A Comprehensive Review on Lightweight Cryptographic Mechanisms for Industrial Internet of Things Systems

SAAD KHAN, CISUC, Department of Informatic Engineering, University of Coimbra, Coimbra, Portugal
PEDRO AFONSO FERREIRA LOPES MARTINS, CISUC, Department of Informatic Engineering, University of Coimbra, Coimbra, Portugal
BRUNO SOUSA, CISUC, Department of Informatic Engineering, University of Coimbra, Coimbra, Portugal
VASCO PEREIRA, CISUC, Department of Computer Science, Universidade de Coimbra, Coimbra, Portugal

The integration of Industrial Internet of Things (IIoT) devices within Industrial Control Systems (ICS) presents significant cybersecurity challenges, primarily due to the limited resources of these devices. Traditional cryptographic algorithms are often unsuitable for IIoT environments due to their high computational, memory, and energy requirements. Lightweight Cryptographic Algorithms have emerged as efficient and secure alternatives, specifically designed for resource-constrained environments. This article systematically reviews lightweight symmetric cryptographic mechanisms, specifically Block and Stream ciphers, and evaluates their critical attributes from an IIoT perspective. In addition, the security strengths and vulnerabilities of these algorithms against known cryptanalytic attacks, including Differential, Linear, Related Key, and others, are discussed. The article also discusses current standardization efforts by organizations such as the National Institute of Standards and Technology (NIST), International Organization for Standardisation (ISO)/International Electrotechnical Commission (IEC), highlighting their applicability in ICS environments. Finally, it identifies open research issues and future directions for improving lightweight cryptographic security in ICS, providing valuable insights for security practitioners and researchers seeking to robustly secure IIoT deployments.

CCS Concepts: • **Security and privacy → Block and stream ciphers**; **Cryptanalysis and other attacks**; *Embedded systems security*; • **General and reference → Surveys and overviews;**

Additional Key Words and Phrases: Lightweight encryption algorithms, symmetric encryption, lightweight cryptography, block cipher, stream cipher, cryptanalysis of ciphers, industrial IoT systems, industrial control systems, security

This work has been supported by Project "Agenda Mobilizadora Sines Nexus." Reference No. 7113), supported by the Recovery and Resilience Plan (PRR) and by the European Funds Next Generation EU, following Notice No. 02/C05-i01/2022, Component 5 - Capitalisation and Business Innovation - Mobilising Agendas for Business Innovation. This work is also funded by national funds through the FCT - Fundação para a Ciência e a Tecnologia, I.P., in the framework of the Project UIDB/00326/2025 and UIDP/00326/2025.

Authors' Contact Information: Saad Khan, CISUC, Department of Informatic Engineering, University of Coimbra, Coimbra, Coimbra, Portugal; e-mail: saadkhan@dei.uc.pt; Pedro Afonso Ferreira Lopes Martins, CISUC, Department of Informatic Engineering, University of Coimbra, Coimbra, Coimbra, Portugal; e-mail: pedromartins@student.dei.uc.pt; Bruno Sousa, CISUC, Department of Informatic Engineering, University of Coimbra, Coimbra, Coimbra, Portugal; e-mail: bmsousa@dei.uc.pt; Vasco Pereira, CISUC, Department of Computer Science, Universidade de Coimbra, Coimbra, Coimbra, Portugal; e-mail: vasco@dei.uc.pt.

ACM 0360-0300/2025/09-ART23
https://doi.org/10.1145/3757734

ACM Reference Format:
Saad Khan, Pedro Afonso Ferreira Lopes Martins, Bruno Sousa, and Vasco Pereira. 2025. A Comprehensive Review on Lightweight Cryptographic Mechanisms for Industrial Internet of Things Systems. *ACM Comput. Surv.* 58, 1, Article 23 (September 2025), 37 pages. https://doi.org/10.1145/3757734

1 Introduction

For decades, **Industrial Control Systems (ICS)** have relied on sensors, actuators, and microcontrollers. The emergence of **Industrial Internet of Things** (**IIoT**) has transformed them with high-speed connectivity, real-time analytics, and intelligent automation. This transformation enables seamless monitoring, remote control, and data-driven decision making, improving efficiency, adaptability, and operational intelligence in industrial environments such as manufacturing, energy grids, transportation networks, and maritime logistics [1–3]. It is also crucial to know that this integration into ICS introduces cybersecurity vulnerabilities due to its resource-constrained devices, real-time communication protocols, and distributed architectures. These risks must be carefully managed to ensure the safety and security of industrial systems [4].

The security landscape of IIoT-driven ICS is different from traditional Operational Technology (OT) networks. Unlike conventional computing systems with ample computational power and storage capacity to implement cryptographic security, IIoT devices operate under stringent resource constraints such as memory and energy [5]. Traditional encryption schemes require substantial computational overhead, memory, and power, which IIoT devices in ICS environments cannot support [6]. Industrial-grade embedded systems such as Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), and industrial edge computing units handle automation, sensors networks, and secure data transmission [7]. These systems often operate on low-power microcontrollers such as ARM Cortex-M and AVR MCUs, commonly featuring memory constraints, with Flash memory (ROM) and SRAM (RAM) capacities in typical IIoT-focused deployments generally ranging from 32 KB to 128 KB [8, 9].

To ensure secure communication, protocols such as Message Queuing Telemetry Transport (MQTT), Long Range Wide Area Network (LoRaWAN), Profibus, and Modbus are used, which do not have built-in security by default; thus, additional security measures (e.g., MQTT over Transport Layer Security (TLS), cryptographic implementations, or gateway-based security controls) are essential to ensure robust and secure communication [10, 11]. Moreover, security resilience is essential, as IIoT deployments are frequently exposed to sophisticated cryptanalytic attacks, such as Differential Cryptanalysis, Side-channel Attacks (SCA) [12], Related Key Attacks, Man-in-the-Middle (MITM) Attacks [13], Biclique Attacks, and linear cryptanalysis, all of which pose severe risks in IIoT-based ICS environments [14].

Symmetric algorithms in Lightweight Cryptography (LWC) were introduced and standardized by authoritative standardization bodies and research communities to ensure real-time performance, low-latency communication, and resilience against cyber threats without excessive overhead, while asymmetric algorithms are excluded due to their computational complexity, which makes them less suitable for resource-constrained IIoT devices. In this research, we evaluated LWC according to established standards. Based on our analysis of the results from **National Institute of Standards and Technology** (**NIST**)'s LWC Standardization competition (2018–2023) [15], and **International Organization for Standardisation** (**ISO**)/**International Electro-technical Commission** (**IEC**) 29192 [16], we conclude that, as of today, the minimum acceptable key size is 80 bits, whereas 128 bits is recommended. The size of the round ranges from a minimum size of 6 rounds to a maximum of 40 rounds, ensuring a balance between security and efficiency for IIoT applications [15, 16]. A full definition of Lightweight Encryption is provided in Definition 3.2,

Table 1. Research Questions for Systematic Literature Review

No.	Question	Rationale	Reference
RQ1	How can ICS security practitioners identify the most suitable lightweight cryptographic algorithm for their IIoT environment?	The answer helps to select most suitable LWC algorithm based on security, feasibility, standard compliance, and adaptability in IIoT environment.	Section 3.2 and 3.3 Tables 4 5, 6, 7, and 8
RQ2	How robust are lightweight cryptographic algorithms against common cryptographic attacks?	This enables the evaluation of the robustness of lightweight cryptographic algorithms against common cryptanalytic attacks, thereby addressing their security effectiveness in IIoT applications.	Section 3.4, 3.2, 3.3, and Table 4
RQ3	Which security research gaps and challenges are yet to be solved in IIoT systems?	The answer highlights the critical issues, and help researchers to identify challenges and possible future directions.	Section 5

in Section 3.1. This article proposes a systematic review to provide an in-depth understanding of LWC by classifying symmetric algorithms (block and stream ciphers) based on their cryptographic properties, deployment feasibility in resource-constrained environments, and security resilience against cryptanalytic threats. To systematically address these aspects, we formulate three research questions, as outlined in Table 1.

1.1 Motivation

As IIoT continues to expand into the critical domain of ICS, ensuring robust security remains a key challenge. Although LWC has emerged as a promising solution for protecting resource-constrained IIoT environments, a comprehensive analysis specifically tailored to its deployment within ICS is still lacking. Numerous surveys have been conducted on lightweight cryptographic algorithms. Thakor et al. [17] compared 41 symmetric lightweight cryptographic algorithms, evaluating them on critical performance metrics such as block/key size, memory, gate area, latency, throughput, power and energy requirements, and software and hardware efficiency. Rana et al. [18] examined lightweight algorithms in the context of Internet of Things (IoT) network security, comparing them based on key metrics such as key size, block size, and rounds. Similarly, Agrawal et al. [19] examined 17 cryptographic methods, emphasizing the security of the embedded systems. Furthermore, Kong et al. [20] significantly expand the scope, summarizing 100 cryptographic algorithms from both hardware and software perspectives for resource-constrained devices used in wireless sensor networks and IoT. Ekwueme et al. [21] examined various lightweight cryptographic block cipher algorithms to secure IoT devices, discussing their advantages in constrained environments. They evaluate cryptographic algorithms based on block size, key size, number of rounds, security efficiency, computational overhead, hardware vs. software implementation, and attack resistance. Mousavi et al. [22] explored symmetric, asymmetric, and hybrid encryption techniques to secure IoT networks and mitigate cyber threats. Cryptographic methods are assessed based on confidentiality, integrity, authentication, authorization, availability, non-repudiation, accountability, anonymity, and security against cryptographic attacks. Rao et al. [23] investigated lightweight cryptographic solutions specifically designed for IoT devices, focusing on authentication and data integrity. They evaluated Elliptic Curve Cryptography (ECC) encryption techniques based on key size, power efficiency, computational complexity, optimizations, and software vs. hardware performance. Naser et al. [24] reviewed ultra-lightweight encryption algorithms, focusing on their applicability in IoT environments with stringent power and memory constraints. They evaluated encryption schemes based on key size, Initialization vector (IV), energy efficiency, computational overhead, resistance to cryptanalytic attacks, and performance in embedded systems. Extensive research has been conducted on lightweight cryptographic algorithms, focusing mainly on their applicability in general IoT ecosystems. These studies often emphasize security concerns such as data privacy and user authentication while overlooking stringent security requirements, real-time operational constraints, and regulatory compliance challenges, which are unique to IIoT-driven ICS environments. Unlike consumer IoT, IIoT systems operate in mission-critical infrastructures where security breaches can lead to severe operational disruptions, making cryptographic efficiency, attack resilience, and industry-standard compliance essential. Our study directly addresses these industrial challenges by systematically evaluating LWC algorithms through the lens of IIoT security, standardization, and implementation feasibility in resource-constrained environments. Despite the existing body of work on lightweight cryptographic algorithms, there remain significant gaps when it comes to their application in IIoT use cases. The identified gaps are as follows:

