Elsevier.

[91] G. Saldamli, L. Ertaul, A. Shankaralingappa, Analysis of lightweight message authentication codes for IoT environments, in: Presented at the Fourth International Conference on Fog and Mobile Edge Computing (FMEC), 2019.

[92] A.D. Dwivedi, Security analysis of lightweight IoT cipher: Chaskey, Cryptography 4 (3) (2020) 22.

[93] V.A. Thakor, M.A. Razzaque, M.R.A. Khandaker, Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities, IEEE Access 9 (2021) 28177–28193, http://dx.doi.org/10.1109/ACCESS.2021.3052867, IEEE.

[94] Q. Li, C. Zhong, K. Zhao, X. Mei, X. Chu, Implementation and analysis of AES encryption on GPU, in: IEEE 14th International Conference on High Performance Computing and Communication & IEEE 9th International Conference on Embedded Software and Systems, IEEE, 2012, pp. 843–848, http://dx.doi.org/10.1109/HPCC.2012.119.

[95] S.S. Dhanda, B. Singh, P. Jindal, Lightweight cryptography: A solution to secure IoT, Wirel. Pers. Commun. 112 (3) (2020) 1947–1980, http://dx.doi.org/10.1007/s11277-020-07134-3.

[96] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, C. Manifavas, A review of lightweight block ciphers, J. Cryptogr. Eng. (2018) 141–184, http://dx.doi.org/10.1007/s13389-017-0160-y, Springer Link.

[97] B. Rashidi, Flexible structures of lightweight block ciphers PRESENT, SIMON and LED, IET Circuits Devices Syst. 14 (3) (2020) 369–380, http://dx.doi.org/10.1049/iet-cds.2019.0363, IEEE.

[98] L. Li, B. Liu, Y. Zhou, Y. Zou, SFN: A new lightweight block cipher, Microprocess. Microsyst. 60 (2018) 138–150, http://dx.doi.org/10.1016/j.micpro.2018.04.009.

[99] C. Pei, Y. Xiao, W. Liang, X. Han, Trade-off of security and performance of lightweight block ciphers in Industrial Wireless Sensor Networks, EURASIP J. Wireless Commun. Networking (2018) http://dx.doi.org/10.1186/s13638-018-1121-6, Springer Nature.

[100] J.H. Kong, L.-M. Ang, K.P. Seng, A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments, J. Netw. Comput. Appl. (2015).

[101] A. Hodjat, I. Verbauwhede, Area-throughput trade-offs for fully pipelined 30 to 70 Gbits/s AES processors, IEEE Trans. Comput. 55 (4) (2006) 366–372, http://dx.doi.org/10.1109/TC.2006.49, IEEE.

[102] M. Rana, Q. Mamun, R. Islam, An S-box design using irreducible polynomial with affine transformation for lightweight cipher, in: Presented at the EAI QSHINE 2021-17th EAI International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, Melbourne, Australia, 2021.

[103] A. Poschmann, S. Ling, H. Wang, 256 Bit Standardized Crypto for 650 GE – GOST Revisited, in: Cryptographic Hardware and Embedded Systems, CHES 2010, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 219–233.

[104] T. Suzaki, K. Minematsu, S. Morioka, E. Kobayashi, TWINE: A lightweight, versatile block cipher, in: Presented at the ECRYPT Workshop on Lightweight Cryptography, 2011.

[105] G. Piret, T. Roche, C. Carlet, PICARO – A Block Cipher Allowing Efficient Higher-Order Side-Channel Resistance, in: Applied Cryptography and Network Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 311–328.

Muhammad Rana is a Ph.D. candidate at Charles Sturt University, Australia, with a particular interest in the Internet of Things, Cryptographic Algorithm and Network Security. He currently works on resource-constrained IoT devices, the security problem of IoT network, and simulation techniques in a different lightweight algorithm. Muhammad received BC in Computer Science at Charles sturt University. He received his Master degree from Federation University. Authentication, security algorithms of IoT and wireless communication is his research curiosity.

Dr Mamun is a Senior Lecturer of Computing in the School of Computing and Mathematics, Faculty of Business, Justice and Behavioural Sciences, Charles Sturt University. He earned a B.Sc. Engineering degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), a Masters degree (by research) in Global Information and Telecommunication Studies from Waseda University Japan, and a Ph.D. degree with a specialisation in distributed computing from Monash University, Australia. Before joining CSU, Quazi has worked as a sessional academic and guest Lecturer in the Faculty of Information Technology of Monash University. Quazi's research interests include, but not limited to, distributed systems, ad hoc and sensor networks, wireless networks, privacy and security in information networks. He is an active member of the Advanced Networks Research Lab (ANRL) and ICT Security Group of Charles Sturt University.

Dr Rafiqul Islam is working as an Associate Professor at the School of Computing and Mathematics, Charles Sturt University, Australia. Dr Islam has a strong research background in cybersecurity, focusing on malware analysis and classification, Authentication, security in the cloud, privacy in social media, IoT and Dark Web. He led the Cybersecurity research team and has developed a strong background in leadership, sustainability, collaborative research in the area. He has a strong publication record and has published more than 170 peer-reviewed research papers, book chapters and books. Dr Islam is the associate editor of the International Journal of Computers and Applications and guest editors of various reputed journals. He is the senior member of IEEE.