A Review of Lightweight Security and Privacy for Resource-Constrained IoT Devices

Kumar, Sunil; Kumar, Dilip; Dangi, Ramraj; Choudhary, Gaurav; Dragoni, Nicola; You, Ilsun

Published in:
Computers, Materials and Continua

Link to article, DOI:
10.32604/cmc.2023.047084

Publication date:
2024

Document Version
Publisher's PDF, also known as Version of record

Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.047084

REVIEW

A Review of Lightweight Security and Privacy for Resource-Constrained IoT Devices

Sunil Kumar[1], Dilip Kumar[1], Ramraj Dangi[2], Gaurav Choudhary[3], Nicola Dragoni[4] and Ilsun You[5,*]

[1]Department of Computer Science and Engineering, National Institute of Technology Jamshedpur, Jamshedpur, India

[2]School of Computing Science and Engineering, VIT University Bhopal, Bhopal, 466114, India

[3]Center for Industrial Software, The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Sonderborg, Odense, Denmark

[4]Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, 2800, Denmark

[5]Department of Information Security, Cryptology, and Mathematics, Kookmin University, Seoul, South Korea

*Corresponding Author: Ilsun You. Email: ilsunu@gmail.com

Received: 24 October 2023 Accepted: 11 December 2023 Published: 30 January 2024

ABSTRACT

The widespread and growing interest in the Internet of Things (IoT) may be attributed to its usefulness in many different fields. Physical settings are probed for data, which is then transferred via linked networks. There are several hurdles to overcome when putting IoT into practice, from managing server infrastructure to coordinating the use of tiny sensors. When it comes to deploying IoT, everyone agrees that security is the biggest issue. This is due to the fact that a large number of IoT devices exist in the physical world and that many of them have constrained resources such as electricity, memory, processing power, and square footage. This research intends to analyse resource-constrained IoT devices, including RFID tags, sensors, and smart cards, and the issues involved with protecting them in such restricted circumstances. Using lightweight cryptography, the information sent between these gadgets may be secured. In order to provide a holistic picture, this research evaluates and contrasts well-known algorithms based on their implementation cost, hardware/software efficiency, and attack resistance features. We also emphasised how essential lightweight encryption is for striking a good cost-to-performance-to-security ratio.

KEYWORDS

IoT; a sensor device; lightweight; cryptography; block cipher; smart card; security and privacy

1 Introduction

The Internet of Things (IoT) has become increasingly popular as a subject of study in recent years, primarily because of its potential applications in various domains, including smart transportation, logistics, homes, cities, healthcare, environment, infrastructure, Industry 4.0, agriculture, and many more. At the core of every IoT solution are the devices themselves [1]. IoT devices are embedded with sensors, processors, and actuators to sense, collect, transmit, process, and actuate data. They connect to other devices, networks, and services to enable data-driven automation and control. This data can be used to improve the efficiency and productivity of various industries and

applications [2]. IoT devices can communicate with each other and with other computing devices to exchange data and information. This data can be used to monitor, control, and optimize various operations, such as energy management, manufacturing processes, transportation, and logistics [3]. By gathering data from multiple sources, IoT devices can also enable automated decision- making, which increases efficiency and productivity across industries [4].

Fig. 1 shows that there are two types of IoT devices: resources abundant (servers, PCs, tablets, cellphones, and so on) and resources scarce (industrial sensors, RFID tags, actuators, etc.) [5]. The market will be overrun by the IoT, which will lead to an efficient data exchange rate between all parties. The IoT will flood the market, resulting in an effective data exchange rate across all parties [6].

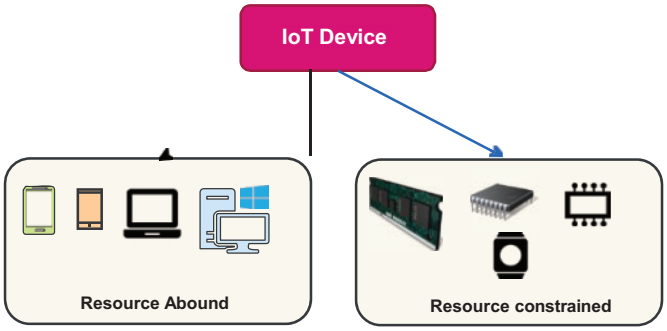

Figure 1: Two broad categories of IoT devices

Homomorphic encryption (HE) is a cryptographic technique that allows computations to be performed on encrypted data without requiring decryption [7]. Lightweight cryptography, on the other hand, focuses on developing cryptographic algorithms that are optimized for resource-constrained environments, such as IoT devices, embedded systems, and mobile devices, where computational power, memory, and energy are limited [8]. This can be a challenge for lightweight devices that have limited processing power and memory. However, researchers have been working on developing optimized versions of homomorphic encryption schemes that are better suited for resource-constrained environments [9]. These optimized schemes aim to reduce the computational overhead and memory requirements while still providing some level of homomorphic functionality [10].

1.1 Security Challenges and Security Requirements of Resource-Constrained IoT Devices

When the focus switches from servers to sensors, and billions of intelligent devices (connected devices) are functioning across many platforms, there is a multitude of never-before-seen issues for their stakeholders or consumers. A wide range of problems may be broken down into categories such as security, privacy, interoperability, lifespan, support, and technology [11]. Internet of Things devices are tempting targets for hackers because of the variety of security threats they face as a result of their proximity to and interaction with the real world. Because of their convenience and susceptibility, they make appealing targets [12,13].

There are use cases where it might be very challenging to keep IoT devices secure. Constant system updates, conformity to privacy and regulatory standards, availability, confidentiality, data integrity, authenti cation, and authorisation are just some of the many concerns that must be addressed [14]. Fig. 2 highlights the security problems associated with the IoT and the security requirements.

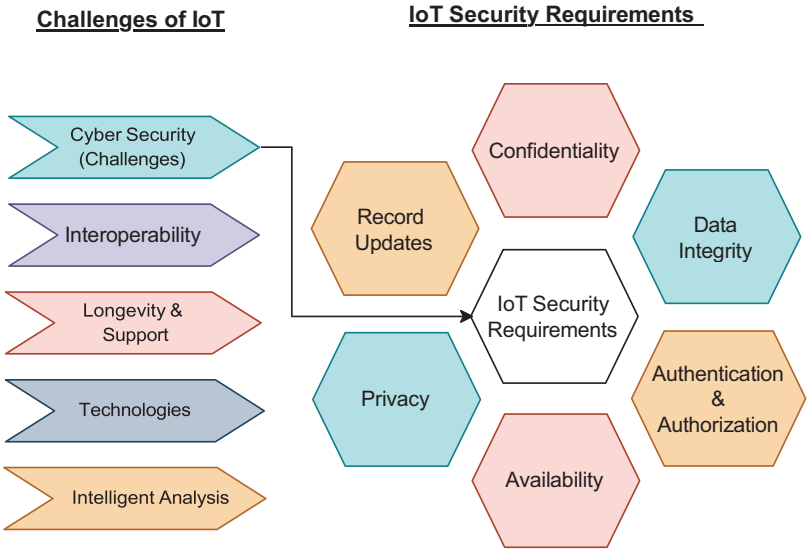

Figure 2: IoT Security requirement and challenges

In this section, various security issues concerning IoT devices are elaborated on. A list of essential security features that need to be considered includes:

- **Confidentiality:** All data exchanged through IoT communication channels must be heavily encrypted to prevent access by malicious actors. In addition, no information should be leaked to potential snoopers.
- **Data Integrity:** To prevent any tampering with data in transit by an unauthorised party, Data Integrity is an essential component of any large-scale centralised system. Content and data integrity at rest must also be safeguarded. For a broader definition of data freshness, see data integrity.
- **Authentication and Authorization:** Strong key base limitations and protocols should be used to authenticate any data traffic handled by IoT devices. Kerberos protocol is effective for providing security across numerous checks in a centralised approach to data transfer.
- **Availability:** IoT availability refers to the degree to which connected devices and infrastructure are functional, easily available, and trustworthy. Availability is a critical component of IoT security that ensures connected devices and infrastructure can be accessed and used as required.
- **Privacy:** Each device in a low-power network must be able to maintain its unique identity and prevent unauthorised access to sensitive data. The anonymity problem will be solved by enforcing privacy regulations.
- **Record Update:** Secure communication protocols, reliable update methods, and comprehensive test-ing are all necessary for effective record updates on IoT devices. Balancing the need to update devices with the need to do so in a secure manner is a delicate balancing act.

1.2 Significant Challenges Required to Implement Traditional Cryptographic Algorithms in Resource Con-Strained IoT Device

The primary challenges of deploying traditional cryptography in IoT devices (Fig. 3) [15].

- Low memory required (registers, RAM, ROM)
- Small computing power

- Tiny physical area is required for the design of the assembly
- Required less battery power
- Required real-time processing and quick response

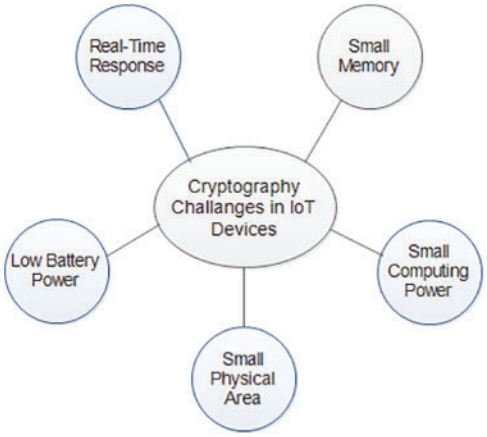

Figure 3: Challenges with traditional cryptography

IoT devices such as sensors and RFID tags are often small in size, have a limited amount of usable memory (RAM or ROM) for storing and running programs, and have limited processing power [5]. As well as limited physical space for the installation of the IoT devices [16].

IoT devices frequently struggle in real-time applications to strike a balance between speedy and precise answers while guaranteeing essential security measures within the constraints of available resources [17,18,19]. By using a subset of cryptographic methods that demand little memory, processing power, or energy, lightweight cryptography successfully addresses the aforementioned problems with traditional encryption [8]. Lightweight cryptography is not only applicable to devices with limited resources (e.g., RFID tags, sensors, etc.). Still, it may also be easily applied to devices with more resources, directly or indirectly.

1.3 Application of Lightweight Cryptography in IoT

Lightweight cryptography is being used in real-world Internet of Things applications where efficient and safe cryptographic solutions are needed for devices with limited resources [20]. Here are a few case studies showcasing how lightweight cryptography is applied in various IoT applications:

Smart Home Security: Lightweight cryptography is used in a smart home security system to encrypt data sent between various Internet of Things gadgets including smart locks, security cameras, and doorbell cameras. These gadgets often have little memory and processing capabilities. To provide safe remote access and data sharing, devices use lightweight cryptographic methods.

IIoT (Industrial IoT): In an industrial IoT setting, where sensors and actuators are placed in factories and supply chains, lightweight cryptography is vital for ensuring the confidentiality and integrity of data. The safe and effective functioning of industrial processes is ensured by using lightweight algorithms to safeguard data transfer and control systems.

Mobile Health Monitoring: Wearable health gadgets, such as fitness trackers and medical sensors, use lightweight cryptography. As these devices often operate on small batteries, protecting sensitive

medical information must rely on lightweight encryption. Users may now keep tabs on their health without worrying about compromising the privacy of their medical records.

Intelligent Power Grids: The transfer of data and control instructions between smart metres, grid equipment, and the centralised grid management system in smart grid systems is encrypted using lightweight cryptography. These low-power cryptographic systems protect data authenticity and privacy with little re-source drain.

Inventory Control: To ensure the safety of data sent between GPS-enabled tracking devices and the main monitoring platform, asset tracking systems utilise lightweight encryption. In this way, pre-cious assets may be tracked in real-time, and their whereabouts can be kept secret from prying eyes.

Linked Automobiles: Communications between the many Internet of Things (IoT) components in a car, such as the infotainment system, telematics, and sensors, are encrypted using lightweight cryptography in connected vehicle systems. These lightweight cryptographic methods help keep vehicle data secure and private without slowing down response times.

Intelligent Farming: Soil moisture sensors, drones, and controlled irrigation systems are only a few ex-amples of the Internet of Things devices that benefit from lightweight cryptography in precision agriculture. These gadgets provide information on the state of crops and their surrounding environments, while also saving electricity and protecting the privacy of crucial agricultural records.

Environmental Monitoring: Lightweight cryptography is employed in Internet of Things sensors for environmental monitoring applications in uninhabited areas like woods and seas. Even in low-power settings, these sensors may gather information about the weather, air quality, or animals, and send it on to a central monitoring system.

Management of Retail Stock: Radio-frequency identification (RFID) systems use lightweight encryp-tion to ensure the integrity of stock-tracking and anti-counterfeiting procedures in the retail industry. RFID tags employ lightweight cryptographic methods to prevent unauthorised access and ensure the integrity of the data they store.