(1) Existing surveys focus on individual cipher types but lack a holistic assessment that integrates cryptographic properties, implementation constraints, and practical deployment challenges, limiting a well-rounded understanding of their suitability for resource-constrained IIoT-driven environments.
(2) Industrial cybersecurity standards such as NIST, and ISO/IEC, are crucial and widely adopted in ICS to ensure security and interoperability. However, the compliance of existing LWC algorithms with these standards remains under-explored, creating challenges for adoption, regulatory alignment, and secure deployment in constrained IIoT systems.
(3) The literature lacks a systematic evaluation of cryptanalytic vulnerabilities in LWC schemes, particularly their resilience against advanced attack vectors in IIoT-driven systems.

As identified in our study, these gaps hinder industrial stakeholders from making informed security decisions, leaving IIoT systems vulnerable to evolving cyber threats and constraining the development of robust cryptographic defenses.

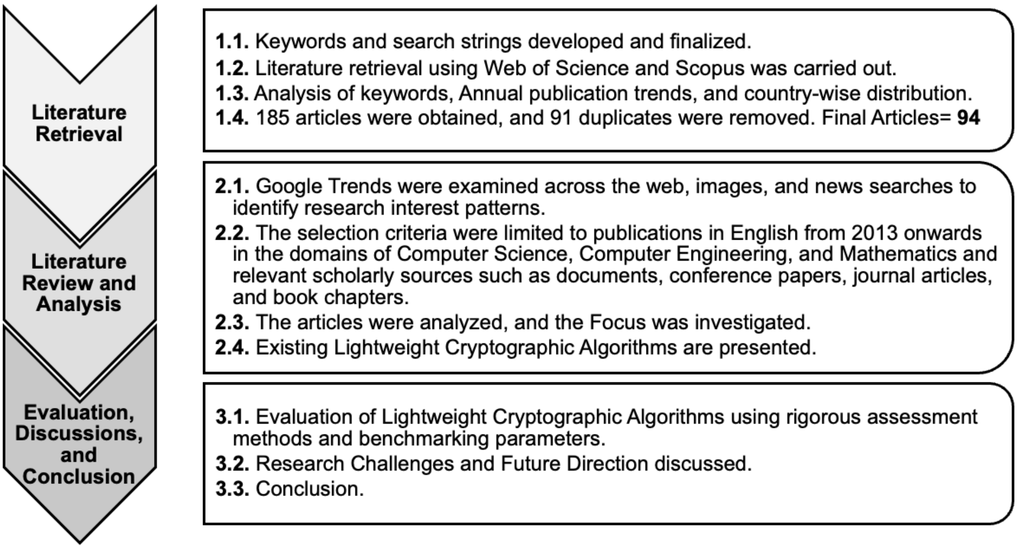
1.2 Major Contributions

We aim to provide a comprehensive state-of-the-art review by systematically analyzing lightweight cryptographic algorithms for resource-constrained devices between 2013 and 2024. The Major contributions of our survey are:

(1) Detailed overview of lightweight encryption algorithms for IIoT-driven systems. Additionally, this study provides essential background knowledge on LWC, offering a foundational understanding of its principles, design considerations, and critical features.
(2) Comprehensive information of diverse symmetric lightweight encryption algorithms by discussing 24 block ciphers and 13 stream ciphers focusing on key attributes, such as cryptographic parameters, real-world implementation feasibility, and alignment with established standards for security in IIoT environment.
(3) Identification of the vulnerability landscape of lightweight cryptographic algorithms in the IIoT, by presenting possible cyber threats related to cryptanalytic attacks in IIoT environments.
(4) Identification of critical open issues and research challenges in securing IIoT systems.

1.3 Article Organization

Section 2 presents the methodology used to carry out our comprehensive review. Section 3 discusses the definitions of the document along with a detailed description of Block and Stream Ciphers. It also discusses the cryptanalysis of ciphers and the hardware platform choice for the Lightweight Cryptographic Algorithm's implementation on IIoT systems. Section 4 presents a detailed review of Block and Stream ciphers with all their important attributes related to IIoT-driven ICS and their compliance with industrial standards. Section 5 identifies open issues, research challenges, and future directions. Finally, Section 6 draws conclusions.


Literature Retrieval
Literature Review and Analysis
Evaluation, Discussions, and Conclusion

1.1. Keywords and search strings developed and finalized.
1.2. Literature retrieval using Web of Science and Scopus was carried out.
1.3. Analysis of keywords, Annual publication trends, and country-wise distribution.
1.4. 185 articles were obtained, and 91 duplicates were removed. Final Articles= **94**

2.1. Google Trends were examined across the web, images, and news searches to identify research interest patterns.
2.2. The selection criteria were limited to publications in English from 2013 onwards in the domains of Computer Science, Computer Engineering, and Mathematics and relevant scholarly sources such as documents, conference papers, journal articles, and book chapters.
2.3. The articles were analyzed, and the Focus was investigated.
2.4. Existing Lightweight Cryptographic Algorithms are presented.

3.1. Evaluation of Lightweight Cryptographic Algorithms using rigorous assessment methods and benchmarking parameters.
3.2. Research Challenges and Future Direction discussed.
3.3. Conclusion.

Fig. 1. Research methodology.

2 Systematic Literature Retrieval Process

The current study followed the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology [25] for conducting a systematic review of the literature, as shown in Figure 1. Ten essential elements of PRISMA guidelines were followed for conducting the systematic review procedure:

(1) As shown in Table 2 and Figure 1, the review is based on a keyword-based literature selection published over the past 12 years.
(2) To be eligible for inclusion in the current study, keywords must appear in the title, abstract, or keyword sections of the article.
(3) The information sources are from the Web of Science (WoS) and Scopus repositories.[1]
(4) The search strings specified in Table 2 were used for the literature search.
(5) The research selection procedure included screening, and deleting duplicates, as well as qualitative analysis (reading abstracts and keywords) and quantitative analysis of Lightweight Cryptographic Algorithms for IIoT Systems.
(6) The articles obtained from Scopus and WoS were examined in detail, the keywords matched, and the relevant information regarding Lightweight Cryptographic Algorithms and IIoT Systems was collected for the data collection or extraction procedure.
(7) The data items included a Google Trends analysis encompassing web, image, and news searches conducted globally from 1 January 2013 to 31 December 2024, to identify patterns of research interest based on relevant keywords and search strings.
(8) The risk of bias in individual studies did not affect the review process. A clearer, non biased counting method was adopted.
(9) To ensure consistency, the findings of this study were compared with those of other studies (mainly articles referenced in Section 1.2).
(10) All studies included in the review were assessed under consistent criteria to minimize the risk of bias across investigations. Nonetheless, some relevant articles may not have been

[1] The following repositories were considered: (i) Web of Science, and (ii) Scopus.

Table 2. Literature Search Engines, Search Strings, Steps, and Results

Search Engine	Search Strings	Results
Scopus	(TITLE-ABS-KEY ("Industrial IoT") OR TITLE-ABS-KEY ("Industry 4.0") OR TITLE-ABS-KEY ("Lightweight Cryptographic Algorithms") OR TITLE-ABS-KEY ("Smart Industry") OR TITLE-ABS-KEY ("Internet of Things in Smart Industries") OR TITLE-ABS-KEY ("Cryptographic Algorithms for Lightweight IIoT Devices")) AND PUBYEAR >2013	Basic Search: 3921 Time Limit: 3921 Article Limit: 2655 Lightweight-Cryptographic Algorithms: 142
Web of Science (WoS) Time Span: 2013-2024. Indexes: SCI-EXPANDED, CPCI-S, ESCI,SSCI, BKCI-S, CPCI-SSH, BKCI-SSH. Language: English.	("Industrial IoT") OR ALL FIELDS: ("Industry 4.0") OR ALL FIELDS: ("Lightweight Cryptographic Algorithms") OR ALL FIELDS: ("Smart Industry") OR ALL FIELDS: ("Internet of Things in Smart Industries") OR ALL FIELDS: ("Cryptographic Algorithms for Lightweight IIoT Devices")	Basic Search: 1415 Time Limit: 1415 Article Limit: 108 Lightweight-Cryptographic Algorithms: 43
		Total Retrieved: 185 Duplicates: 91 Final Shortlisted : 94

retrieved due to strict inclusion/exclusion criteria and limitations inherent to keyword-based search and database indexing.

All research articles in this study were double-checked, and only articles focused on Lightweight Cryptographic Algorithms were shortlisted, retrieving 185 articles: 142 from Scopus and 43 from WoS. These retrieved articles were cross-checked to remove duplications; as a result, 91 articles were removed because they appeared in both sources. This reduced the total number of research articles included in the current study to 94, highlighting the limited yet growing body of literature in this emerging domain.