Cyber-Enabled Urban Areas: Lightweight cryptography is used to protect data sent over the internet by smart city infrastructure, such as traffic lights, environmental sensors, and garbage collection trucks. These cryptography technologies improve municipal operations while protecting the confidentiality and security of resident data.

1.4 Our Contribution

Researchers recently proposed various LWC algorithms. Many studies have discovered security flaws in specific LWC algorithms, such as [21,22]. Several studies [23,24,25] compared software or hardware implementations of these algorithms on various platforms and under multiple scenarios. Most of these studies considered algorithms appropriate for specific fields or applications. These publications lack a holistic understanding of the suggested LWC algorithms regarding hardware software performance and cryptanal-ysis. In [26], the authors studied numerous LWC algorithms but did not give a comprehensive overview of their applications, costs (memory, area, battery, energy), or performance. It is challenging to compare attacks on various lightweight cryptography (LWC) techniques since there is not a thorough security com-parison across the attacks that target these algorithms. In this case, only throughput and speed measured in clock cycles per byte can be used to evaluate the performance of lightweight cryptography (LWC) methods. Not taken into account are other crucial characteristics, including memory utilization, gate area, power consumption, and energy

consumption. Furthermore, these algorithms compete constantly across several rounds. The aim is to determine which algorithm is better.

We explain the essential characteristics of lightweight cryptography algorithms (LWC) proposed by eminent research organisations in the cryptography area to aid comprehension [5,27]. The alignment of LWC algorithms with each criterion to meet each condition is also highlighted in Table 1, and key terms used in this paper are mentioned in Table 2.

Table 1: Characteristics of LWC

Characteristics of LWC	Description
Physical	Area (in Gate equivalent (GE), Lookup Tables (LUT), Logic Block, Less Memory, Less Energy (J)
Performance	Throughput (measured in cycles per byte), Power (measured in Watta) and Latency (per clock cycles)
Security	Ample amount of security (use low bits), Attack replicas, Side channel attack confirmation necessities.

Table 2: Abbreviations and key terms

Abbreviations	Full forms	Abbreviations	Full form
AES	Advanced encryption standard	LUT	Lookup tables
ARX	Add-rotate-XOR	NIST	National institute of standards and technology
CPU	Central processor unit	RAM	Random acces memory
FN	Feistel network	RFID	Radio frequency identification
FPGA	Field programmable gate array	ROM	Read only memory
GFN	Generalized feistel network	SPN	Substitution permutation network
GE	Gate equivalent	VHDL	Hardware description language
IoT	IoT	WSN	Wireless sensors networks

1.5 Organization of Paper

The rest of the paper is structured as follows: In Section 2, we then delve into the basics of LWC, including the structure of lightweight cryptographic algorithms, lightweight cryptographic primitives, NIST standards for lightweight cryptography, design strategies for lightweight cryptography, and performance metrics for hardware and software. Section 3 focuses on existing lightweight block cipher algorithms based on the described structure. It includes a comparison of the software and hardware performance of LWC algorithms and an analysis of cryptanalysis attacks on LWC algorithms.

Section 4 outlines future research challenges and directions, while Section 5 provides the concluding remarks.

2 Basics of Lightweight Cryptography

Cryptography is the only arguable solution a security expert has always had for transmitting information over secured or unsecured mediums. The theory of communication under secrecy systems first appeared at Bells Systems, which defines converting a set of plaintext information into a set of possible ciphers. A common encryption strategy is known as the symmetric encryption scheme. The block cipher operates in various modes to provide confidentiality approved by NIST. The process of the aforementioned system entails the sharing of a secret key between the many people engaged in private communication. The primary objective of an asymmetric encryption scheme is to allow the encryption algorithm to be public while keeping the key secret. Fig. 4 shows the transformation of information from source to destination via encryption and a secret key, followed by the reverse engineering process at the destination end using the same shared private key.

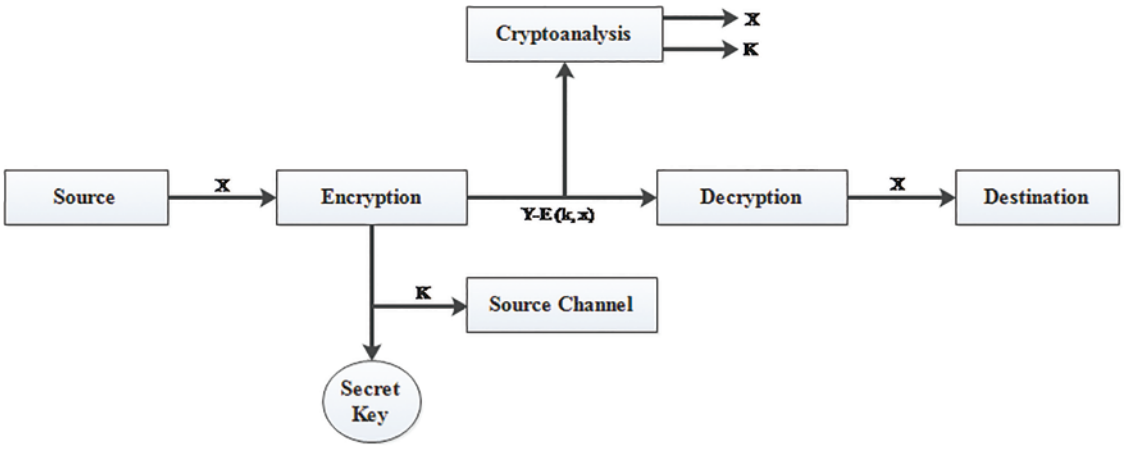

Figure 4: Model of a symmetric cryptosystem

2.1 Structure-Wise Lightweight Cryptographic Algorithms

Symmetric keys and asymmetric keys are the two basic types of cryptographic algorithms (Fig. 5). Symmetric keys use a single key for both encryption and decryption, in contrast to asymmetric cyphers, which use two different keys for each operation [27]. The only drawback of symmetric key encryption is the requirement to securely share the key between communicating parties without jeopardizing its confidentiality. This might be resolved by providing the key to a reliable third party in advance. This also assures data confidentiality, integrity, and authentication. Two sets of keys are essential for asymmetric cryptography. The recipient's public key is used to protect the confidentiality and integrity of the data, while the recipient's private key is used to authenticate the sender (as a digital signature) [28]. The recipient initially decrypts it with the sender's public key and then with his or her private key. Asymmetric encryption's main problem is its huge key, which adds significantly and decreases efficiency [24].

When using block ciphers, each block is encrypted and decrypted separately, whereas stream ciphers treat each input bit (or word) individually [28]. Claude Shannon [26,29] developed the properties of confusion and diffusion to strengthen cryptography. Confuse the ciphertext-key relationship

using substitution (S-box), whereas diffusion disperses the statistical structure of plaintext using permutation Â [26,28]. In comparison, stream ciphers rely on confusion, while block ciphers employ both confusion and diffusion. Block ciphers utilize XOR functions to encrypt data in a manner that is not easily reversible. On the other hand, hash functions are one-way mathematical transformations that convert an arbitrary input into a fixed-length, non-invertible bit string.

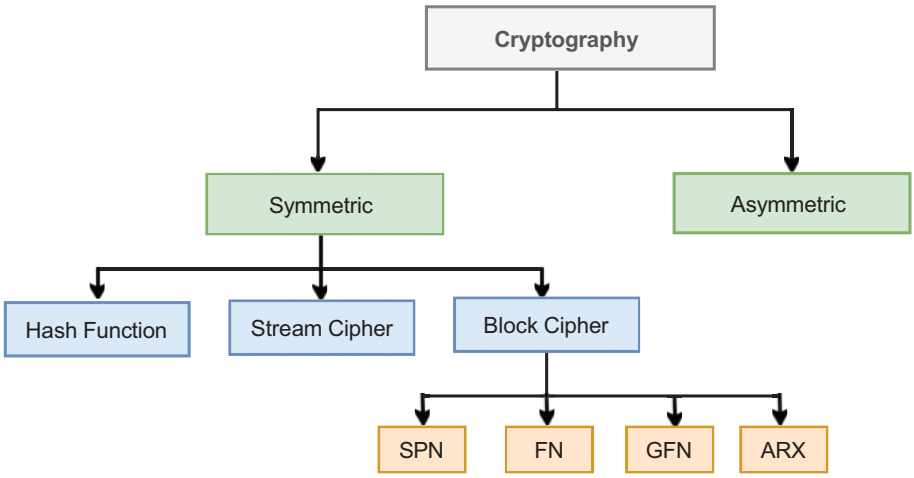

Figure 5: Structure wise classification of lightweight cryptography

Description of structure-wise lightweight cryptographic algorithms is shown in Table 3. The substitution-permutation network (SPN) modifies the data using permutation tables and substitution boxes to get it ready for the following cycle. A Feistel network (FN), on the other hand, splits the input block in half and applies diffusion to one-half every round. Each round begins with a switch between the two halves. The number of Feistel functions used determines how each pair of sub-blocks behaves in the Generalised Feistel Network (GFN), a version of the Feistel network [30]. Asymmetric encryption and decryption (ARX) operations, on the other hand, do not need an S-box. Despite being quick and simple to create, ARX lacks several security characteristics when compared to SPN and Feistel cyphers.

Table 3: Structure wise lightweight cryptographic

Symmetric cryptography structure	LWC algorithms
SPN	AES, SKINN, PRESENT, RECTANGLE, Midori, mCrypton, Nokeon, Prince, Pride, Print, Klein, MANTIS, LED
Feistel network	KASUMI, ITUbee, LBLOCK, MISTY, Roadrunner, TEA/XTEA/XXTEA/, Few, Simon
ARX	QTL, RC5, CHAM, CHASKEY, Speck, IDEA, HIGHT, LEA
GFN	PICCOLO, TWINE, CLEFIA, Piccolo, Twis HISEC

Block cyphers are preferred over stream cyphers in the context of IoT devices with limited resources. Block cyphers, notably symmetric lightweight block cyphers, and their useful applications are the main topics of this study.

2.2 Lightweight Cryptographic Primitives

Block cyphers, hash functions, message authentication codes, and stream cyphers are only a few ex- amples of the different cryptographic elements included in primitives. Assuring data security, excellent performance, and low resource utilization is the goal of this progression, and the structure-wise LWC algorithms are shown in Table 3. Listed below are some lightweight cryptographic primitives: Lightweight block cipher: The performance assistance of the Lightweight block ciphers project includes limited block sizes (60/80-bit), optimal key sizes (<96-bit), a more straightforward rounds function (4-bit S-boxes), key schedules (produce sub-keys on the fly), and minimal hardware implementations in terms of gate equivalents (GE) or lookup tables (LUT). It is essential to choose lightweight block cipher over conventional block cipher for power-constrained devices; lightweight block cipher specifically follows lightweight design cri-teria to balance tradeoffs. To minimize memory utilization, lightweight block cipher may utilize a 64-bit block size instead of the 128-bit used in standard AES block ciphers. Small square sizes also cut down on the amount of plaintext that needs to be encoded. For instance, a 232-block uneven layout can be used to identify a 64-bit square figure. This may result in plaintext recovery, key recovery, or confirmation label imitations, depending on the calculation. It is also essential to use a small key size for a lightweight block cipher, but the key size should not be so small that it can be easily brute-forced in seconds. We need to define the key scope of up to 112 bits specified by NIST for good efficiency. The key schedule is a significant part of any encryption scheme. This efficient, simple key scheduling process can generate keys on the fly to minimize extra overhead and achieve high performance in developing complex ciphers. It is crucial to ensure that each key is produced individually.

1. **Lightweight hash function:** Reduced communication size and reduced internal state and output sizes are likely components of lightweight hash functions' processes. The smaller internal state allows the application to use the collision-resistant hash function, which has security against several cryptanalytic attacks on the generated hash. The lightweight hash function needs to use a smaller message size of up to 256 bits instead of the conventional message of 264 bits.
2. **Lightweight message authentication codes:** A single secret key tag that is often used to authenticate communication is created using the MAC. NIST preferred to make use of shorter tags in a statement. Some existing standard algorithms used as lightweight message authentication codes are CHASKEY [31], LightMAC [32], and TuLP [33].
3. **Lightweight stream ciphers:** NIST-approved cipher suited for power-constrained devices suitable for widespread adoption. Standards supported by lightweight stream ciphers are [34] Trivium [35] and Mickey [36].
4. **NIST standards for lightweight cryptography:** In 2013, NIST started looking towards cryptography for limited resources. NIST has begun a process to solicit, evaluate, and standardize schemes providing authenticated encryption with associated data (AEAD) and optional hashing functionalities for constrained environments where the performance of current NIST cryptographic standards is not acceptable. This follows two workshops and discussions with stakeholders from industry, government, and academia. A request for algorithms, including specifications, a selection procedure, and assessment criteria, was released by NIST in 2018 [37]. In the first round of the standardisation process, NIST received 57 proposals in March 2019. With the introduction of Round 1 in April 2019, the NIST initiated the first phase

of their standardisation process for lightweight cryptography. The 32 candidate algorithms that made it through to the second stage of examination are included in NISTIR 8268, along with an explanation of how they were scored in the first round. When NIST revealed the 32 candidate algorithms that progressed to the second stage are outlined in NISTIR 8268, along with an explanation of their scores from the first round. In August 2019, NIST disclosed the 32 candidates for Round 2, marking the official start of the second phase of the lightweight cryptography standardization process. This phase concluded in March 2021 with the announcement of finalists. Details of the second-round evaluations and the top 10 applicants can be found in NISTIR 8369. The final phase commenced with the unveiling of the top 10 candidates and concluded in February 2023 when NIST selected the Ascon family after careful consideration, as outlined in NISTIR 8454 [38].

2.3 Design Strategies of Lightweight Cryptography

Designing a lightweight cryptography algorithm is more complicated than it may seem at first glance in terms of security. Since we talk about "lightweight" primitives, there must be a way to define the weight of an algorithm. Primitive is determined by the amount of time and space resources required to run it. This weight can be measured in both software and hardware [39]. Lightweight software does not mean lightweight hardware. Finally, the power consumption of target devices is crucial in both scenarios [24,40]. Today's civilization is dominated by electronic devices that make man's existence impossible. Several household devices with embedded operating systems can connect to the Internet or even form a wireless network [41]. Various terminals, readers, and sensors surround people everywhere. These developments exacerbate data security issues. We are unable to provide a cryptographic primitive that can be used by many target devices. As we can see, AES is a safe and efficient encryption method [29,42]. AES is recommended for high-end and low-end devices to avoid resource and power limits. With limited power, typical cryptographic techniques cannot be used. Examples of such devices include RFIDs, inexpensive smart cards (including wireless), wireless sensors, indicators, measuring devices, and custom controllers. Every lightweight block cipher creator must customize low-powered devices, balancing resource efficiency, performance (in throughput), and resistance to cryptographic attacks [43]. See Fig. 6.

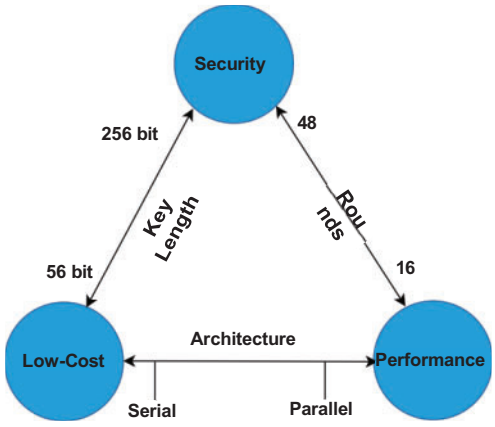

Figure 6: Design tradeoffs for lightweight cryptography system

In general, it is possible to effectively optimize any two out of the three objectives: security and low costs, security and execution, or low expenses and performance. However, it is tough to advance every

one of the three structure objectives in the meantime. For instance, pipelined engineering can use safe and elite equipment, joining numerous countermeasures against cryptanalytic attacks. The following structure would have a high territory necessity related to high costs. Then again, it is conceivable to structure safe and easy equipment usage with the downside of limited performance. For the most part, there are three methodologies for giving cryptographic natives to amazingly lightweight applications, which include RFID, wireless sensor networks (WSNs), aggregation networks, pervasive devices, IoT, and embedded devices:

2.4 Performance Metrics for Hardware and Software

The performance metrics of hardware like FPGA devices can be measured using the following metrics: area, clock cycles, time, throughput, power, energy, and efficiency.

- **Area:** It is the overall requirement for the FPGA design, which is estimated at μm^2. Looking at the region prerequisites freely usually expresses the region as lookup tables (LUTs) in FPGA. Each lookup table represents the function that is mapped to an equivalent number of gates equivalent (GE). Thus, each lookup table may consume 3 to 8 inputs, depending on a different FPGA architecture. Whereas generally, memory is measured in KB [26]. The program or algorithm must be stored in ROM, whereas intermediate values must be kept in RAM.
- **Clock Cycles:** The speed of a PC processor, or CPU, is dictated by the clock cycle, which measures the time between two beats of an oscillator. As a rule, the higher the number of pulses every second, the quicker the PC processor will almost certainly process data. The checking speed is estimated in Hz.
- **Time:** Time is determined by dividing the total number of clock cycles by the operating frequency. A millisecond (ms) is a unit of measurement for time Â.
- **Throughput:** It is a parameter used to measure an FPGA device's performance. The Throughput is not calculated directly when implementing various cryptographic algorithms using Verilog or VHDL in the synthesis process. It must be obtained physically. The throughput is determined using the highest clock speed and the number of bits per clock cycle. The highest clock speed can easily be calculated from the Latency, i.e.,

$$Highest\ Clock\ Speed = \frac{1}{Latency} \tag{1}$$

- **Power:** The power consumption is estimated from an FPGA device's routing file and gate-level netlist. The Place routing file is routed to the interconnection of the FPGA architecture. At the same time, the gate-level netlist consists of lookup tables (LUTs) and clock buffers. The physical layout is not required to estimate power consumption in FPGAs. Xilinx provides a tool called Power. It uses logic-library and standard-cell power characteristics. Power consumption is measured in microwatt μW to determine power consumption [44].
- **Energy:** Energy utilization signifies power utilization over a specific period. It very well may be de- termined by increasing the power utilization with the essential time of the activity. For the effectiveness of lightweight cryptographic algorithms, it may be interesting to evaluate the energy utilization per output bit and the energy consumption measured in microjoules (μJ).
- **Efficiency:** The efficiency of a lightweight cryptographic algorithm can be calculated on an FPGA device using the total area consumed per throughput (in gigabytes per second), i.e.,

$$eff = \frac{Area\ (LUTs)}{Throughput} \tag{2}$$

3 Literature Survey

Lightweight encryption algorithms are designed to provide secure cryptographic operations while min-imizing computational and memory resources. Particularly for low-resource devices like embedded systems or IoT (IoT) devices, these algorithms work effectively [45]. Here are a few examples of lightweight encryption algorithms:

AES: The Advanced Encryption Standard (AES) is a standardized algorithm commonly used for resource-constrained devices at the application layer. AES is a symmetric block cipher that encompasses three variants of the Rijndael cipher, which have been standardized by NIST [1,46]. AES has three variants: 128/192/256-bit block sizes. The inward state is dependent on the 128-bit key size. Encryption comprises the accompanying tasks performed over the 128-bit inner state composed as a 4 4 framework of bytes of cryptography standards. Every byte gets infused with SubBytes, ShiftRows, MixedColumns, and Add-RoundKey functions. AES's maximum permissible key size is 128/192/256 bits, and it may be cracked via biclique and man-in-the-middle attacks on whole rounds of AES [47,48].

CHAM-64/128: Koo et al. [49] have proposed an innovative lightweight block cipher called CHAM, particularly for resource-constrained devices. CHAM operates on a fixed-size block of 64 bits for input, and the key sizes for CHAM-128/128-bit and CHAM-128/256-bit are 128 bits and 256 bits, respectively. CHAM works on a generalized Feistel network based on A-addition, R-rotation, and X-XOR operations for generating ciphertext. CHAM consumes less hardware area as compared to SIMON by 73%. CHAM ruins 1110 GE for 64/128, 1899 GE for 128/128, and 2087 GE for 128/256-bit, respectively. The design of CHAM-64 takes 80 rounds to generate full ciphertext, CHAM-128/128 takes 80 rounds, and CHAM-128/256 takes 96 rounds to create full ciphertext [49].

CLEFIA: CLEFIA is a traditional block cipher of 128 bits; it's fundamental size shifts from 128, 192, and 256 bits, respectively [47]. CLEFIA attains 1.6 GB/s with less than 6K gates equivalent. CLEFIA is susceptible to integral attack for 12, 13, and 14 rounds and improbable attack for 13, 14, and 15 rounds, respectively [50,51]. The CLEFIA encryption procedure is an innovative symmetrical lightweight block cipher algorithm invented by Sony Corporation, concentrated on digital rights management (DRM) resolu-tions [52]. CLEFIA makes encryption more secure by using techniques like dissemination switch systems, which use different dispersion lattices in a certain order to protect against differential and linear cryptanalysis attacks, and whitening keys, which combine information with parts of the key before and after the first and last rounds. CLEFIA was implemented over FPGAs for innovation given their favourable circumstances regarding computation adaptability, time to showcase, advancement expenses, and organization time for committed solutions [27,53].

FeW: FeW is an innovative, lightweight block cipher designed for software-based systems. It operates on 64-bit plaintext and generates 64-bit ciphertext over 32 rounds. FEW incorporates a combination of Feistel and generalized Feistel structures to enhance security against simple cryptographic attacks. Moreover, FEW demonstrates resistance to linear and zero correlation attacks, ensuring robust protection for the design [54].

HIGHT: The HIGHT algorithm is suitable for low-resource devices. HIGHT is an ARX-based GFS coordinate with key brightening [55]. XOR and bitwise rotations are the significant operations used. Three different information pivots are XORed together in the F0 and F1 subfunctions. The master key and 128 subkeys are used in the array to generate 8 bytes of whitening keys. In various regions of the internal state, addition and XOR are used simultaneously during key whitening and encryption. There are two attacks on complete HIGHT [56,57].

KASUMI: KASUMI is specially designed to fulfill security requirements for the mobile network, which includes GSM, EDGE, UMTS, and GPRS. KASUMI supports 3GPP confidentiality and the 3GPP integrity algorithm. It takes the input of a 64-bit block for a key size of 128 bits. The Feistel function operates for eight rounds to produce ciphertext. The input is divided into two halves of 32-bit strings, and then a round procedure is applied to each half to produce 32-bit ciphertext, which is again combined to form 64-bit ciphertext. KASUMI suffers from related key differential cryptanalysis attacks [58,59].

MIDORI: Midori was created with limited power sources in mind, making it ideal for applications like medical implants. Midori64 and Midori128 are included in the package. Iterations of 16 and 20 are used to generate a 128-bit key on 64-bit and 128-bit blocks, respectively [60].

KHUDRA: Khudra, a creative lightweight block cypher for field-programmable gate arrays, was built. It works with an 8-byte block size and a key size of 10 bytes. The Feistel network operates with the Feistel function, which performs substitution-permutation and diffusion of input blocks to generate a series of outputs as ciphertext. The outer structure of khudra uses 18 rounds, and the inner system takes 6 rounds to perform the cryptographic operation to generate ciphertext [57].

KLEIN: An innovative lightweight block cipher with 4×4 S-box [61]. All protections against side-channel attacks might be implemented on only one piece of equipment in this new location. Block cipher-based hash functions and message authentication codes are created by KLEIN writers. The 4×4 S-box utilizes the sub-nibbles step for implementation. Furthermore, all the S-boxes in the S-box layers are indistinguishable. The diffusion layer comprises two stages. Firstly, sixteen 4-bit elements are combined into 8 bytes, which are then rotated to the left by two stages, resulting in the second byte moving to the previous position (Rotate Nibbles). Subsequently, the bytes are divided into two sets of four bytes each, represented as vectors of $(GF(28))^4$ (Mix Nibbles). This last activity is very similar to the AES Mix Column. A Feistel key plan has two calls per S-box round and around counter XOR. They also attempted to anticipate related-key attacks and provide less demanding concealment to counteract side-channel attacks. The real issues identified with the KLEIN algorithm are differential attacks for 8 rounds and truncated differential attacks for various rounds [62].

KATAN and KTANTAN: A trivial and well-organized hardware-arranged stream cipher. Block size is 04/06/08 bytes, and key size settled at 10 bytes. KATAN and KTANTAN accept a compact 80-bit key implanted on the device only once, which cannot be updated further. Both KATAN and KTANTAN used different key schedule approaches to generate the complex ciphertext. In each cycle, bits from the registers are entered into two nonlinear Boolean capacities. The boolean capabilities give the registers the minimum critical bits (after moving). To guarantee adequate blending, 254 rounds of the figure are executed. Significant issues include a differential attack on KATAN-04-byte for 115 rounds, a multi-dimensional MiTM attack for 175/130/112 rounds for KATAN-04/06/08-bytes, and three subdivisions of the MiTM attack on KATAN [61,63,64,65].