3 Background

ICS follows a segmented network architecture, as described in NIST SP 800-82 [26], which aligns with the hierarchical structure of the Purdue Enterprise Reference Architecture (PERA), organizing components from Level 0 (physical sensing and actuation) to Level 4/5 (enterprise management and cloud integration) [27, 28]. At Level 0, IIoT devices interact directly with physical processes, making their security crucial. Due to the constraints in computational power, asymmetric encryption proves to be inefficient, requiring the adoption of symmetric encryption techniques for effective safeguarding [29]. Recognizing these issues, cybersecurity standards establish security frameworks to mitigate IIoT-related risks in industrial settings. Table 3 provides an overview of these critical standards, highlighting their importance in protecting IIoT devices and industrial networks from emerging cyber threats. Meanwhile, lightweight cryptographic algorithms offer advantages to multiple sectors, with distinct algorithms designed for specific applications such as automotive and manufacturing. These encryption algorithms provide benefits such as reduced computational overhead, higher processing rates, and lower energy consumption, which is critical for battery-operated devices. Furthermore, the streamlined design of these algorithms leads to

Table 3. Cybersecurity Standards Applicable for IIoT: Data Security, Communication Protocols in ICS

Standard Number	**Use Case (IIoT Security Context)**
IEC 62443 [30, 31]	Provide guidance on using ISA/IEC 62443 for IIoT projects.
IEC 62351-3:2024 [32]	Specifies the use of TLS to secure communication over TCP/IP-based protocols used in power systems. Ensures authentication, confidentiality, and integrity for data flows.
ISO/SAE 21434:2021 [33]	Specifies cybersecurity risk management for connected and autonomous vehicles, addressing embedded systems, in-vehicle networks, and external communication interfaces.
IEEE 2030.5-2023 [34]	Enables secure and scalable communication between distributed energy systems and management platforms, making it applicable to smart grid infrastructures.
IMO-MSC-FAL.1/Circ.3 [35]	Provides cybersecurity measures for maritime operations, securing vessels and port facilities.
NIST SP 800-82 [26]	Provides security architecture and best practices for protecting ICS environments, including guidance on risk mitigation, layered defense, and incident response.
ENISA [36]	Offers IoT cybersecurity guidelines for various critical infrastructure sectors, including manufacturing and transportation

reduced hardware complexity and lower implementation costs, making them highly attractive to enterprises seeking cost-effective yet robust cryptographic solutions.

3.1 Definitions

This section presents definitions relevant to understanding the key concepts included in this survey.

Definition 3.1 (Lightweight Devices). Lightweight devices refer to highly resource-constrained platforms characterized by limited computational capabilities, restricted memory (both RAM and ROM), and low energy availability. According to NISTIR 8114, examples include low-power microcontrollers, embedded systems, RFID tags, and constrained IIoT sensors [37].

Definition 3.2 (Lightweight Cryptography). According to NISTIR 8114 [37], lightweight cryptography is explicitly designed for environments where devices have limitations in processing power, memory, and energy availability, like lightweight devices (as per Definition 3.1). These algorithms provide the same fundamental security properties as conventional cryptographic methods (i.e., confidentiality, integrity, and authentication) while optimized for minimal computational and memory overhead. ISO/IEC 29192 standard series also specifies cryptographic primitives that are feasible for constrained environments. The minimum acceptable key size is 80 bits, whereas 128 bits is recommended. The size of the round ranges from a minimum size of 6 rounds to a maximum of 40 rounds, ensuring a balance between security and efficiency for IIoT applications [15, 16]. According to [38], lightweight cryptographic encryption should have a Gate equivalent (GE) value between 1000 and 2000 GEs.

Definition 3.3 (Synchronous Cryptography). Synchronous cryptography is a cryptographic paradigm based on temporal synchronization, in which communicating entities coordinate cryptographic activities using a standard clock or exact time reference [39, 40].

Definition 3.4 (Asynchronous Cryptography). Asynchronous cryptography is a cryptographic paradigm in which communication entities do not share a clock or use synchronized time. In this paradigm, cryptographic processes, particularly encryption and decryption, occur independently of precise temporal synchronization, allowing for adaptation to communication delays and uncertainty [41].

Definition 3.5 (Lightweight Cryptographic Primitives). Over the last decade, several lightweight crypto primitives such as block ciphers, hash functions, message authentication codes, and stream ciphers have been developed to improve performance over traditional cryptographic standards. These primitives differ from standard algorithms in that lightweight primitives are not intended for a wide variety of applications and may restrict the attacker's capability [42]. For example, the amount of data available to the attacker using a single key may be restricted. However, this does not imply that these lightweight algorithms are ineffective; instead, the goal is to exploit improvements to provide designs that strike a better balance between security, performance, and resource needs for specific resource-constrained environments [43]. The lightweight cryptographic primitives are organized into five categories: block and stream ciphers, hash functions, message authentication codes, and authenticated encryption algorithms.

3.2 Block Cipher

Block ciphers use a secret key to encrypt one block of plain text bits at a time into a block of cipher-text bits. Each cycle consisted of a series of basic changes that introduce confusion, diffusion and avalanche effect [44]. In each round, a round key is utilized, which is produced from the secret key using a key schedule method. Block ciphers are classified into numerous categories based on their algorithm structure.

(1) **Substitution-Permutation Network (SPN):** SPN is a block cipher design paradigm that is frequently utilized in current cryptographic algorithms, including the Advanced Encryption Standard (AES). In an SPN, nonlinear S-boxes are used to substitute the input, which is then permuted using transpositions or shifts. Each round consisted of substitution (S-) and permutation (P-) boxes. S-boxes are often nonlinear transformations that confuse, whereas P-boxes are linear and diffuse [45]. These alternating substitution and permutation layers help to provide the necessary confusion and diffusion features for cryptographic strength. The design's modular nature allows for fast implementation, and its resilience to different cryptographic attacks makes SPNs a popular choice for developing secure block ciphers [46].

(2) **Feistel Network (FS):** An FS is a block cipher that divides the input into two halves, applies a round function to one half, XORs the other with the function's output, and then swaps the halves. This procedure is repeated numerous times, resulting in a symmetric structure for encryption and decryption. Feistel networks, such as Data Encryption Standard (DES), are simple, Parallelizable, and adequate for cryptographic tasks. The input block is divided into two halves, Left and right (Li, Ri), and the output block is $\left(L_{i+1}, R_{i+1}\right)$ = (Ri,Li)⊕F(R_i, K_{i+1}), where F is the round-function [47].

(3) **Add-Rotate-XOR (ARX):** The Add-Rotate-XOR (ARX) cryptographic primitive performs just three operations: modular addition, bit-wise rotation, and bit-wise XOR. This simplified architecture increases efficiency by lowering computing complexity while maintaining cryptographic strength. ARX constructs are used in lightweight cryptography, where simplicity and resource efficiency are essential [48].

(4) **Generalized Feistel Network (GFN):** GFN splits the input block into n rounds, with each round consisting of a round-function layer and block-permutation layer, which is often a cyclic shift. We refer to Type-1 GFN when the round function is applied only to one component and Type-2 GFN when it is used to n/2 parts. Extended GFN refers to the presence of an extra linear layer between two levels.[49].

(5) **Linear Feedback Shift Registers (LFSR)-based:** LFSR-based cryptography technique employs one or more LFSRs within the round function, supplemented by nonlinear functions. This technology combines LFSR's linear features with nonlinear changes to improve the

algorithm's cryptographic strength. LFSR-based designs are widely used in cryptographic constructions because they balance computational efficiency and strong security, making them ideal for applications with limited resources and real-time processing needs [50].

(6) **L-Structure Design (LS):** The LS-design cryptography paradigm integrates linear diffusion L-boxes with nonlinear bit slice S-boxes inside each round. This combination tries to achieve ideal masked implementations, increasing resistance to side-channel analysis [51]. The LS-design, which uses a synergistic combination of linear and nonlinear elements, not only ensures cryptographic robustness but also addresses concerns about side-channel vulnerabilities, making it ideal for secure implementations in scenarios requiring greater protection against potential information leakage [51].

(7) **Extended L-Structure Design (XLS):** It is an LS-design variant that includes the Shift Columns function and Super S -boxes. The input state is separated into four blocks of 4 × 8 dimensions. Similarly, the S box is connected to the columns, and the L box is connected to the rows simultaneously rather than one by one [52].

A block cipher has several crucial features for its cryptographic functioning, including key length, block size, rounds, GE, throughput, and other relevant factors. In this research study, we discuss these characteristics to determine their importance in the context of block cipher design and performance evaluation.

- **Key Length** or key size refers to the number of bits in a key employed in cryptographic encryption procedures. Encryption strength is determined by the complexity of the method, which prevents the key from being discovered. The key size used in the encryption process determines the encryption intensity. Longer keys produce a more robust cipher; however, they require more power and complexity [53].
- **Block Size** operates with a fixed sequence of bits. The input and output blocks in block cipher encryption have the same size. Larger block sizes require more CPU and battery power. As a result, a smaller block size method is appropriate for IoT end devices [54].
- **Rounds** Ciphers employ round-based execution as part of the key size, which makes them particularly secure. The mechanism is safer with a larger key size and more rounds [55]. However, these systems consume large amounts of energy and processing resources. Cryptographic design requires a reduction in the number of round iterations necessary. Consequently, the resources and latency needed to maintain the significant functionality of the IoT nodes are reduced. Classical standard ciphers are not suitable for IoT nodes because several round repeats are required to provide the desired security [56].
- **Gate equivalent (GE) Area:** The GE of lightweight cryptography specifies the physical area required to perform the algorithm. An appropriate lightweight primitive requires a small gate area. As complementary metal-oxide semiconductor (CMOS) technology evolves from micrometer (μm) to nanometer (nm) scales, integrated circuits (ICs) benefit from increased transistor density, reduced power consumption, and higher clock frequencies. This technology value defined by the CMOS process node directly influences the throughput, GE, and energy efficiency of cryptographic hardware implementations. This makes technology vale a critical parameter in evaluating lightweight ciphers for resource constrained environments [57].
- **Throughput** is defined as the number of bits converted per second at a given frequency during the encryption and decryption procedures of the encryption algorithm. The selected frequency is used to represent the throughput. The equation for determining throughput is:$T = B$ x F/ N. Where T denotes the throughput, B denotes the data size in bits, often known as the block size, and F denotes the frequency. N represents the number of cycles in a block [58].