LBLOCK: LBlock is a noteworthy cryptographic calculation exceptionally actualized to target both hardware and software benchmarks [66]. The core of its outline usage is done on Feistel development, which has two twigs of 4 bytes and is then swapped afterward. The XORed yield of the F task is essentially rotated by 1 byte. The computation function of the algorithm incorporates an XOR operation with subkeys, an 8×4 S-box layer, and permutation shuffling of 4-bit words. The key schedule involves the use of two additional S-boxes that differ from those employed in the Feistel function. LBlock is defenseless against impossible differential attacks for 21 rounds of LBLOCK, related-key attacks, zero-correlation attacks for 22 rounds, and integral attacks for 22 rounds [67,68,69].

ITUbee: ITUbee is a highly efficient software cipher with a code size of 586 bytes and a total execution time of 2937 cycles. This version of encryption is considered to be the most compact. The key and block sizes are identical (80-bit). Instead of key scheduling, round-dependent constants are used to reduce software overload.

LED: The LED Block Cipher is an SPN based on AES. Encryption is done in 48-slice increments with XORing of the key in between [70]. An individual round is formed by XORing an AES panache round. A key of 16 bytes is divided into two subkeys of 8 bytes each, whereas a key of 8 bytes is XORed with an internal state. Ad-hoc attacks (12 rounds of LED-08-bytes, 32 slices of LED-16-bytes) increased power usage due to rounds.

MANTIS: The SKINNY algorithm and its low-latency variation, MANTIS, have an 8-byte block size, a 16-byte key, and an 8-byte tweak. The MANTIS algorithm is based on the PRINCE algorithm and key schedule. Mantis' reflexivity is boosted by low expectations. The leading key is split into two halves, one for input and output whitening and the other for rounding. However, additional steps are taken to incorporate the change. The round function performs successive operations: MixColumns, PermuteCells, AddTweakey, Ad-dConstant, and SubCells. MANTIS resists truncated differential cryptanalysis with a probability of 2–67.73. A practical key recovery attack was successful on MANTIS for 228 chosen plaintext with a complexity of 238 [71]. MANTIS is very similar to the Midori block cipher. Most of the cryptanalysis of Midori can be used for MANTIS, particularly meet-in-the-middle attacks, slide attacks, and integral attacks on MANTIS. The most successful attacks on MANTIS are invariant subspace attacks with a compactness of 296 weak keys. Though MANTIS has sufficient security, which can be helpful in specific applications.

MCRYPTON: MCrypton was explicitly proposed for securing low-cost RFID tags and sensor nodes. This cipher is a derivative of CRYPTON, and the design is based on SPN with a core space systematized in a 4×4 matrix of half a byte. Out of four S-boxes, two S-boxes use an inverse function in GF (2^4). The other two s-boxes are inverses of the first two. Security issues in mCrypton MiTM 7 rounds for mCrypton-64/96, MiTM 8, 9 rounds for mCrypton-128-bit [72].

PICCOLO: Piccolo configuration was based on a generalized Feistel structure with four 2-byte branches, which utilizes a complex stage for the dispersion layer rather than a fundamental shift and whitening of keys [73]. The twigs of the Feistel structure are 2 bytes; the permutation function is 1 byte. The hardware imple- mentation of Piccolo is provided using just 4 NOR gates, 3 XOR gates, and 1 XNOR gate. The 4×4 S-box proposal provides respectable non-linearity and differential consistency. Piccolo was tested vulnerable to a biclique attack for Piccolo-10 bytes for 28 rounds. Piccolo-10-bytes and Piccolo-12-bytes versions are susceptible to related-key impossible differential attacks for 14 rounds [74,75].

PRESENT: The version of the PRESENT algorithm was purely based on the SPN structure, which is simple, efficient, and concerned with bit-orientation [76]. The hardware implementation of PRESENT was based on unpretentious cabling. However, software implementation is quite tedious. Its compact S-box was carefully chosen for its virtuous cryptographic properties while consuming little hardware area. The PRESENT algorithm was tested for vulnerability to statistical saturation for successive 24 rounds. PRESENT also suffers a multi-dimensional linear and truncated differential attack for 26 rounds [57,76].

QTL: QTL emerges as a new 8-byte cipher surplus with the key size of 8-bytes and 12-bytes [77] QTL is another variation summed up with a generalized Feistel structure. QTL generates ciphertext with 16 rounds for the QTL-64-bit version and 20 for the QTL-128-bit version. Each round is specialized with two more rounds in the internal structure. QTL is the quick dispersion of SPNs,

which tightens the security features of QTL in the Feistel network. QTL uses 4×4 S-boxes to provide appropriate security and consumes only 22 GE. It also uses a constant function that is dissimilar to the round function. QTL design consumes less hardware implementation for the 8-bytes, necessitating 1025.52 GE and 1206.52 GE (gate equivalents). QTL is vulnerable to known practical key recovery attacks conceivable on the 8-byte version of QTL [78].

RECTANGLE: RECTANGLE block cipher is decently based on bit-slice for both types of implemen- tation [79]. It was designed with two modifications. RECTANGLE-8-bytes with key size 10-bytes and RECTANGLE-12-bytes with key size 12-bytes operate for a complete 25 rounds. The non-linearity layer uses a 4×4 S-box on the state segments, while the direct layer uses a settled pivot with an alternate sum on each line. RECTANGLE runs on key scheduling in a matrix, going through each cycle of the encryp- tion procedure with S-Box. For 19 rounds, RECTANGLE is vulnerable [80]. The most recent variant of RECTANGLE distributed in the science journal China is not defenceless against these assaults any longer. RECTANGLE works extremely well in both software and hardware implementations.

ROADRUNNER: SPN structure as its core function in the Feistel structure. Roadrunner uses a typical Feistel structure based on a bit-sliced implementation. In the key schedule, 4 bytes of the master keys are utilized in a steady progression. Roadrunner utilizes four S-Boxes. It comes with an 8-byte block size, and the key size is 10 bytes for consecutive 10 and 12 rounds [81]. Implementing a 4-bit S-Box was preferred prudently with a direct route to calculate decent cryptographic properties. Roadrunner is vulnerable to truncated differential attacks for 5 rounds on the 10-byte version and 7 rounds on the 12-byte version.

SIMON & SPECK: The National Security Agency (NSA) authors present a lightweight family of a block cipher called SIMON and SPECK to deliver security to low-power-constrained devices [82]. The design of the algorithm is flexible, simple, and secure. The SIMON algorithm works on block size ranges of 32, 48, 64, 96, and 128 bits and key size ranges of 64, 72, 96, 128, 144, 128, 192, and 256 bits for 32, 36, 42, 44, 52, 54, 68, 69, and 72 rounds to generate the ciphertext. SPECK is an addition-rotation- XOR-based algorithm that works on block sizes of 32, 48, 64, 96, and 128-bit and key sizes of 64, 72, 96, 128, 144, 128, 192, and 256-bit for 22, 22, 23, 26, 27, 28, 29, 32, 33, and 34, respectively. The family is susceptible to differential attack, linear differential attack, impossible differential attack, and multi- dimensional linear attack on SIMON. SPECK is defenceless against differential attacks and rectangle attacks [83]. The authors also confirmed important differential trials on SIMON 34, 48, and 64-bit versions. Another notable discovery concerning SIMON demonstrates an efficient algorithm for calculating the differential probabilities (DP) of the modular AND operation with both independent inputs and rotationally dependent inputs. SIMON has been extensively utilized in threshold search and differential search tools, making this finding particularly significant.

TEA (Tiny Encryption Algorithm): The TEA encryption algorithm divides the data into 64-bit blocks and employs a 128-bit key. It consists of 64 rounds that collectively generate a sophisticated ciphertext. TEA was created in 1994 at the Cambridge Computer Laboratory by Roger Needham and David Wheeler, with a strong emphasis on simplicity in its design. TEA is mounted with a related-key attack on the full cipher [84], which improves a changed rendition called XTEA [85]. Regardless of the straightforwardness of the essential round capacities, an equipment execution of TEA still requires 2355 gate equivalents.

TWINE: TWINE is one of the most versatile lightweight algorithms based on GFS with sixteen 4-bit twigs [86]. The Feistel function executes for 32 rounds, incorporated with the key schedule and s-box. TWINE consumes 1779 GE for a 10-byte key size and 2285 GE for a 12-byte key size. It is a more current stage to accelerate diffusion that requires fewer rounds to diffuse to each sub-block. TWINE is also susceptible to zero-correlation attacks for TWINE-80-bit for 23 rounds and TWINE-128-bits for 25 rounds, respectively.

NOEKEON [64] uses 128-bit blocks and keys with 16-rounds of iteration. The NESSIE project did not use encryption because of its low resilience against attacks.

XTEA: XTEA is another lightweight block cipher designed based on ARX structure, and software implementation was done with the smallest amount of source code freely available over the World Wide Web. XTEA was specifically entrenched in the Linux kernel. XTEA is an improvised version of a previous project called TEA that had similar purposes but numerous weaknesses [87]. It uses modular operations for addition (modulo 232), left shift operations, convenient shift operations, and XOR operations in its round function to process input plaintext to achieve encrypted ciphertext at the end of rounds. XTEA also faces cryptanalytic issues with differential attacks on 14 rounds of XTEA, which are continuously based on 12 rounds of impossible differential attacks. Hong et al. proved XTEA is vulnerable to a truncated differential attack for 23 rounds with a probability of 2120.65 encryptions [88].

The author of this paper discusses lightweight cryptographic systems for IoT networks, as well as a comparative examination of major modern ciphers, and assesses the security of many recently proposed block cipher and stream cipher algorithms [89]. This article examines lightweight cryptographic solutions for the Internet of Things (IoT). This study includes a wide range of security measures, from lightweight cryptographic methods to comparing different types of block ciphers. It also compares hardware *vs.* software solutions and several contemporary uses of the most trusted and researched block cipher, Advanced Encryption Standard (AES), in terms of design, mix-column and S-box modification strategies, and threats to IoT security. According to the study, lightweight AES is an effective security solution for restricted IoT devices [90] Sattar B. Sadkhan gives an analysis of contemporary hardware H/W and software S/W implementations of symmetric and asymmetric ciphers [91].

We listed a comparative analysis of several lightweight block ciphers according to their design proper-ties with known security issues are shown in Tables 4 and 5.

Table 4: A comparative overview of existing surveys with our paper

References	Cipher coverd	Cryptanalytic attack	Area	Power	Energy	Throughput	Real time processing
[92]	13	Yes	Yes	–	Yes	–	Yes
[26]	20	Yes	Yes	–	–	Yes	–
[15]	21	–	Yes	–	–	–	Yes
[93]	17	–	Yes	Yes	–	Yes	–
[94]	21	Yes	Yes	–	–	–	Yes
[95]	9	–	Yes	–	–	Yes	–

(Continued)

Table 4 (continued)

References	Cipher coverd	Cryptanalytic attack	Area	Power	Energy	Throughput	Real time processing
[58]	15	Yes	Yes	Yes	Yes	–	Yes
[61]	20	–	Yes	Yes	–	–	Yes
[67]	22	–	Yes	–	Yes	–	–
[70]	19	Yes	Yes	–	–	Yes	Yes
This paper	24	Yes	Yes	Yes	Yes	Yes	Yes

Table 5: Review of lightweight block ciphers

Ref.	Algorithm	Design	Block size	Key size	Rounds	Cryptanalytic attack
[96,50,97]	AES	SPN	128	128/198/256	10/12/14	Impossible differential attack, Related-key attack and vulnerable to biclique attack
[49]	CHAM	ARX	64/128	64/128	16/32	Differential and linear attack to full cipher
[27,98,64]	CLEFIA	GFS	128	128/192/256	18/22/26	Vulnerable to integral attacks for 12, 13, 14 rounds of CLEFIA 128; CLEFIA 128 is susceptible to improbable differential attacks for 13, 14, 15 rounds.
[54]	FeW	GFS	64	80/128	32	Zero correlation attack

(Continued)

Table 5 (continued)

Ref.	Algorithm	Design	Block size	Key size	Rounds	Cryptanalytic attack
[55,56,57]	HIGHT	GFS	64	128	32	Related key attack, biclique attack and Impossible differential cryptanalysis attack
[58,59]	KASUMI	Feistel	64	128	8	Boomerang, Related key & Single key attack
[61]	KLIEN	SPN	64	64/80/96	12/16/20	Truncated differential attack
[61,63,64,65].	KATAN/ KTANTAN	SPN	32/48/64	80	12	Related key attack
[67,68,69].	LBLOCK	Feistel	64	80	32	Zero-correlation, Related key attack, Integral attack, and Impossible differential attack
[70]	LED	SPN	64	64/128	32/48	Key recovery attack and Ad-Hoc attack
[71]	MANTIS	SPN	64	128+64	14	Meet-in-the-middle attack
[72].	MCRYPTON	SPN	64	64/96/128	12	Meet-in-the-middle (MitM) attack
[99]	MIDORI	SPN	64/128	128	16/20	MitM attack
[99]	NOEKEON	SPN	128	128	16	Related key attack
[100]	PICCOLO	GFN	64	80/128	25/31	Integral attack
[73,101]	PRESENT	SPN	64	80/128	31	Impossible differential & Biclique attack