3.3 Stream Cipher

Stream ciphers encrypt a small amount of data (one or more bits) simultaneously. They generate a pseudo-random key stream using a secret key, combined with plain text bits to form cipher text bits. When the combining function is bitwise XORing, its binary additive stream ciphers. The fundamental security guideline for stream ciphers is never to encrypt two separate messages using the same key/IV pair, which acts as an extra input, guaranteeing that each message is encrypted uniquely, and increasing security by avoiding pattern analysis. Thus, stream ciphers often have a long keystream time, after which a different key and/or IV need to be employed [59]. Smaller key sizes such as 80 bits, smaller IV/nonce sizes such as 64 bits, a smaller internal state such as 80 or 100 bits, shorter key scheduling, and a smaller hardware implementation, determined by both cipher design and the underlying CMOS technology value, can make the typical stream cipher technique more lightweight. According to the researchers, Stream ciphers frequently include shift registers, which are simple to build in hardware and are therefore worth considering while searching for suitable algorithms [59].

3.4 Cryptanalysis of Ciphers

Cryptanalysis is the scientific study of cryptographic systems to understand their principles, flaws, and weaknesses. It is frequently used to decrypt encrypted data or violate the security of cryptographic protocols [60]. Cryptanalysts investigate and exploit flaws in cryptographic algorithms and systems using a variety of approaches, such as mathematical analysis, statistical methodologies, and computer algorithms. The overall goal is to improve the security of cryptographic methods by finding and correcting any flaws [61].

Cryptanalysis refers to various methodologies for identifying and exploiting weaknesses in cryptographic systems. Cryptanalysis commonly uses the Brute Force Attack, in which all potential keys are carefully evaluated; this strategy is wildly successful against weak keys or shorter key lengths [62]. Frequency Analysis examines symbol distributions inside encrypted data to identify trends and perhaps reveal the encryption secret [63]. At the same time, Plaintext Attacks use knowledge of plaintext-ciphertext combinations to derive information about the key or encryption process and allow attackers to choose plaintexts to analyze, disclosing key information [64]. Differential and linear cryptanalysis techniques use unique aspects in connecting plaintext, ciphertext, and key [65]. Side-channel attacks make use of leaked information from cryptographic procedures, such as power usage or time [66]. Fault analysis inserts flaws into the system to extract crucial details [67]. The Meet-in-the-Middle attack looks for a key by encrypting and decrypting with every conceivable key [68]. Algebraic attacks use algebraic approaches to model encryption algorithms and extract essential information [69]. These various tactics highlight the need for ongoing advancements in the cryptography procedure to defeat developing attack attempts successfully.

The suitability of LWC in IIoT-based ICS can be significantly affected by its security weaknesses. Many LWC mechanisms, while optimized for low computational overheads and power efficiency, are vulnerable to cyber attacks, as discussed earlier. Furthermore, some LWC algorithms reduce key sizes to optimize performance, thereby weakening authentication and increasing susceptibility to brute-force attacks, which can compromise secure device-to-device communication in protocols such as MQTT and OPC UA Secure [70, 71]. Therefore, while LWC offers advantages in resource-constrained environments, its adoption in ICS must be carefully evaluated by ensuring resistance to cryptanalytic attacks, robust key management practices, and compliance with industrial security standards, making it a feasible choice only when implemented with hardened security measures. Table 4 shows the identification of various cryptanalytic attacks on block and stream ciphers, ensuring that our cryptographic analysis is relevant and thorough.

Table 4. Cryptanalysis of Block and Stream Ciphers

Algorithm	**Cipher**	**Differential Cryptanalysis**	**Side-Channel Attacks**	**Related Key Attacks**	**MITM/Biclique**	**Linear Cryptanalysis**
AES	Block	[118] (2011)	[119–121] (2020)	[122] (2018)	[122] (2018)	-
ASCON	Block	[123] (2020)	[124] (2022)	[125] (2021)	[126] (2022)	[123] (2020)
Camellia	Block	[127, 128] (2010, 2013)	[129] (2021)	-	[130] (2014)	-
CLEFIA	Block	[101] (2001)	[99] (2007)	[120] (2020)	-	-
HIGHT	Block	[131] (2009)	-	[132] (2011)	[133] (2013)	[134] (2014)
ICEBERG	Block	[135] (2012)	-	-	-	-
KTANTAN	Block	-	-	[136] (2012)	[113] (2011)	-
Lblock	Block	[137, 138] (2013, 2014)	-	-	[139] (2012)	-
LED	Block	[140] (2015)	-	[141] (2014)	[142] (1994)	-
mCrypton	Block	-	[143] (2014)	[144] (2009)	[117] (2006)	-
PICCOLO	Block	[145] (2014)	-	-	[133, 141] (2013, 2014)	-
PRESENT	Block	[146, 147] (2013, 2014)	-	[119, 120] (2020, 2020)	[122, 148] (2018, 2014)	-
Modified PRESENT	Block	[149] (2020)	-	[149] (2020)	-	-
PRINCE	Block	[150] (2015)	-	[151] (2012)	[151] (2012)	-
QARMA	Block	[152] (2020)	[153] (2020)	-	[152] (2020)	-
RECTANGLE	Block	[154, 155] (2014, 2018)	[154] (2014)	[154] (2014)	-	-
SAT_Jo	Block	[156] (2019)	-	[156] (2019)	-	[156] (2019)
SEA	Block	[157] (2006)	[157] (2006)	[157] (2006)	-	[157] (2006)
SFN	Block	[158] (2025)	-	[159] (2018)	[158] (2025)	-
SEED	Block	[160] (2012)	[161] (2012)	-	-	[160] (2012)
SIMON	Block	[162, 163] (2013, 2014)	-	[164] (2014)	-	-
SLIM	Block	[165] (2022)	-	-	-	-
TEA	Block	-	-	[166] (2011)	-	-
XTEA	Block	-	-	[166] (2011)	-	-
A2U2	Stream	[167] (2011)	[168] (2012)	[167] (2011)	-	-
BEAN	Stream	[169] (2011)	-	[170] (2013)	-	-
CAR30	Stream	-	[171] (2013)	-	-	-
CAvium	Stream	[172] (2012)	-	-	-	[172] (2012)
ChaCha20	Stream	[173, 174] (2016, 2021)	[175, 176] (2016, 2021)	[173] (2016)	[177] (2023)	[178] (2024)
Enocoro	Stream	[179] (2010)	-	-	-	[179] (2010)
Grain	Stream	[180, 181] (2014, 2018)	-	-	-	[180, 181] (2014, 2018)
MICKEY 2.0	Stream	[182] (2013)	-	[183] (2006)	-	-
Quavium	Stream	-	-	[184] (2016)	-	-
Rabbit	Stream	-	[185] (2014)	-	-	-
Salsa20	Stream	-	-	[186] (2015)	-	-
Trivium	Stream	[187] (2011)	-	-	-	-
WG-8	Stream	[188] (2019)	-	[189] (2014)	-	[188] (2019)

4 Lightweight Cryptographic Algorithms

The cryptographic community has developed over Fifty Symmetric LWC algorithms with an emphasis on lowering costs (memory, processing power, physical space or GE, energy consumption) and improving hardware and software performance (latency, throughput). However, many existing cryptographic algorithms, while labeled as lightweight, are primarily optimized for performance or implementation cost and are not explicitly designed to meet the stringent resource constraints of lightweight IoT devices. [72]. Our article provides an in-depth categorization and analysis of various LWC algorithms tailored specifically for resource-constrained IIoT environments. According to the Definition of LWC in Definition 3.2, some of these algorithms, such as Serpent, Blowfish, Twofish, and RC4 have not been considered. They have been intentionally excluded due to their larger key sizes, which exceed the limit of 80–128 bits to be considered as lightweight security criteria, or due to documented cryptanalytic vulnerabilities [73, 74]. Furthermore, explicitly deprecated algorithms, including DES-L, DES-XL, A5/1, and E0, are omitted to ensure compliance with contemporary security standards and best practices [75–77]. The organization of this review aligns explicitly with the contribution criteria outlined in 1.2, ensuring a systematic evaluation of each lightweight cryptographic algorithm. The subsequent sections comprehensively evaluate and compare each algorithm based on these critical parameters to ensure their suitability for resource-constrained IIoT environments.

4.1 Lightweight Block Ciphers

This section presents the different lightweight block ciphers.

4.1.1 AES. [78] is a typical example of an SPN-based algorithm that has been standardized by NIST in 2001 and as specified in Federal Information Processing Standard (FIPS) 197. It works on 128-bit blocks with 128, 192, and 256-bit key variations. The minimal GE need for AES has been observed to be about 2400 GEs [79], which is still too high for some small-scale real-time applications. The size of the code of AES is 2606 bytes, and the RAM size is 388 bytes. It demonstrates the relatively efficient performance when developing additional resources [80]. The AES algorithm is the cornerstone of modern cryptography, with applications in various areas. Its strong security characteristics have applications in information security, network security, data storage, financial transactions, government and military communications, cloud computing, IoT, applications and software security, and healthcare data protection. AES's flexibility and efficiency make it a reliable choice for protecting the security and confidentiality of sensitive data in a wide range of computing environments [81]. AES is supported on all major operating systems, including Microsoft Windows, macOS, Linux, Android, iOS, Unix, and embedded systems. AES is an essential component of current cryptographic methods, and it works effortlessly across a wide range of devices, from desktop file encryption to mobile data security [82]. Its standardized design provides broad interoperability and use in network devices and security appliances, highlighting its adaptability and dependability in modern computer settings. AES also has the availability of Arduino and Raspberry-Pi for the implementation on lightweight devices [83]. AES provides strong resistance against cryptanalytic attacks due to its well-structured SPN. Its implementations are vulnerable to side-channel attacks, including power and timing analysis attacks [84].