(Continued)

Table 5 (continued)

Ref.	Algorithm	Design	Block size	Key size	Rounds	Cryptanalytic attack
[102,103,104]	PRINCE	SPN	64	128	12	Multiple differential attacks, Reflection attack, Sieve-in-the-Middle (SitM) attack
[77,78,105]	QTL	ARX	32/64/128	32/64/128	12	Linear and Differential attack
[79]	RECTANGLE	SPN	64	80/128	25	Related-key & vulnerable to biclique attack
[81]	ROADRUNNER	Feistel	64	80/128	10/12	Differential and linear attack
[104,72,106,107]	SIMON	SPN	32/48/64/128	64/128/192/256	32/36/44/72	related weak key attack; differential attack
[108,109]	TEA	GFS	64	128	64	Related key attack for complete round cipher; Impossible differential cryptanalytic attack on TEA; a differential attack on rounds of TEA.
[69,43,86]	TWINE	GFS	64	80/128	36	Vulnerable to Zero correlation attacks for 23 rounds for 80-bit key size and 25 rounds for a 128-bit key size of TWINE; Full biclique attacks on TWINE
[110,44]	XTEA	Feistel	64	128	64	Zero correlation attack

3.1 Comparison of Software and Hardware Performance of LWC Algorithms

Numerous researchers have conducted studies to analyze the performance of common lightweight cryptography algorithms [25,46,49,51] on a variety of platforms, including NXP micro-controllers, AVR microcontrollers, and ARM [29] microcontrollers. Several lightweight cryptographic methods were examined in terms of area (GE), logic process (m), power consumption (W), throughput (Mbps), RAM/ROM requirements (KB), and delay (cycle/block). Various lightweight cryptographic algorithms were assessed in various scenarios throughout these trials. Among the described LWC algorithms, the hardware and software performance was evaluated on 0.09/0.13/0.18/0.35 m technology and 8/16/32 bit micro-controller platforms, respectively, as shown in Table 6.

Table 6: Compression among software and hardware performance of various LWC algorithms

Algorithm	Software implementation								Hardware implementation						
	Key size	Bock size	ROM (byte)	RAM (byte)	Latency (cycles/ block)	Energy (Âμ*J*/ *bit*)	Throughput @ 4MHz (Kbps)	Software Efficiency (Kbps/KB)	Key size	Bock size	Area	Power (Âμ*W*)	Energy (Âμ*J* /*bit*)	Throughput @ 4MHz (Kbps)	Hardware efficiency (KGE)
AES	128	128	918	0	4192	16.7	122	132.9	128	128	2400	2.4	42.38	56.64	23.6
CLEFIA	128	128	1920	78	3646	4.9	140.42	73.14	128	128	2678	2.67	36.82	76	28.37
HIGHT	128	64	5718	47	6377	25.5	40.14	7.02	–	–	–	–	–	–	–
KASUMI	128	64	1264	24	11939	47.6	21.4	16.93	128	64	3437	3.44	29.9	115.14	33.5
KATAN	80	64	338	18	72063	2892	3.5	10.55	80	32	802	0.8	64.16	12.5	15.58
KLIEN	64	64	2980	50	7901	10.6	32.4	10.87	64	64	1220	1.83	5918	30.9	25.32
LBLOCK	80	64	976	58	18988	25.6	13.48	13.81	80	64	1320	2	9.9	200	151.51
LED	80	64	2164	368	35161	–	7.28	3.36	64	64	966	1.45	282.55	5.1	5.27
MCRYPTON	96	64	1076	28	16457	68	15.5	14.41	128	64	2594	4.66	138.61	33.51	12.91
MIDORI	–	–	–	–	–	–	–	–	128	64	1542	60.6	1.61	400	259.4
NOEKEON	128	128	364	32	23516	94.9	21.8	59.62	128	128	2604	4.68	1362.21	3.44	1.32
PICCOLO	80	64	966	70	21448	28.9	11.93	12.33	80	64	1136	1.13	4.8	237.04	208.66
PRESENT	128	64	660	0	10792	43.1	23.7	35.91	80	64	1570	2.35	11.77	200	127.38
PRESENT	80	80	716	0	2607	10.4	122.7	171.37	–	–	–	–	–	–	–
PRINCE	128	64	1108	0	3614	14.4	70.8	63.9	128	64	2953	2.95	5.53	533.3	180.59
RECTANGLE	–	–	–	–	–	–	–	–	80	64	1467	1.46	5.96	246	167.68
SPECK	96	48	134	0	408	1.6	470.5	3511.19	96	48	884	0.88	73.67	12	13.57
SIMON	96	48	170	0	594	203	323	1900	96	48	763	076	48.32	15.8	20.7
TEA	128	64	648	24	7408	30.3	34.5	53.24	128	64	2355	3.53	35.32	100	42.46
TWINE	80	64	1180	140	20505	–	12.48	10.58	80	64	1503	1.05	5.91	178	118.42
XTEA	128	64	504	0	17514	70	14.6	28.97	–	–	–	–	–	–	–

The below graph (Fig. 7) shows the top 10 most memory-efficient LWC algorithms and their respective memory (RAM and ROM) requirements. Fewer requirements of ROM and RAM, SPECK, and SIMON take the top place in the NIST contest.

Midori tops the hardware and software efficiency list, followed by PICCOLO and GOST, with a slight gap. The first ten hardware-efficient LWC algorithms are shown in Fig. 8.

In terms of software and hardware throughput, SPECK and SIMON have the lowest Throughput, whereas PRIDE has the maximum throughput rates shown in Fig. 9.

In summary, SIMON and SPECK flourish in software but fall out of the top 10 hardware-efficient LWC algorithms. Also, AES-derived algorithms like PRESENT and DES-derived algorithms like DESL/DESLX and CLEFIA have been generally accepted algorithms (by standardization organizations) for high-security reasons [111,112]. The comparison of software and hardware performance among various Lightweight Cryptography (LWC) algorithms is shown in Table 6.

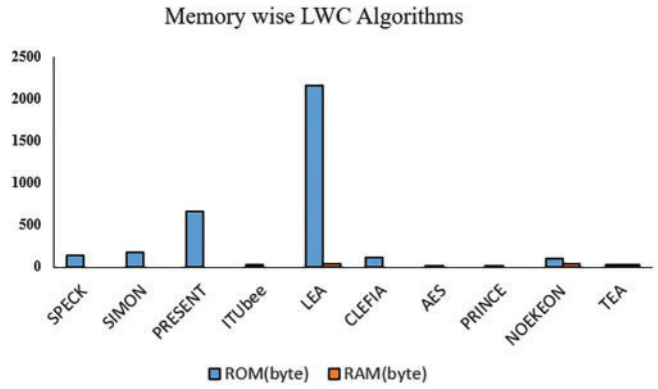

Figure 7: Top 10 memory wise LWC algorithms

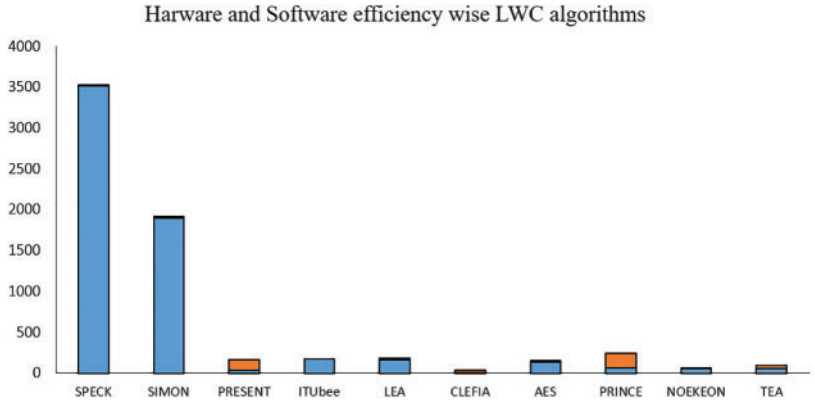

Figure 8: Top 10 efficiency wise LWC algorithms

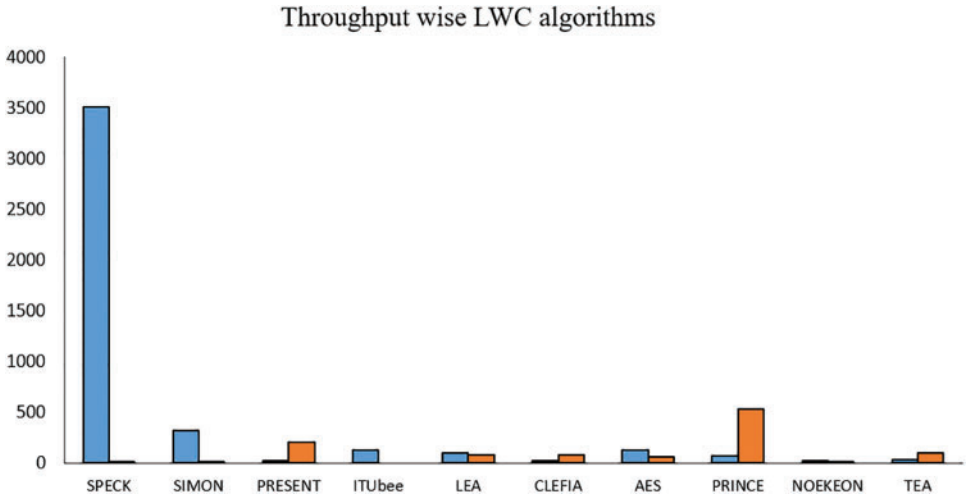

Figure 9: Software/Hardware throughput wise Top 10 LWC algorithms

3.2 Limitations of Specific LWC Algorithms Versus Existing Hardware-Based Solutions

Traditional hardware-based cryptographic solutions may be difficult for resource-constrained devices like IoT devices or low-power sensors owing to their large computational and memory needs, however, lightweight cryptography techniques are intended to offer security in these situations. When opposed to hardware-based alternatives like Artificially Intelligent Electronic Money (AIEM), the benefits of lightweight cryptographic algorithms are tempered by their own set of constraints and possible flaws [113]. Let us have a look at some of the restrictions and flaws:

Strong Security: Lightweight cryptography methods may be less secure than regular ones because they use fewer resources. They may be more susceptible to brute force or sophisticated cryptanalysis, rendering them unsuitable for secure applications like financial transactions.

Short Keys: To avoid computational cost, several lightweight cryptography techniques employ lower key lengths. With increasingly powerful computer technology, shorter keys might be more vulnerable to assaults. In contrast, AIEM systems may use longer, more secure encryption keys.

Quantum Attack Vulnerability: Lightweight cryptographic methods may be more vulnerable to quan-tum assaults than hardware. Quantum computers might break lightweight algorithms more readily, threat-ening IoT security.

Lack of Post-Quantum Resilience: Some AIEM and other hardware-based cryptography methods are post-quantum robust. They are less vulnerable to quantum assaults, which is important for long-term secu-rity. Not all lightweight algorithms are quantum resilient.

Key Management: Advanced cryptographic solutions like AIEM employ more resilient key manage- ment methods than lightweight cryptography techniques. Cryptographic system security requires proper key management.

Limited Function: Lightweight cryptography techniques are often used for authentication and data en-cryption. AIEM systems are more appropriate for complicated and varied applications due to their versatility and extensive range of features.

Resource Limits: Lightweight algorithms are developed for resource-constrained systems, how-ever, they may suffer from performance and memory limits. This may create weaknesses in certain situations.

3.3 Cryptanalysis Attack on LWC Algorithms

The security of any lightweight cryptographic algorithm should be considered along with per-formance and cost. Any lightweight cryptography algorithm's attack resistance can be assessed by cryptanalysis. Various attacks and deciphering techniques are used in cryptanalysis to identify algorithm flaws. This includes differential, linear, integral, and algebraic cryptanalysis [114,115], respectively. Differential cryptanalysis compares inputs to higher-order, truncated, impossible, and boomerang types. Plaintext, ci-phertext, and key are approximated linearly using the piling-up lemma (Matsui's invention). S-P-N block ciphers with substitution-permutation networks benefit from integral cryptanalysis. A square attack and a saturation attack are also documented. As a result of its simplicity and use in lightweight versions, algebraic cryptanalysis has proven beneficial. These cryptanalyses use MITM, brute force, and side-channel ciphertext exclusively. An example of a differential fault attack is found in the internal structure of the algorithm [116]. The comparative analysis of different attacks on lightweight block ciphers is shown in Table 7.