4.1.2 ASCON. It is an online, single-pass, nonce-based Authenticated Encryption (AE) algorithm that incorporates robust keyed initialization and keyed finalization phases, offering stronger security compared to conventional authenticated encryption schemes [85]. It is intended to be lighter in terms of both hardware and software performance. ASCON's major feature is to aid in effectively implementing side-channel attack resistance. Several ASCON cryptanalysis results have been published in [86]. In 2023, it won the CAESAR competition for lightweight, high-performance, authenticated encryption with strong security and protection features. ASCON is specified in ISO/IEC 29192-5:2019, which is a part of the ISO/IEC 29192 standard series covering lightweight cryptography for constrained environments.

ASCON, a lightweight and energy-efficient algorithm, is used in sectors that need resource efficiency. ASCON is mainly utilized in IoT, wireless sensor networks, and Radio-Frequency Identification (RFID) systems. However, it is also suitable for protecting communication in embedded systems, smart cards, and mobile devices [87]. Its function extends to edge computing, contributing to lightweight cryptography research while displaying adaptability in resource-constrained contexts. ASCON can be implemented on various operating systems, including, but not limited to, Windows, Linux, MacOS, and embedded systems, enabling adaptability and interoperability across several computer platforms [87]. ASCON also has Arduino and Raspberry-Pi implementations available for lightweight devices [87, 88]. Its lightweight architecture makes it suitable for use in various operating situations that require effective cryptographic solutions. ASCON is optimized for lightweight applications, offering efficient performance in resource-constrained environments. Reduced-round versions are vulnerable to MITM attacks and differential cryptanalysis, leading to potential security concerns [89].

4.1.3 Camellia. It is introduced by Nippon Telegraph and Telephone Corporation (NTT) and Mitsubishi Electric Corporation in 2000, a block cipher supporting key sizes of 128, 192, and 256 bits, operating on a block size of 128 bits. It follows a Feistel network structure, integrating logical operations, 8×8-bit S-boxes, and input/output whitening, enhancing its cryptographic robustness [90]. The cipher employs 18 rounds for 128-bit keys and 24 rounds for 192/256-bit keys. Its hardware implementation typically occupies around 6511 gates, providing efficiency suitable for hardware-constrained environments [90, 91]. An optimized software implementation on a Pentium III CPU (800 MHz) achieved a throughput of more than 276 Mbps [92]. Camellia has been internationally standardized by ISO/IEC, adopted by Internet Engineering Task Force (IETF) (RFC 3713, RFC 4312), and recommended by the Cryptography Research and Evaluation Committees (CRYPTREC) and New European Schemes for Signatures, Integrity, and Encryption (NESSIE) projects [93, 94]. Security analyses demonstrate Camellia's strength against linear and differential cryptanalysis, although reduced-round variants show theoretical vulnerabilities to impossible differential attacks [95, 96]. Camellia is available in multiple languages, including C, Java, Python, and C#. Camellia is available for Arduino and Raspberry-Pi for implementation on lightweight devices [97, 98].

4.1.4 CLEFIA. It was introduced by SONY Corporation and validated by NIST, provides a 128-bit block with a choice of 128, 192, or 256 bit key through 18, 22, or 26 rounds [99]. CLEFIA is standardized under the ISO/IEC 29192-2:2012 standard. It demonstrates good performance and strong resistance to different attacks, including differential cryptanalysis and linear cryptanalysis, at a rather high cost, since the most compact version consumes 2488 GE (encryption only) for a 128-bit key [100]. Reduced-round versions of CLEFIA are vulnerable to improbable and impossible differential attacks. Such attacks exploit encryption transitions that either occur with very low probability or do not occur, allowing cryptanalyst to eliminate incorrect key candidates and eventually recover secret keys [101]. CLEFIA's remarkable resistance to security assaults is due to its dual confusion and diffusion capabilities. On the contrary, this necessitates more memory and limits its utility to ultra-small applications. CLEFIA only has Raspberry-Pi available for implementation on lightweight devices [102].

The CLEFIA algorithm has applications in various fields that require strong cryptographic solutions. Its adaptability makes it useful for data security in multiple applications, including cloud computing, network communication, and data storage. CLEFIA is especially well-suited for scenarios that require high security and efficient implementation, making it applicable in various industries, from financial transactions and secure communication protocols to secret data protection across multiple computer environments. It is not fundamentally tied to a particular operating system. Its platform-independent implementation makes it compatible with various operating systems, including Windows, Linux, macOS, and embedded systems [103]. CLEFIA's adaptability enables it to be incorporated into multiple computer environments, guaranteeing interoperability and adequate cryptographic security across several platforms.

4.1.5 High Security and Lightweight (HIGHT). processes 64-bit data 32 times with a 128-bit key [104]. It uses basic computational techniques to perform compact round functions (no S-boxes). The smallest version acquires 2608 GE for a throughput of 188 Kbps [105]. HIGHT is defined in ISO/IEC 18033-3:2010. ISO/IEC 18033-3 is an international standard that includes a wide range of lightweight cryptographic algorithms for encryption.

It is suited for applications that need high resource efficiency. Its applications include IoT devices, RFID systems, and wireless sensor networks [106]. HIGHT's efficient implementation, low power consumption, and safe cryptographic capabilities make it an attractive option for ensuring communication and data integrity in resource-constrained contexts.

Reduced-round versions of HIGHT have been susceptible to various cryptanalytic attacks, such as differential and linear cryptanalysis, indicating potential vulnerabilities in these configurations [106]. HIGHT can be customized and deployed on various operating systems, including but not limited to Windows, Linux, macOS, and embedded systems [107]. Its adaptability enables implementation in various computing contexts that require lightweight cryptographic solutions, such as IoT devices and resource-constrained systems. HIGHT is also available on Arduino and Raspberry-Pi for the implementation on Lightweight devices [107].

4.1.6 ICEBERG. It is designed for re-configurable hardware deployment, with the ability to alter the key at each clock cycle without sacrificing quality. Its hardware implementations, particularly targeting platforms like Field Programmable Gate Arrays (FPGAs), allow for key changes at every clock cycle without sacrificing speed, and round keys are generated "on-the-fly" in encryption and decryption modes. There is no need to store the round. It operates on a 64-bit input with a 128-bit key over 16 iterations, with a demand of 5800 GE and a throughput of 400 Kb/s [108]. Reduced-round versions of ICEBERG have been found vulnerable to bit-pattern-based integral attacks, indicating potential weaknesses in these configurations [109]. It is scalable across architectures (loop, unrolled, pipeline) and FPGA technology. ICEBERG is available on Arduino for the implementation on lightweight devices [110].

4.1.7 KTANTAN. It is inspired by KeeLoq, a cipher family that uses an 80-bit key on several block sizes (32-bit, 48-bit, and 64-bit) over 254 iterations [111]. They may be run on small-scale hardware (KATAN 802 GE and KTANTAN 462 GE), which was developed primarily for RFID tags and sensor networks. Instead of KeeLoq's Nonlinear Feedback Shift Register (NLFSRs), they use a linear structure (LFSR). KATAN offers simpler key scheduling than KeeLoq, and it has no key generation activities (which reduces the need for GE). KTANTAN's uses are restricted since the key remains unaltered once established. KTANTAN-48 (588 GE) is better suited for RFID tagging. When implemented on an 8-bit platform, it exhibits poor performance with low throughput and high energy consumption in software due to the abuse of bit manipulation [112]. Advanced cryptanalytic techniques, such as improved MITM attacks, have been effective against reduced-round versions of KTANTAN, indicating potential vulnerabilities in these configurations [113]. KTANTAN only has Raspberry-Pi available for implementation on lightweight devices.

4.1.8 LBlock. It is a lightweight cipher that uses 32 iterations on 64-bit input and 80-bit keys [114]. The smallest hardware deployment requires 1320 GE for a throughput of 200 Kb/s, whilst the most efficient software implementation requires 3955 clock cycles to encrypt a single block (on an 8-bit microcontroller). Reduced-round versions of LBlock have been found vulnerable to various cryptanalytic attacks, including differential and linear cryptanalysis, indicating potential weaknesses in these configurations [106]. Its architecture prioritizes simplicity and performance and is ideal for use in resource-constrained applications, including embedded systems, IoT devices, and wireless sensor networks [106]. Its implementation is typically platform-independent. Thus, it may be ported to various operating systems, including but not limited to Windows, Linux, macOS, and embedded systems. LBlock only has the availability of Raspberry-Pi to implement on lightweight devices [106].

4.1.9 LED. It incorporates characteristics from PRESENT (S-box), AES (row-wise data processing), and PHOTON (mix column method). LED lacks key scheduling, which is a distinguishing feature. This strategy decreases the space of the chip while increasing security risks, such as associated key attacks. The absence of a key schedule in LED increases the risk of key-related attacks, potentially compromising its security. It processes 64-bit input for 32 or 48 times using various keys such as 64-bit (966 GE), 80-bit (1040 GE), 96-bit (1116 GE), and 128-bit (1265 GE) [115]. The design of it makes LED ideal for use in IoT devices, wireless sensor networks, and embedded systems. LED is not fundamentally linked to a certain operating system. Its implementation is portable to various operating systems, including Windows, Linux, macOS, and embedded systems. LED only has the availability of Raspberry-Pi for the implementation on lightweight devices [106].

4.1.10 miniature of Crypton (mCrypton). It is a low-cost, low-energy version of Crypton appropriate for hardware and software deployments [116]. It was launched in 2005 because the old version, Crypton, had minor weaknesses in the key schedule and undesirable properties in S-boxes. It has a better single-key setting with improved meet-in-the-middle attack [116]. It iterates 13 times on the 64-bit block using various keys (64-bit, 96-bit, and 128-bit). mCrypton has been found to be vulnerable to MITM attacks [117]. mCrypton only has the availability of Raspberry-Pi for the implementation on lightweight devices.

4.1.11 PICCOLO. It is yet another ultra-lightweight cryptographic technique that is well suited for resource-constrained devices (RFID, and sensors [190]. It uses two key sets, 80-bit, and 128-bit, to complete two iterations, 25 and 31, using 64-bit input. The most straightforward hardware deployment (80-bit key) requires 432 GE and another 60 GE for decryption. Its implementation can be tailored to various operating systems, including Windows, Linux, macOS, and embedded devices [191]. Its tiny design and low power consumption make it a good option for resource-constrained situations, such as IoT devices, RFID systems, and wireless sensor networks. Reduced-round versions of Piccolo have been found vulnerable to fault analysis attacks, indicating potential weaknesses in these configurations [143]. PICCOLO only has the availability of Raspberry-Pi for implementation on light devices.

4.1.12 PRESENT. It is an ISO/IEC 29192-2:2012 recognized hardware and software efficient algorithm. It is an SPN that employs 64-bit blocks in two key variants: 80-bit and 128-bit keys with GE requirements of 1570 and 1886, respectively [51]. The minimal GE needed for a version of PRESENT is around 1000 GE (encryption alone). However, an appropriate degree of security requires 2520-3010 GE [102]. It is a hardware-efficient technique that employs 4-bit S-boxes (substitution layer - substitutes eight S-boxes with a single S-box). Still, it requires several cycles in software (permutation layer), necessitating an upgraded version.

Its small size and low computing complexity make it ideal for deployment in limited contexts like RFID systems, smart cards, and other embedded devices [106]. PRESENT's efficient implementation allows for safe data encryption in settings where computing resources are restricted, ensuring secrecy in applications like IoT devices, wireless sensor networks, and other lightweight cryptographic environments. PRESENT's key schedule has been identified as a potential weakness, with studies suggesting that it may be susceptible to related-key and slide attacks [102]. Its implementation can be tailored to various operating systems, including Windows, Linux, macOS, and embedded systems. The algorithm's efficient design makes it appropriate for resource-constrained systems and applications. PRESENT has the availability of Arduino and Raspberry-Pi for the implementation on lightweight devices [54, 192].

4.1.13 Modified PRESENT. It is the original PRESENT cipher is being modified by lowering encryption rounds, adjusting the Key Register updating mechanism, and inserting a new layer between the existing encryption-decryption process's S-box layer and P-layer. The key register is updated by encrypting its value with the delta value function of another lightweight cipher, Tiny encryption algorithm (TEA). Adding a layer allows us to lower the PRESENT round from 31 to 25, the bare minimum for security. Encrypting the key register improves the performance of the proposed technique [149]. If specific attacks, such as differential or linear cryptanalysis, are not carefully analyzed. Therefore, any modification to the PRESENT cipher must undergo a thorough security evaluation to ensure that the enhancements do not compromise its robustness [149]. Modified PRESENT is unavailable for Arduino and Raspberry-Pi for implementation on lightweight devices.

4.1.14 PRINCE. It is a lightweight hardware and software efficient algorithm that runs on 64-bit input utilizing a 128-bit key 12 times. The most basic hardware solution requires 2953GE at a throughput of 533.3 Kb/s. It uses a tiny amount of energy, 5.53 μj/bit [193]. The small structure of PRINCE and the minimal processing overhead are ideal for limited situations such as RFID systems, IoT devices, and embedded systems. Despite its design for efficiency, PRINCE's relatively small number of rounds and simplistic key schedule may expose it to certain attacks, such as related-key and MITM attacks, particularly in reduced-round scenarios [151]. Its implementation is adaptable to various operating systems, including Windows, Linux, macOS, and embedded systems. PRINCE only has the availability of Raspberry-Pi for the implementation on lightweight devices [194].

4.1.15 Quasigroup-based Authenticated Cipher with Resilience to Multiple Adversaries (QARMA). It is influenced by reflection ciphers such as PRINCE, which extends with a tuning input, and MANTIS [195]. QARMA, unlike earlier reflector designs, is a three-round Even-Mansour scheme rather than an FX-construction, with a non-involutionary and keyed middle permutation [196]. QARMA is available in 64- and 128-bit block sizes, with block and tweak sizes being equivalent and keys being twice as long as blocks. Its distinct architectural characteristics make it suitable for use in various security-critical sectors such as automotive security, where it has been employed for secure memory encryption and integrity protection (Armv8-A processors) [197]. QARMA's applications include secure communication protocols, data integrity verification, and the protection of sensitive information in contexts where resistance to various adversary scenarios is critical [198]. QARMA's adaptability and cryptographic resilience make it a great tool to protect data in various settings, including communication networks, IoT devices, and other cryptographic applications. Reduced-round versions of QARMA have been found to be vulnerable to MITM attacks, indicating potential weaknesses in these configurations [198]. Its implementation can be adapted to various operating systems, including Windows, Linux, macOS, and embedded systems. QARMA only has the availability of Raspberry-Pi for the implementation on lightweight devices [199].

4.1.16 RECTANGLE. It is a lightweight block cipher that may be used in various applications. The rounds are reduced to 25 (compared to 31 in PRESENT) with minor modifications to the SPN structure [154]. However, reduced-round version of RECTANGLE have been found vulnerable to differential fault analysis, indicating potential weaknesses in these configurations [155]. RECTANGLE is not available for Arduino and Raspberry-Pi for the implementation on lightweight devices.

4.1.17 SAT_Jo. The system is based on an SPN with lightweight block cipher suitable for IoT tag-based functionalities. By 24 orders of the Galois field, this system computes a 4x4 S-box. This scheme involves 31 rounds and is based on the SPN of a block cipher with Data Encryption Standard- Lucifer (DES-L) and PRESENT [200]. It used a 64-bit block and an 80-bit key size. This solution provides acceptable security to resource-constrained tag-based application nodes. However, SAT_Jo's structural similarities to the PRESENT cipher may expose it to integral distinguishes, potentially compromising its security in certain scenarios [200]. It also provides superior performance, resource constraints, and security for resource-constrained IoT devices [201]. SAT_Jo is not available for Arduino and Raspberry-Pi for the implementation on lightweight devices.

4.1.18 SEA. It is developed for small IoT devices, particularly memory-constrained devices [202]. It employs a 96-bit key on two suggested block sizes of 96-bit and 8-bit, with a hardware requirement of 3758GE for the lightest hardware version. On 8-bit micro-controllers, the optimized software execution requires 426 bytes with an encryption cycle of 41604 [112]. However, SEA's performance may be suboptimal in certain hardware implementations due to its reliance on basic operations, potentially leading to higher latency compared to other lightweight ciphers [202]. SEA is not available for Arduino and Raspberry-Pi.

4.1.19 SEED. It is a symmetric block cipher developed by the Korea Information Security Agency (KISA), characterized by a 128-bit key and block size structured through a Feistel network spanning 16 rounds [203]. The cipher was primarily designed to achieve robust cryptographic security alongside efficient hardware and software implementations. SEED utilizes strategically designed substitution boxes (S-boxes) combined with linear transformations to ensure substantial resistance against known cryptanalytic attacks, notably differential and linear cryptanalysis [203]. SEED is standardized under ISO/IEC 18033-3 and thoroughly documented in IETF RFC 4269, SEED provides credible international recognition and interoperability [204, 205]. SEED has been implemented in various programming languages such as C, Python, Java, and Assembly (ARM) [206]. SEED is available for Arduino and Raspberry-Pi for the implementation on lightweight devices [207].

4.1.20 SFN. employs a unique encryption algorithm that encrypts the SPN structure and the FS structure. It has modified the SPN topology by utilizing involution-related features of the nonlinear and linear components. The altered one allows the encryption and decryption program or circuit to function as the FS structure [208]. It also included a MixRows in the SP network structure and included these three innovative concepts into the lightweight block cipher known as SFN. Cryptanalysis has revealed vulnerabilities in SFN's full-round version, including susceptibility to related-key differential and MITM attacks, which can compromise its security [158]. SFN is not available for Arduino and Raspberry-Pi for the implementation on lightweight devices.

4.1.21 SIMON. It was created by the National Security Agency (NSA) and is noted for its minimal hardware implementations, such as Micro-controllers, FPGAs, Arduino Platforms, Raspberry-pi. Due to several weaknesses, it was not initially published as a standard, but it was eventually standardized by ISO as an RFID air interference document ISO/29167-21:2018 [209]. It provides various key sizes (64-bit, 72-bit, 96-bit, 128-bit, 144-bit, 192-bit, 256-bit) across a 32-bit, 48-bit, 64-bit, 96-bit, 128-bit block of 32, 36, 42, 44, 52, 54, 68, 69, 72 rounds [210]. The smallest version consumes 763GE to execute. However, reduced-round versions of SIMON have been susceptible to differential cryptanalysis, indicating potential vulnerabilities in these configurations [210]. Its implementation is accessible to various operating systems, including Windows, Linux, macOS, and embedded systems. Its compact and low power consumption make it ideal for resource-constrained situations like IoT devices, wireless sensor networks, and embedded systems [211]. SIMON only has the availability of Raspberry-Pi for the implementation on lightweight devices.

4.1.22 SLIM. It is a 32-bit block cipher based on the Feistel structure [212]. SLIM uses the same key for both encryption and decryption. The method performs well in hardware and software settings and has a small implementation footprint. It is designed for efficient encryption in resource-constrained environments such as RFID systems and the Internet of Health Things (IoHT). Cryptanalysis has revealed that reduced-round versions of SLIM are vulnerable to differential attacks, indicating potential security weaknesses in these configurations [165]. SLIM also has Arduino and Raspberry-Pi available for the implementation on lightweight devices [213].