Table 7: Comparative analysis of the different attacks on lightweight block ciphers

LWC algorithms	Related key attack	Linear cryptanalysis	Differential cryptanalysis	Side-channel attack	Integral/square cryptanalysis	Biclique /MITM
AES	✓	–	✓	✓	–	✓
CHAM-64/128		–	–	–	–	✓
CLEFIA	✓	–	–	✓	✓	–
FeW	–	–	–	–	–	–
HIGHT	✓	✓	–	–	–	✓
KASUMI	✓	✓	–	–	–	–
KHAZAD	–	–	–	–	–	–
KHUDRA	–	–	–	–	–	–
KLEIN	✓	–	–	✓	–	✓
KATAN and KTANTAN	–	–	–	–	–	✓
LBLOCK	–	✓	–	–	✓	✓
LED	✓	✓	–	–	–	✓
MANTIS	–	–	–	–	–	–
MCRYPTON	–	–	–	✓	–	–
PICCOLO	–	✓	–	–	–	✓
PRESENT	✓	–	✓	✓	–	–
QTL	–	–	–	–	–	–
RECTANGLE	✓	–	–	✓	✓	–
ROADRUNNER	–	–	–	–	–	–
SIMON & SPECK	✓	✓	–	–	–	–
TEA	✓	–	–	–	–	–
TWINE	–	–	–	–	✓	✓
XTEA	✓	–	–	–	–	–

1. **Related Key Attack:** A related-key attack is a cryptographic attack in which an adversary analyzes the behaviour of a cryptographic algorithm when it is used with related keys. Related-key attacks are particularly concerning because they might reveal weaknesses that are not apparent under regular cryptanalysis.
2. **Linear Cryptanalysis:** Linear cryptanalysis is another technique where attackers analyze linear approximations between plaintext, ciphertext, and key bits. If an LWC algorithm exhibits linear be-haviour, it might be vulnerable to this type of attack.
3. **Differential Cryptanalysis:** Differential cryptanalysis involves observing how differences in input data lead to differences in output. Attackers analyze the differences and relationships between plain-texts and ciphertexts to deduce key information. LWC algorithms, just like any other cryptographic primitive, can be vulnerable to differential cryptanalysis if not designed properly [117].
4. **Side-Channel Attacks:** Side-channel attacks exploit information leaked during the execution of an algorithm, such as power consumption, timing, or electromagnetic radiation. While not

a direct cryptanalysis attack, side-channel attacks can reveal key information and weaken the security of LWC algorithms.

5. **Integral Cryptanalysis:** integral cryptanalysis is a differential cryptanalysis technique that targets ciphers' algebraic properties, especially the propagation of differences through the cipher's rounds. It involves analyzing the behavior of the cipher when certain differences in the input are propagated through the various rounds.
6. **Biclique Cryptanalysis:** Biclique cryptanalysis is an advanced technique used to analyze and poten-tially break cryptographic algorithms, particularly block ciphers. This approach extends traditional differential cryptanalysis by considering both differential characteristics and boomerang characteristics simultaneously, making it more powerful against certain ciphers [118].

4 Future Developments in Lightweight Cryptography for IoT

Lightweight encryption for the Internet of Things (IoT) is a rapidly developing area because of the special problems provided by low-power devices. Improvements to security, efficiency, and responsiveness to new threats and technologies are anticipated to drive future research and development in this field [119]. Future directions of lightweight cryptography for the Internet of Things are discussed below:

1. Quantum-Resistant Algorithms: As quantum computing advances, lightweight cryptographic algo-rithms that withstand quantum assaults are in demand. Quantum-resistant lightweight cryptographic methods may be developed to secure IoT devices.
2. Enhanced Security with Post-Quantum Algorithms: Research and development will integrate post-quantum cryptography into lightweight algorithms, enhancing resistance to conventional and quantum assaults.
3. Standardization and Interoperability: Ensuring interoperability across multiple IoT ecosystems requires widespread adoption of lightweight cryptographic protocols. Standardised lightweight cryptography techniques and protocols may enable secure device communication between manufacturers.
4. Adaptive Cryptography: Valuable lightweight and adaptable cryptographic solutions for shifting security needs. Self-adjusting cryptographic algorithms that dynamically adapt their security settings depending on the threat environment may provide stronger security in the future [120].
5. Enhance Key Management: Key management is crucial for security. Innovative and effective key management strategies for lightweight cryptography in IoT may provide strong and scalable encryption key management solutions.
6. Energy Efficiency: To address IoT power limits, future lightweight cryptography may prioritise energy-efficient algorithms. Cryptographic activities might use less energy using new methods.
7. Integrating Blockchain and Distributed Ledger Technology: Integrating blockchain and distributed ledger technology may improve the security and trustworthiness of lightweight cryptography in IoT. These tamper-resistant storage and decentralised consensus systems may complement lightweight encryption for data integrity and authentication [121].
8. Machine Learning and AI: These technologies may enhance the efficiency of lightweight cryptography techniques. AI-driven solutions may optimise cryptographic processes in real-time for IoT devices and identify and react to security risks.

9. Research on Lightweight Cryptographic Primitives: It includes block cyphers, hash functions, and authentication techniques. This study may provide more efficient and safe lightweight cryptographic algorithm building blocks.
10. User-Friendly Systems: Ensuring developer and end-user usability is essential for lightweight cryptographic systems. To promote lightweight cryptography usage, future innovations may ease implementation and setup [122].

5 Challenges and Directions for Open Research

Due to its importance in protecting resource-constrained devices like IoT devices and low-power em- bedded systems, lightweight cryptography (LWC) has received a lot of attention in recent years [123,124]. Lightweight cryptographic methods have been the subject of much research and review by the cryptographic community. The security of LWC presents many important study topics and difficulties, including the following:

1. **Creation of Algorithms:**
   - **Research:** Creating and evaluating resource-constrained device-specific lightweight cryptographic algorithms.
   - **Challenges:** Developing algorithms that are both safe and efficient, while still being light on re- sources. To that end, researchers have been working on streamlined authentication procedures, hash functions, and block cyphers.
2. **Security Analysis:**
   - **Research:** The suggested LWC algorithms' security will be evaluated by testing them against a variety of cryptographic attacks.
   - **Challenges:** Protecting LWC algorithms against assaults like side-channel attacks and algebraic attacks, which exploit their inherent weakness, is a top priority.
3. **Standardization Efforts:**
   - **Research:** Contributing to the development of lightweight cryptographic standards via participation in standardisation procedures.
   - **Challenges:** To guarantee that LWC algorithms satisfy the necessary security requirements and interoperability standards, it is necessary to navigate the complicated terrain of cryptographic standards.
4. **Implementation Security:**
   - **Research:** Analysing the safety of LWC algorithms in practice across a range of hardware and software environments.
   - **Challenges:** Finding and fixing implementation flaws, protecting against side-channel attacks, and making sure devices with varying hardware specifications can run LWC algorithms safely is a top priority.
5. **Key Management and Secure Protocols:**
   - **Research:** Creating safe and effective mechanisms for managing keys in LWC algorithms.
   - **Challenges:** Secure key generation, distribution, and management in low-resource settings, as well as the creation of attack-resistant protocols.

6. **Post-Quantum Security:**
   - **Research:** Considering the vulnerability of many current cryptography systems to assaults from quantum computers, researchers are looking at how well LWC algorithms hold up.
   - **Challenges:** Creating LWC algorithms that are safe even after the introduction of quantum computers, which pose a danger to traditional cryptography methods.
7. **Standardization of Lightweight Cryptographic Primitives:**
   - **Research:** Developing common lightweight cryptographic building blocks for usage in a range of security protocols; examples include block cyphers and hash functions.
   - **Challenges:** Facilitating the safe, effective, and platform-agnostic use of standardised primitives.
8. **IoT Security Ecosystem:**
   - **Research:** Resolving issues with device administration, secure booting, and encrypted communication that are unique to the Internet of Things environment.
   - **Challenges:** Creating all-encompassing security solutions that protect every stage of the Internet of Things (IoT) ecosystem.
9. **Privacy and Regulatory Compliance:**
   - **Research:** Protection of personal information by requiring that LWC algorithms and implementations meet all applicable requirements.
   - **Challenges:** Striking a balance between security and privacy while adhering to regional and sector-specific legislation.

The need to protect low-power gadgets is driving continued exploration and development of lightweight cryptography [125]. To facilitate the broad deployment of IoT and other low-power devices without sacrificing security, it is essential that the security problems in this sector be met.

6 Conclusion

IoT security is a significant challenge as connected devices expand exponentially across many industries [126]. As a result, there is a need for a lightweight algorithm that balances cost, performance, and security. IoT devices with limited processing power can use lightweight cryptography to encrypt their communications. Comparisons of NIST-defined LWC properties (cost, performance, and security) and discussions of different gaps and open research issues [127]. The literature study indicates that NIST has accredited PRESENT and CLEFIA block ciphers for cost- and security-related considerations. On the other hand, SIMON and SPECK have the smallest implementations. When it comes to performance measures, no single LWC method can meet all requirements, regardless of the environment in which it is being used. However, when new LWC algorithms are developed, further assaults are disclosed [128]. This is a constant and never-ending process.

Acknowledgement: The authors wish to express their appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper.

Funding Statement: This work has been supported by project TRANSACT funded under H2020-EU.2.1.1.—INDUSTRIAL LEADERSHIP—Leadership in Enabling and Industrial Technologies—Information and Communication Technologies (Grant Agreement ID: 101007260).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: S. K., D. K., R. D., G. C., N. D., and I. Y.; data collection: S. K., D. K., R. D., G. C.; analysis and interpretation of results: S. K., D. K., R. D., G. C., N. D., and I. Y.; draft manuscript preparation: S. K., D. K., R. D., G. C., N. D., and I. Y. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] V. A. Thakor, M. A. Razzaque and M. R. Khandaker, "Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities," *IEEE Access*, vol. 9, pp. 28177–28193, 2021.

[2] P. Singh, B. Acharya and R. K. Chaurasiya, "Lightweight cryptographic algorithms for resource-constrained iot devices and sensor networks," in *Security and Privacy Issues in IoT Devices and Sensor Networks*. Elsevier, pp. 153–185, 2021. https://doi.org/10.1016/B978-0-12-821255-4.00008-0

[3] B. B. Ehui, Y. Han, H. Guo and J. Liu, "A lightweight mutual authentication protocol for IoT," *Journal of Communications and Information Networks*, vol. 7, no. 2, pp. 181–191, 2022.

[4] C. Kunndra, A. Choudhary, P. Mathur, K. Pareek and G. Choudhary, "CNN-LSTM: A deep learning model to detect botnet attacks in the Internet of Things," in *Int. Conf. on Cryptology & Network Security with Machine Learning*, Singapore, Springer, pp. 353–365, 2022.

[5] P. Mall, R. Amin, A. K. Das, M. T. Leung and K. K. R. Choo, "PUF-based authentication and key agreement protocols for IoT, WSNs, and Smart Grids: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8205–8228, 2022.

[6] M. Abinaya and S. Prabakeran, "Lightweight block cipher for resource constrained iot environment—an survey, performance, cryptanalysis and research challenges," in *IoT Based Control Networks and Intelligent Systems: Proc. of 3rd ICICNIS 2022*, St. Joseph's College of Engineering and Technology, Kottayam, Kerala, pp. 347–365, 2022.

[7] D. Natarajan and W. Dai, "Seal-embedded: A homomorphic encryption library for the internet of things," in *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 756–779, 2021. https://doi.org/10.46586/tches.v2021.i3.756-779

[8] M. N. Khan, A. Rao and S. Camtepe, "Lightweight cryptographic protocols for IoT-constrained devices: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4132–4156, 2020.

[9] S. Di Matteo, M. L. Gerfo and S. Saponara, "VLSI design and FPGA implementation of an NTT hardware accelerator for homomorphic seal-embedded library," *IEEE Access*, vol. 11, pp. 72498–72508, 2023.

[10] F. Mendoza-Cardenas, A. J. Aparcana-Tasayco, R. S. Leon-Aguilar and J. L. Quiroz-Arroyo, "Cryptography for privacy in a resource-constrained IoT: A systematic literature review," *IEIE Transactions on Smart Processing & Computing*, vol. 11, no. 5, pp. 351–360, 2022.

[11] K. Seyhan, T. N. Nguyen, S. Akleylek and K. Cengiz, "Lattice-based cryptosystems for the security of resource-constrained IoT devices in post-quantum world: A survey," *Cluster Computing*, vol. 25, no. 3, pp. 1729–1748, 2022.

[12] R. Raj and M. Ghosh, "A lightweight blockchain framework for secure transactions in resource constrained IoT devices," in *5th Int. Conf. on Recent Advances in Information Technology (RAIT)*, Dhanbad, India, IEEE, pp. 1–7, 2023.