4.1.23 TEA. It is appropriate for very tiny, computationally inefficient, and low-cost hardware [214]. It uses a 128-bit key on a 64-bit input to accomplish 32 cycles with 3872 GE needs. Because of its easy key scheduling, it is subject to brute-force attacks. Another disadvantage of the TEA structure is that it uses three equivalent keys for decryption, making it vulnerable to attackers [166]. Its simplicity and efficiency make it ideal for resource-constrained situations, including micro-controllers in IoT devices, RFID systems, and other embedded applications. TEA's small architecture enables it to be used when computing resources are restricted, providing a balance of cryptographic security and efficiency.

The TEA is operating system independent and may be used on various platforms, including Windows, Linux, macOS, and embedded systems. Its adaptability enables developers to include TEA into programs across several operating systems, making it appropriate for a broad range of cryptographic contexts, including desktop settings, embedded devices, and mobile platforms such as Android and iOS [215]. TEA also have the availability of Arduino and Raspberry-Pi for the implementation on lightweight devices.

4.1.24 XTEA. It is an upgraded variant of TEA that utilizes the same key and block size but with additional iterations (64 rounds), requiring 3490 GE [216]. It allows for more complicated key scheduling while retaining Shift, XOR, and addition capabilities. It raises the number of rounds to 64, increasing resistance to cryptanalysis, and includes a changed key schedule for better dissemination [217]. XTEA retains compatibility with TEA while resolving discovered flaws, resulting in a more robust and secure block cipher that supports variable key sizes. XTEA was modified further with XXTEA to be immune to related-key rectangle attack (on 36 rounds) [218]. XTEA also have the availability of Arduino and Raspberry-Pi for the implementation on lightweight devices.

Summary of Section 4.1. This section examines and evaluates block cipher algorithms appropriate for use as lightweight encryption methods in IIoT devices. We examine famous block ciphers such as AES, ASCON, PRESENT, RECTANGLE, and others to address the unique issues provided by resource-constrained devices in industrial applications. Table 5 summarizes the known lightweight block ciphers, listed alphabetically, with their Design patterns, key Length, block size in bits, number of rounds, number of GE's if known, Code size in bytes, RAM size in bytes, and Throughput in Mbps. Meanwhile, Table 6 summarizes the standards and availability of implementation of block ciphers on Arduino and Raspberry-Pi. By analyzing their performance, computational efficiency, and adaptability to the strict limits of IIoT devices, the research intends to give insights into selecting optimal block cipher solutions for improving the security infrastructure of IIoT systems. The findings add to the current discussion of safeguarding IIoT ecosystems with specialized and effective cryptographic techniques, providing direction to practitioners and academics in the area.

4.2 Lightweight Stream Ciphers

4.2.1 A2U2. It is a domain-specific stream cipher [224]. It was created for the highly constrained resource environment of printed electronic RFID tags. The security area must be no more than 500 GE, with power consumption restricted to tens of Ws [224]. Throughput should also be sufficient to allow for real-time interactions with many tags. A2U2 is one of the most compact stream cipher studied in this study. The smaller version was intended to use 284 GE and deliver 50 Kbps throughput. Despite its compact design, A2U2 has been found vulnerable to ultra-efficient key recovery attacks.

Table 5. Lightweight Cryptographic Algorithms Using Block-Cipher Encryption/Decryption Mechanism

Algorithm	**Design Pattern F/SPN**	**Key Length (bits)**	**Block Size (bits)**	**Rounds**	**GE Area**	**Code size (bytes)**	**RAM size (bytes)**	**Through-put (Mbps)**
AES	SPN	128/192/256	128	10/12/14	2400	2606	388	56
ASCON	SPN	128	128	12	4000	1600-1900	1600-1900	1000
Camellia	F	128/192/256	128	18/24	6511	2000-4000	1000-2000	276
CLEFIA	SPN	128/192/256	128	18/22/26	2488	1500-2000	1000-2000	200-300
HIGHT	F	128	64	32	3048	13,476	288	188
ICEBERG	SPN	128	64	16	5800	-	-	-
KTANTAN	-	80	64	12/16/20	688	2000-4000	1000-2000	25.1
Lblock	F	80	64	32	1320	1000-2500	1000-2000	100-200
LED	SPN	128	64	32	1265	2000-3000	1000-2000	133.33
mCrypton	-	128	64	13	2709	-	-	-
Modified PRESENT	SPN	80/128	64	31/36	1884	1000-2000	1000-2000	100-200
PICCOLO	SPN	80/128	64	31/36	1260	1000-2000	1000-2000	100-300
PRESENT	SPN	80/128	64	32	2195/1886	1738	274	206
PRINCE	SPN	128	64	12	3491	1000-2500	1000-2000	533
QARMA	SPN	128	128	16	2000-3000	1000-2500	1000-2000	1705
RECTANGLE	SPN	128/256	64/128	18/20	1787	2000-3000	1000-2000	246
SAT_Jo	SPN	80	64	31	1167	-	-	14.9
SEA	F	96	96	-	3758	2132	-	-
SEED	F	128	128	16	25000	4096	1000-2000	400-600
SFN	F	96	64	32	1876	-	-	200
SIMON	F	64/96/128 /192/256	32/48/64 /96/128	32-72	763	1000-2000	1000-2000	15.8
SLIM	F	80	32	32	553	1000-2000	1000-2000	10-20
TEA	F	128	64	64	2100	648	196	-
XTEA	F	128	64	64	1000-2000	1000-2000	1000-2000	10-20

In these attacks, an adversary can fully recover the secret key by querying the cipher twice and solving sparse linear equations, compromising its security [225]. The system queries the victim tag's encryption function twice and solves 32 spare linear equations, breaking the cipher in 0.16 seconds on a laptop computer [225].

4.2.2 BEAN. It has an 80-bit key size and is more compact than Grain. Grain is the foundation for BEAN [226]. BEAN makes use of two Feedback with Carry Shift Registers (FCSRs) and an S-box. The two FCSRs update themselves using different primitive polynomials, while the S-box adds improved diffusion features for key stream creation and fixes cryptanalysis difficulties presented by FCSRs. It also offers binary output generation without the need for additional hardware, which is an advantage over Grain [169]. BEAN utilizes the same memory as Grain in software implementations but takes less time to create the key stream bits. However, due to the weak output function of BEAN, an efficient distinguished attack and a state recovery attack are proved in [227].

4.2.3 CAR30. It is a stream cipher that, like CAvium, employs Cellular Automata (CA). It uses conventional Rule 30 of CA, as well as a maximum length linear hybrid CA [228]. This combination of a linear and a non-linear CA, and their operation across several rounds, minimizes the linearity with the adjacent sequence during key stream generation, relieving the security difficulties associated with CA-based ciphers. Furthermore, the suggested approach allows for varying key and IV sizes without affecting the cipher's architecture or structure. CAR30 offers

Table 6. Available Standardization and Implementation of Block Ciphers on Arduino and Raspberry-Pi

Algorithm	Arduino	Raspberry-pi	Standards
AES	Yes [82, 83]	Yes [81]	FIPS 197, ISO/IEC 18033-3:2010 [84, 204]
ASCON	Yes [87]	Yes [88]	NIST SP 800-232 ipd [219]
Camellia	Yes [97]	Yes [98]	ISO/IEC 18033:3-2010, IETF RFC (3713 and 4312) [93, 94, 204]
CLEFIA	No	Yes [103]	ISO/IEC 29192-2:2012 [220]
HIGHT	Yes [107]	Yes [107]	ISO/IEC 18033:3-2010 [204]
ICEBERG	Yes [110]	No	-
KTANTAN	No	Yes [106]	-
Lblock	No	Yes [106]	-
LED	No	Yes [106]	-
mCrypton	No	Yes [221]	-
PICCOLO	No	Yes [191, 222]	-
PRESENT	Yes [192]	Yes [106]	ISO/IEC 29192-2:2012 [220]
Modified PRESENT	No	No	-
PRINCE	No	Yes [223]	-
QARMA	No	Yes [199]	-
RECTANGLE	No	No	-
SAT_Jo	No	No	-
SEA	No	No	-
SEED	Yes [207]	Yes [206]	ISO/IEC 18033:3-2010, IETF RFC 4296 [204, 205]
SFN	No	No	-
SIMON	No	Yes [211]	ISO/IEC 29167-21:2018 [209]
SLIM	Yes [213]	Yes [213]	-
TEA	Yes [215]	Yes [215]	-
XTEA	Yes [217]	Yes [215]	-

customized security and extensibility for any key or IV length [229]. In each iteration, the suggested setup employs 128-bit keys and 120-bit IVs to generate 128-bit blocks of key stream. Its statistical randomness features passed the NIST statistical test suite for random and pseudorandom number generators for cryptographic applications. CAR30 cipher can prevent fault attacks by exploiting the inherent properties of Cellular Automata [228].

4.2.4 CAvium. It is another new Trivium-based concept. It employs CA with both nonlinear and linear rules, resulting in a secure design that is substantially quicker than the corresponding LFSR and NLFSRs structures in reaching the appropriate setup state of a cipher [230]. Thus, Cavium drastically lowers Trivium's lengthy key setup phase (from 1152 to 144 rounds) and prevents cryptanalysis attacks on the reduced round variants while preserving equivalent hardware complexity. Independent cryptanalysis results are not reported, to the best of our knowledge [172].

4.2.5 ChaCha20. It is a high-speed stream cipher developed by Daniel J. Bernstein in 2008 as a variant of the Salsa20 cipher. It operates on a 512-bit state, utilizing a 256-bit key and a 96-bit nonce, and processes data in 20 rounds to generate a pseudo-random key stream. This design enhances security and performance, making ChaCha20 particularly efficient in software implementations [73]. The cipher is often paired with the Poly1305 message authentication code (MAC) developed by Bernstein to provide authenticated encryption. This combination, known as ChaCha20-Poly1305, ensures both the confidentiality and integrity of the encrypted messages [231]. ChaCha20 was specifically designed to resist side-channel attacks, including cache-timing attacks, enhancing its suitability for secure applications. Its performance is notable; for instance, on mobile devices lacking

AES hardware acceleration, ChaCha20-Poly1305 can be up to three times faster than AES-128-GCM, leading to reduced decryption times and improved battery life [231]. The cipher's internal structure includes a 96-bit nonce IV and a 32-bit block counter, which together ensure the uniqueness of key stream blocks for a given key. This configuration allows for the encryption of up to 2^{32} blocks (each 64 bytes), totaling 256 GiB of data per key/nonce pair.