[13] S. Li, H. Song and M. Iqbal, "Privacy and security for resource-constrained IoT devices and networks: Research challenges and opportunities," *Sensors*, vol. 19, no. 8, pp. 1935, 2019.

[14] A. S. Alluhaidan and P. Prabu, "End to end encryption in resource-constrained IoT device," *IEEE Access*, vol. 11, pp. 2169–3536, 2023.
[15] B. J. Mohd, T. Hayajneh and A. V. Vasilakos, "A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues," *Journal of Network and Computer Applications*, vol. 58, pp. 73–93, 2015.
[16] K. Tange, D. Howard, T. Shanahan, S. Pepe, X. Fafoutis *et al.,* "RTLS: Lightweight tls session resumption for constrained IoT devices," in *Information and Communications Security: 22nd Int. Conf.*, Copenhagen, Denmark, Springer, 2020.
[17] T. K. Goyal, V. Sahula and D. Kumawat, "Energy efficient lightweight cryptography algorithms for IoT devices," *IETE Journal of Research*, vol. 68, no. 3, pp. 1722–1735, 2022.
[18] A. Biryukov and L. Perrin, "State of the art in lightweight symmetric cryptography," *Cryptology ePrint Archive*, pp. 1–55, 2017.
[19] Z. Sheng, S. Yang, Y. Yu, A. V. Vasilakos, J. A. McCann *et al.,* "A survey on the IETF protocol suite for the Internet of Things: Standards, challenges, and opportunities," *IEEE Wireless Communications*, vol. 20, no. 6, pp. 91–98, 2013.
[20] E. R. Naru, H. Saini and M. Sharma, "A recent review on lightweight cryptography in IoT," in *2017 Int. Conf. on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, Tamil Nadu, India, IEEE, 2017.
[21] S. A. Ansar, S. Arya, S. Aggrawal, S. Saxena, A. Kushwaha *et al.,* "Security in IoT layers: Emerging challenges with countermeasures," in *Computer Vision and Robotics: Proc. of CVR 2022*, Singapore, Springer Nature Singapore, pp. 551–563, 2023.
[22] R. Almukhlifi and P. L. Vora, "Linear cryptanalysis of reduced-round simeck using super rounds," *Cryptography*, vol. 7, no. 1, pp. 8, 2023.
[23] S. Singh, P. K. Sharma, S. Y. Moon and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–18, 2017. https://doi.org/10.1007/s12652-017-0494-4
[24] I. Bhardwaj, A. Kumar and M. Bansal, "A review on lightweight cryptography algorithms for data security and authentication in iots," in *2017 4th Int. Conf. on Signal Processing, Computing and Control (ISPCC)*, Waknaghat, India, IEEE, 2017.
[25] C. Thorat and V. Inamdar, "Implementation of new hybrid lightweight cryptosystem," *Applied computing and Informatics*, vol. 1, no. 2, pp. 195–206, 2018.
[26] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou and C. Manifavas, "A review of lightweight block ciphers," *Journal of Cryptographic Engineering*, vol. 8, pp. 141–184, 2018.
[27] H. Chen, W. Wu and D. Feng, "Differential fault analysis on clefia," in *Information and Communications Security: 9th Int. Conf., ICICS 2007*, Zhengzhou, China, Springer, 2007.
[28] W. Stallings, "The principles and practice of cryptography and network security," *Pearson Education*, vol. 20, no. 1, pp. 7, 2017.
[29] G. Bansod, N. Raval and N. Pisharoty, "Implementation of a new lightweight encryption design for embedded security," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 142–151, 2014.
[30] G. Christodoulou, C. Chung, K. Ligett, E. Pyrga and R. van Stee, "On the price of stability for undirected network design," in *Approximation and Online Algorithms: 7th Int. Workshop, WAOA 2009*, Copenhagen Denmark, Springer, 2010.
[31] N. Mouha, B. Mennink, A. van Herrewege, D. Watanabe, B. Preneel *et al.,* "Chaskey: An efficient mac algorithm for 32-bit microcontrollers," in *Selected Areas in Cryptography–SAC 2014*, Montreal, QC, Canada, Springer, 2014.
[32] A. Luykx, B. Preneel, E. Tischhauser and K. Yasuda, "A mac mode for lightweight block ciphers," in *Fast Software Encryption: 23rd Int. Conf.*, Bochum, Germany, Springer, 2016.

[33] Z. Gong, P. Hartel, S. Nikova, S. H. Tang and B. Zhu, "Tulp: A family of lightweight message authentication codes for body sensor networks," *Journal of Computer Science and Technology*, vol. 29, pp. 53– 68, 2014.
[34] M. Hell, T. Johansson and W. Meier, "Grain: A stream cipher for constrained environments," *International journal of Wireless and Mobile Computing*, vol. 2, no. 1, pp. 86–93, 2007.
[35] C. de Canniere, "Trivium: A stream cipher construction inspired by block cipher design prin-ciples," in *Int. Conf. on Information Security*, Samos, Greece, Springer, 2006.
[36] S. Babbage and M. Dodd, "The mickey stream ciphers," in *New Stream Cipher Designs: The eSTREAM Finalists*, Springer, pp. 191–209, 2008. https://doi.org/10.1007/978-3-540-68351-3_15
[37] J. Kaur, A. C. Canto, M. M. Kermani and R. Azarderakhsh, "A comprehensive survey on the implementations, attacks, and countermeasures of the current nist lightweight cryptography standard," arXiv preprint arXiv:2304.06222, 2023.
[38] M. S. Turan, K. McKay, D. Chang, J. Kang, N. Waller *et al.,* "Status report on the final round of the nist lightweight cryptography standardization process," 2023. https://doi.org/10.6028/NIST.IR.8454
[39] S. Panasenko and S. Smagin, "Lightweight cryptography: Underlying principles and approaches," *International Journal of Computer Theory and Engineering*, vol. 3, no. 4, pp. 516, 2011.
[40] G. Choudhary, P. V. Astillo, I. You, K. Yim, R. Chen *et al.,* "Lightweight misbehavior detection management of embedded iot devices in medical cyber physical systems," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2496–2510, 2020.
[41] A. Poschmann, "Lightweight cryptography-cryptographic engineering for a pervasive world," *Cryptology ePrint Archive*, pp. 1–179, 2009.
[42] S. Kumar and D. Kumar, "Securing of cloud storage data using hybrid AES-ECC cryptographic approach," *Journal of Mobile Multimedia*, vol. 19, pp. 363–388, 2023.
[43] M. Çoban, F. Karakoç and Ö. Boztas, "Biclique cryptanalysis of twine," in *Cryptology and Network Security: 11th Int. Conf., CANS 2012*, Darmstadt, Germany, Springer, 2012.
[44] S. Kotel, M. Zeghid, M. Machhout and R. Tourki, "Lightweight encryption algorithm based on modified xtea for low-resource embedded devices," in *Proc. of the 21st Int. Database Engineering & Applications Symp.*, Bristol, UK, 2017.
[45] S. Kumar and D. Kumar, "A survey of lightweight cryptography for power-constrained IoT devices: Security challenges and issues," *Green Engineering and Technology*, pp. 293–313, 2021. https://doi.org/10.1201/9781003176275
[46] J. Daemen and V. Rijmen, *AES Proposal: Rijndael*. Gaithersburg, MD, USA: Scientific Research, 1999.
[47] A. Bogdanov, D. Khovratovich and C. Rechberger, "Biclique cryptanalysis of the full aes," in *Int. Conf. on the Theory and Application of Cryptology and Information Security*, Seoul, Korea, Springer, 2011.
[48] S. Kumar, D. Kumar and N. Singh, "Performance and security analysis using b-128 modified blowfish algorithm," *Multimedia Tools and Applications*, vol. 82, pp. 1–18, 2023.
[49] B. Koo, D. Roh, H. Kim, Y. Jung, D. G. Lee *et al.,* "Cham: A family of lightweight block ciphers for resource-constrained devices," in *Information Security and Cryptology–ICISC 2017*, Seoul, South Korea, Springer, 2018.
[50] Y. Li, W. Wu and L. Zhang, "Improved integral attacks on reduced-round clefia block cipher," in *Int. Workshop on Information Security Applications*, Jeju Island, Korea, Springer, 2011.
[51] C. Tezcan, "The improbable differential attack: Cryptanalysis of reduced round clefia," in *Int. Conf. on Cryptology*, India, Springer, 2010.
[52] T. Shirai, K. Shibutani, T. Akishita, S. Moriai and T. Iwata, "The 128-bit blockcipher clefia," in *Int. Workshop on Fast Software Encryption*, Luxembourg, Springer, 2007.
[53] W. Stallings, *Network Security Essentials: Applications and Standards*. USA: Pearson, 2017.
[54] M. Kumar, P. Sk and A. Panigrahi, "FeW: A lightweight block cipher," *Turkish Journal of Mathematics and Computer Science*, vol. 11, no. 2, pp. 58–73, 2014.

[55] Y. Tsunoo, E. Tsujihara, M. Shigeri, T. Suzaki and T. Kawabata, "Cryptanalysis of clefia using multiple impossible differentials," in *2008 Int. Symp. on Information Theory and its Applications*, Auckland, New Zealand, IEEE, 2008.
[56] D. Hong, B. Koo and D. Kwon, "Biclique attack on the full hight," in *Int. Conf. on Information Security and Cryptology*, Springer, 2011. https://doi.org/10.1007/978-3-642-31912-9_24
[57] O. Özen, K. Varıcı, C. Tezcan and Ç. Kocair, "Lightweight block ciphers revisited: Cryptanalysis of reduced round present and hight," in *Information Security and Privacy: 14th Australasian Conf.*, Brisbane, Australia, Springer, 2009.
[58] P. H. Nguyen, M. J. Robshaw and H. Wang, "On related-key attacks and kasumi: The case of A5/3," in *Progress in Cryptology–INDOCRYPT 2011: 12th Int. Conf. on Cryptology in India*, Chennai, India, Springer, 2011.
[59] T. Saito, "A single-key attack on 6-round kasumi," *Cryptology ePrint Archive*, pp. 1–13, 2011.
[60] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari *et al.,* "Midori: A block cipher for low energy," in *Advances in Cryptology–ASIACRYPT 2015: 21st Int. Conf. on the Theory and Application of Cryptology and Information Security*, Auckland, New Zealand, Springer, 2015.
[61] Z. Gong, S. Nikova and Y. W. Law, "KLEIN: A new family of lightweight block ciphers," in *Int. Workshop on Radio Frequency Identification: Security and Privacy Issues*, Springer, 2011. https://doi.org/10.1007/978-3-642-25286-0_1
[62] V. Lallemand and M. Naya-Plasencia, "Cryptanalysis of klein," in *Int. Workshop on Fast Software Encryption*, Springer, 2014. https://doi.org/10.1007/978-3-662-46706-0_23
[63] S. F. Abdul-Latip, M. R. Reyhanitabar, W. Susilo and J. Seberry, "Fault analysis of the katan family of block ciphers," in *Information Security Practice and Experience: 8th Int. Conf., ISPEC 2012*, Hangzhou, China, Springer, 2012.
[64] N. A. N. Abdullah, N. H. Lot, A. Zawawi and H. A. Rani, "Analysis on lightweight block cipher ktantan," in *2011 7th Int. Conf. on Information Assurance and Security (IAS)*, Melacca, Malaysia, IEEE, 2011.
[65] L. Wei, C. Rechberger, J. Guo, H. Wu, H. Wang *et al.,* "Improved meet-in-the-middle cryptanalysis of ktantan (poster)," in *Information Security and Privacy: 16th Australasian Conf.*, Melbourne, Australia, Springer, 2011.
[66] W. Wu and L. Zhang, "Lblock: A lightweight block cipher," in *Int. Conf. on Applied Cryptography and Network Security*, Springer, 2011. https://doi.org/10.1007/978-3-642-21554-4_19
[67] S. Liu, Z. Gong and L. Wang, "Improved related-key differential attacks on reduced-round lblock," in *Information and Communications Security: 14th Int. Conf., ICICS 2012*, Hong Kong, China, Springer, 2012.
[68] N. Wang, X. Wang and K. Jia, "Improved impossible differential attack on reduced-round lblock," in *ICISC 2015*, Springer, 2015. https://doi.org/10.1007/978-3-319-30840-1_9
[69] L. Wen, M. Q. Wang and J. Y. Zhao, "Related-key impossible differential attack on reduced-round lblock," *Journal of Computer Science and Technology*, vol. 29, no. 1, pp. 165–176, 2014.
[70] J. Guo, T. Peyrin, A. Poschmann and M. Robshaw, "The led block cipher," in *Int. Workshop on Cryptographic Hardware and Embedded Systems*, 2011. https://doi.org/10.1007/978-3-642-23951-9_22
[71] C. Dobraunig, M. Eichlseder, D. Kales and F. Mendel, "Practical key-recovery attack on mantis5," *IACR Transactions on Symmetric Cryptology*, vol. 2016, no. 2, pp. 248–260, 2016. https://doi.org/10.13154/tosc.v2016.i2.248-260
[72] J. Cui, J. Guo, Y. Huang and Y. Liu, "Improved meet-in-the-middle attacks on crypton and mcrypton," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 11, no. 5, pp. 2660–2679, 2017.
[73] K. Shibutani, T. Isobe, H. Hiwarati, A. Mitsuda, T. Akishita *et al.,* "*Piccolo*: An ultra-lightweight blockcipher," in *Cryptographic Hardware and Embedded Systems CHES 2011*, pp. 342–357, 2011. https://doi.org/10.1007/978-3-642-23951-9_23
[74] M. Minier, "On the security of piccolo lightweight block cipher against related-key impossible differentials," in *Progress in Cryptology–INDOCRYPT 2013*, pp. 308–318, Mumbai, India, Springer, 2013.

[75] Y. Wang, W. Wu and X. Yu, "Biclique cryptanalysis of reduced-round piccolo block cipher," in *Information Security Practice and Experience: 8th Int. Conf.*, Hangzhou, China, Springer, 2012.
[76] J. Y. Cho, "Linear cryptanalysis of reduced-round present," in *Topics in Cryptology-CT-RSA 2010: The Cryptographersâ Track at the RSA Conf. 2010*, San Francisco, CA, USA, Springer, 2010.
[77] L. Li, B. Liu and H. Wang, "QTL: A new ultra-lightweight block cipher," *Microprocessors and Microsystems*, vol. 45, pp. 45–55, 2016.
[78] M. Çoban, F. Karakoç and M. Özen, "Cryptanalysis of QTL block cipher," in *Int. Workshop on Lightweight Cryptography for Security and Privacy*, Springer, 2016. https://doi.org/10.1007/978-3-319-55714-4_5
[79] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang *et al.,* "Rectangle: A bit-slice lightweight block cipher suitable for multiple platforms," *Science China Information Sciences*, vol. 58, no. 12, pp. 1–15, 2015.
[80] J. Lu, "Related-key rectangle attack on 36 rounds of the xtea block cipher," *International Journal of Information Security*, vol. 8, pp. 1–11, 2009.
[81] A. Baysal and S. Sahin, "Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors," in *Lightweight Cryptography for Security and Privacy*, Springer, 2015. https://doi.org/10.1007/978-3-319-29078-2_4
[82] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks *et al.,* "The simon and speck lightweight block ciphers," in *Proc. of the 52nd Annual Design Automation Conf.*, 2015. https://doi.org/10.1007/978-3-319-29078-2_4
[83] A. Biryukov, A. Roy and V. Velichkov, "Differential analysis of block ciphers simon and speck," in *Int. Workshop on Fast Software Encryption*, Springer, 2014. https://doi.org/10.1007/978-3-662-46706-0_28
[84] J. Kelsey, B. Schneier and D. Wagner, "Related-key cryptanalysis of 3WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA," in *Int. Conf. on Information and Communications Security*, Springer, 1997. https://doi.org/10.1007/BFb0028479
[85] R. M. Needham and D. J. Wheeler, "*Tea Extensions*," Cambridge University, Cambridge, UK, 1997.
[86] T. Suzaki, K. Minematsu, S. Morioka and E. Kobayashi, "TWINE: A lightweight, versatile block cipher," in *ECRYPT Workshop on Lightweight Cryptography*, vol. 2011, Springer Berlin, Heidelberg, 2011.
[87] S. Maitra and K. Yelamarthi, "Rapidly deployable iot architecture with data security: Implemen- tation and experimental evaluation," *Sensors*, vol. 19, no. 11, pp. 2484, 2019.
[88] S. Hong, D. Hong, Y. Ko, D. Chang, W. Lee *et al.,* "Differential cryptanalysis of tea and xtea," in *Int. Conf. on Information Security and Cryptology*, Springer, 2003. https://doi.org/10.1007/978-3-540-24691-6_30
[89] M. Rana, Q. Mamun and R. Islam, "Lightweight cryptography in iot networks: A survey," *Future Generation Computer Systems*, vol. 129, pp. 77–89, 2022.
[90] I. K. Dutta, B. Ghosh and M. Bayoumi, "Lightweight cryptography for Internet of Insecure Things: A survey," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conf. (CCWC)*, Las Vegas, NV, USA, IEEE, 2019.
[91] S. B. Sadkhan and A. O. Salman, "A survey on lightweight-cryptography status and future challenges," in *2018 Int. Conf. on Advance of Sustainable Engineering and its Application (ICASEA)*, Wasit-Kut, Iraq, IEEE, 2018.
[92] Nayancy, S. Dutta and S. Chakraborty, "A survey on implementation of lightweight block ciphers for resource constraints devices," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 5, pp. 1377–1398, 2022.
[93] M. Cazorla, K. Marquet and M. Minier, "Survey and benchmark of lightweight block ciphers for wireless sensor networks," in *2013 Int. Conf. on Security and Cryptography (SECRYPT)*, Reykjavik, Iceland, IEEE, 2015.
[94] P. Singh, B. Acharya and R. K. Chaurasiya, "A comparative survey on lightweight block ciphers for resource constrained applications," *International Journal of High Performance Systems Architecture*, vol. 8, no. 4, pp. 250–270, 2019.
[95] M. A. Philip and Vaithiyanathan, "A survey on lightweight ciphers for IoT devices," in *2017 Int. Conf. on Technological Advancements in Power and Energy (TAP Energy)*, Kollam, India, IEEE, 2017.

[96] J. Daemen and V. Rijmen, "The rijndael block cipher: AES proposal," in *First Candidate Conf. (AeS1)*, Japan, Springer, 1999.
[97] A. Biryukov and D. Khovratovich, "Related-key cryptanalysis of the full AES-192 and AES-256," in *Advances in Cryptology–ASIACRYPT 2009: 15th Int. Conf. on the Theory and Application of Cryptology and Information Security*, Tokyo, Japan, Springer.
[98] P. Barreto and V. Rijmen, "The khazad legacy-level block cipher," *Primitive Submitted to NESSIE*, vol. 97, no. 106, pp. 106, 2000.
[99] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari *et al.,* "Midori: A block cipher for low energy," *IEICE Technical Report*, vol. 116, no. 35, pp. 45, 2016.
[100] C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi *et al.,* "The skinny family of block ciphers and its low-latency variant mantis," in *Advances in Cryptology–CRYPTO 2016: 36th Annual Int. Cryptology Conf.*, Santa Barbara, CA, USA, Springer, 2016.
[101] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann *et al.,* "Present: An ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems-CHES 2007: 9th Int. Workshop*, Vienna, Austria, Springer, 2007.
[102] J. Borghoff, A. Canteaut, T. Gneysu, E. B. Kavun, M. Knezevic *et al.,* "Prince-a low- latency block cipher for pervasive computing applications," in *Advances in Cryptology ASIACRYPT 2012: 18th Int. Conf. on the Theory and Application of Cryptology and Information Security*, Beijing, China, 2012.
[103] A. Canteaut, M. Naya-Plasencia and B. Vayssiere, "Sieve-in-the-middle: Improved mitm attacks," in *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conf.*, Santa Barbara, CA, USA, Springer, 2013.
[104] H. Soleimany, C. Blondeau, X. Yu, W. Wu, K. Nyberg *et al.,* "Reflection cryptanalysis of prince-like ciphers," *Journal of Cryptology*, vol. 28, pp. 718–744, 2015.
[105] S. Sadeghi, N. Bagheri and M. A. Abdelraheem, "Cryptanalysis of reduced qtl block cipher," *Microprocessors and Microsystems*, vol. 52, pp. 34–48, 2017.
[106] M. Wang, "Differential cryptanalysis of reduced-round present," in *Lecture Notes in Computer Science*, vol. 5023, pp. 40–49, 2008.
[107] D. Yang, W. F. Qi and H. J. Chen, "Impossible differential attacks on the skinny family of block ciphers," *IET Information Security*, vol. 11, no. 6, pp. 377–385, 2017.
[108] J. Chen, M. Wang and B. Preneel, "Impossible differential cryptanalysis of the lightweight block ciphers tea, xtea and hight," in *Progress in Cryptology-AFRICACRYPT 2012: 5th Int. Conf. on Cryptology*, Africa, Ifrance, Morocco, Springer, 2012.
[109] D. J. Wheeler and R. M. Needham, "Tea, a tiny encryption algorithm," in *Int. Workshop on Fast Software Encryption*, Leuven, Belgium, Springer, 1994.
[110] G. Sekar, N. Mouha, V. Velichkov and B. Preneel, "Meet-in-the-middle attacks on reduced-round xtea," in *Topics in Cryptology–CT-RSA 2011*, San Francisco, CA, USA, Springer, 2011.
[111] D. Hong, B. Koo and D. Kwon, "Biclique attack on the full hight," in *Information Security and Cryptology-ICISC 2011: 14th Int. Conf.*, Seoul, Korea, Springer, 2012.
[112] N. Im, S. Choi and H. Yoo, "S-box attack using fpga reverse engineering for lightweight cryptography," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25165–25180, 2022.
[113] G. Fragkos, C. Minwalla, J. Plusquellic and E. E. Tsiropoulou, "Artificially intelligent electronic money," *IEEE Consumer Electronics Magazine*, vol. 10, no. 4, pp. 81–89, 2020.
[114] S. Sallam and B. D. Beheshti, "A survey on lightweight cryptographic algorithms," in *TENCON 2018-2018 IEEE Region 10 Conf.*, Jeju, Korea (South), IEEE, 2018.
[115] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang *et al.,* "Rectangle: A bit-slice lightweight block cipher suitable for multiple platforms," *Cryptology ePrint Archive*, 2014. https://doi.org/10.1007/s11432-015-5459-7
[116] J. Breier, X. Hou and Y. Liu, "Fault attacks made easy: Differential fault analysis automation on assembly code," *Cryptology ePrint Archive*, pp. 1–27, 2017.
[117] S. Kumar, D. Kumar and H. S. Lamkuche, "TPA auditing to enhance the privacy and security in cloud systems," *Journal of Cyber Security and Mobility*, vol. 10, pp. 537–568, 2021.

[118] G. P. Kachare, G. Choudhary, S. K. Shandilya and V. Sihag, "Sandbox environment for real time malware analysis of IoT devices," in *Int. Conf. on Computing Science, Communication and Security*, Cham, Springer International Publishing, pp. 169–183, 2022.
[119] S. Rahmadika, P. V. Astillo, G. Choudhary, D. G. Duguma, V. Sharma *et al.,* "Blockchain- based privacy preservation scheme for misbehavior detection in lightweight IoMT devices," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 710–721, 2022.
[120] B. Larry, M. Kerry, M. Nicky and T. Meltem, "*Report on Lightweight Cryptography*," National Institute of Standards and Technology, 2017.
[121] S. S. M. AlDabbagh, "Design 32-bit lightweight block cipher algorithm (DLBCA)," *International Journal of Computer Applications*, vol. 166, no. 8, pp. 17–20, 2017.
[122] A. Biryukov, A. Roy and V. Velichkov, "Differential analysis of block ciphers simon and speck," in *Fast Software Encryption: 21st Int. Workshop, FSE 2014*, London, UK, Springer, 2015.
[123] A. Bar-On and N. Keller, "A attack on the full MISTY1," in *Annual Int. Cryptology Conf.*, Rome, Italy, Springer, 2018.
[124] Y. Todo, "Integral cryptanalysis on full MISTY1," *Journal of Cryptology*, vol. 30, no. 3, pp. 920–959, 2017.
[125] M. Favaretto, T. Tran Anh, J. Kavaja, M. de Donno and N. Dragoni, "When the price is your privacy: A security analysis of two cheap IoT devices," in *Proc. of 6th Int. Conf. in Software Engineering for Defence*, Rome, Italy, Springer, 2018.
[126] A. Giaretta, N. Dragoni and F. Massacci, "S×C4IoT: A security-by-contract framework for dynamically evolving IoT devices," *ACM Transactions on Sensor Networks (TOSN)*, vol. 18, no. 1, pp. 1–51, 2021.
[127] E. Bejder, A. K. Mathiasen, M. de Donno, N. Dragoni and X. Fafoutis, "Shake: Shared acceleration key establishment for resource-constrained IoT devices," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, New Orleans, LA, USA, IEEE, 2020.
[128] A. Khan, G. Choudhary, S. K. Shandilya, D. M. Sharma and A. K. Sharma, "A hybrid mechanism for advance IoT malware detection," in *Int. Conf. on IoT, Intelligent Computing and Security*, Greater Noida, India, Springer, 2023.