ChaCha20's strengths include its high performance on platforms without specialized hardware support for AES, making it particularly suitable for mobile and embedded systems. Its design simplicity contributes to ease of implementation and reduces the risk of security vulnerabilities. However, like all stream ciphers, it requires a unique nonce for each encryption operation to maintain security; reusing a nonce with the same key can lead to key stream reuse attacks [232]. Implementations of ChaCha20 are available in various programming languages, including C and Python. For instance, a straightforward C library compliant with RFC 7539 is accessible on GitHub. Regarding hardware platforms, ChaCha20 has been successfully implemented on devices like the Raspberry Pi Zero W and Arduino Teensy 3.2. Studies have shown that ChaCha20 performs efficiently on these microcontrollers, making it a viable choice for secure communications in resource-constrained environments [231, 233].

4.2.6 Enocoro. It was created by Hitachi in 2007 as their fourth encryption algorithm. Enocoro achieves AES encryption with one-tenth the amount of power usage. Enocoro-80 and Enocoro-128 [234] are members of the Enocoro family, with keys of 80 and 128 bits, respectively. Enocoro employs 64-bit IVs and a byte-oriented architecture with an S-box, which works well in hardware and software. It generates 1 byte each round and up to 264 bytes for each key and IV pair. Enocoro-80 requires 2700 logic gates in hardware, equivalent to the relevant implementations of the other Ecrypt Stream Cipher Project (eSTREAM) Profile 2 finalists [235]. Enocoro-128 requires 4100 logic gates to achieve 3520 Mbps at 440 MHz. Enocoro-128 needs 4869.5 cycles to start in the software, achieving a throughput of 46.3 cycles/byte. The initialization step is similar to that of Trivium. However, the encryption is faster than that of AES-CTR and grain. Recent studies have identified vulnerabilities in the Enocoro stream cipher family, including susceptibility to fault attacks that can compromise its security [236]. Enocoro has the availability of Raspberry-Pi for the implementation on lightweight devices [237].

4.2.7 Grain 128a. It is an improved version of Grain-128 encryption that enhances the security of the original proposal and incorporates authentication capability [238]. It employs slightly different non-linear functions to mitigate the attacks reported on Grain-128. Grain-128a employs 128-bit keys and 96-bit IVs, with flexible tag widths of up to 32 bits. It comprises three components: an LFSR, an NLFSRs, and a pre-out function. When authentication is employed, the shift registers are timed regularly, and the cipher produces one bit per cycle or one bit every two cycles. The smaller, 32-bit tag variant requires approximately 2,769.5 GE in hardware. The encryption requires 2145.5 GE without authentication, whereas the original Grain-128 requires 2133 GE under identical conditions [239].

Its range of applications includes IoT devices, RFID systems, and other low-power devices where protecting computing resources is essential. The small size of Grain-128a and its ability to deliver strong encryption make it ideal for protecting communication where efficiency and resilience to attacks are essential despite limited resources [181]. Despite its robust design, Grain 128a has been found vulnerable to related-key chosen IV attacks, enabling adversaries to recover the 128-bit secret key with a computational complexity of $2^{96.322}$, requiring 2^{96} chosen IVs [240]. Grain-128a can be implemented on various operating systems, including Windows, Linux, and macOS, as long as the programming language and cryptographic libraries are supported. It is frequently used in embedded systems, IoT devices, and other resource-constrained applications that require lightweight cryptographic solutions [241]. Grain-128a is one of the ciphers produced by the eSTREAM project. Although eSTREAM is not standard, its purpose was to find and promote stream ciphers that can be considered for standardization. Grain-128a has the availability of Raspberry Pi for implementation on lightweight devices [242].

4.2.8 MICKEY 2.0. It is the third eSTREAM profile 2 finalist is Mutual Irregular Clocking Key stream generator (MICKEY) 2.0. To introduce nonlinearity, it employs a Galois LFSR and an NFSR with irregular clocking, as well as some unique ways to ensure period and pseudorandomness. It is worth mentioning that the initial version of the encryption had 80 stages for the two registers. However, due to security vulnerabilities discovered during the early stages of eSTREAM, the states were eventually extended to 100 [243]. It has an 80-bit key and an IV ranging from 0 to 80 bits. Each key/IV combination can create 240 keystream bits, and each key can be used with up to 240 distinct IVs of the same length. A hardware implementation takes up to 3188 GE. Related key attacks with 65 key/IV pairings have a 0.9835 probability of breaking MICKEY 2.0 [182].

MICKEY 2.0's compact design and ability to deliver powerful encryption make it ideal for safeguarding communication when saving computing resources and limiting battery consumption are critical considerations. MICKEY 2.0 can be used on various operating systems, including Windows, Linux, and macOS, as long as the programming language and cryptographic libraries are compatible. MICKEY 2.0 has the availability of Raspberry-Pi for the implementation on lightweight devices [244].

4.2.9 Quavium. It is proposed as a scaleable expansion of Trivium, with the same key sizes (80 bits), IV (80 bits) and internal state (288 bits). It is built on Trivium-like Shift Right (SHR)s with four rounds and primitive polynomials of k order [245]. Quavium employs four Trivium-like SHRs in coupling connection rather than the original Trivium's three SHRs in series connection because the coupling connection preserves the primitiveness of characteristic polynomials. It may also function with two or three rounds. Quavium takes 3496 GE in hardware and 2372 GE in the three-round version, yet it is as fast as Trivium in software, with the three-round version outperforming it. In greater detail, a Trivium C++ implementation on an Intel Core 2 Duo 2.00GHz produces 16.8 cycles/byte. In contrast, an analogous Quavium implementation obtains 17 cycles/byte, and a three-round Quavium implementation achieves 12 cycles/byte [246].

4.2.10 Rabbit. It is a synchronous stream cipher presented in FAST Software Encryption (FSE) 2003, designed for excellent software performance, featuring lightning-fast key setup and encryption capability. It is suitable for internet protocols and other applications that process large volumes of data or many packets; it is also one of the eSTREAM's most efficient software suggestions. Rabbit's creators patented it; however, the algorithm is open source. It is part of the wolfSSL embedded TLS library and is specified in the ISO/IEC18033-4:2011 [247] standard for information technology security approaches. Rabbit employs straightforward procedures that make use of contemporary CPU capabilities. It delivers high nonlinearity that is not dependent on NFSR and S-boxes; Rabbit is based on chaotic map concepts. The key is 128 bits long, while the IV is 64 bits long. The software implementation on a 1.7 GHz Pentium 4 processor consists of 1976 bytes of code and requires 486 cycles for key setup and 5.1 cycles for encrypting a byte [248]. The rabbit was also considered for the eSTREAM profile 2. Its smallest hardware version needs 3800 GE and achieves 88 Kbps throughput.

Its effectiveness makes it suitable for protecting communication channels in wireless networks, Virtual Private Networks (VPN)s, and secure communication protocols. Rabbit's lightweight architecture and ability to create a large keystream effectively make it suitable for resource-constrained contexts, such as IoT devices and embedded systems, where computing efficiency is critical. Rabbit could run on various operating systems, including Windows, Linux, and macOS, as long as the programming language and cryptographic libraries are supported [249]. Rabbit has the availability of Arduino and Raspberry-Pi for the implementation on lightweight devices [249, 250].

4.2.11 Salsa20. It is an eSTREAM software implementation finalist in 2005. It employs 256-bit keys and 128-bit IVs [186]. Three variations have been proposed to address the tradeoff between security and performance, each catering to a distinct application's demands. Salsa20/20 is intended for encryption in common cryptographic applications, while the reduced round ciphers Salsa20/12 and Salsa20/8 are designed for faster but less secure operation [251]. Its architecture is built on basic addition, modulo 232, bit rotation, and bitwise XOR (ARX) operations that are efficiently implemented in software. A more compact solution is possible because the encryption and decryption methods are similar. Salsa20 is a high-speed stream cipher in the Crypto++ cryptography library [186]. The smallest software implementation involves 1452 bytes of code and 18,400 encryption cycles. The smallest hardware implementation takes up 12 126 GE, which is much above the capabilities of lightweight cryptography. Attacks against reduced versions of Salsa20 have been described; there is no better attack than exhaustive key search for the versions Salsa20/20 and Salsa20/12 [186].

Salsa20 is commonly used in secure communication protocols, disk encryption, and wireless security, offering strong protection for data transfer over the Internet and safe storage. Its lightweight architecture is suitable for resource-constrained situations, such as IoT devices. At the same time, its interaction with cryptography libraries and software applications ensures effective and secure data security [252]. Salsa20's adaptability allows it to be integrated into a wide range of systems, including embedded devices and desktop settings. It can be used on various operating systems, including Windows, Linux, macOS, and others, as long as the programming language and cryptographic libraries are compatible with the target platform [253]. Salsa20 has the availability of Arduino and Raspberry-Pi for the implementation on lightweight devices [253, 254].

4.2.12 Trivium. It is an eSTREAM Profile 2 finalist and an ISO/IEC 29192-3:2012 standardized stream cipher for lightweight cryptography [255]. Its creators wanted to see how far a stream cipher could be reduced without losing security, speed, or flexibility. It is a synchronous bit-oriented stream cipher that uses 80-bit keys and IVs. To avoid constructing non-linearity mechanisms for the key-stream output, it employs three SHR and creates a nonlinear internal state. In terms of hardware, its implementation in typical CMOS technology requires GE in 2017, a custom design implementation with dynamic logic and C2MOS flip-flops takes up just 749 GE [255].

Trivium is used in wireless communication protocols and cryptographic applications that require an appropriate balance of speed and security, making it an adaptable option for scenarios requiring lightweight and robust stream cipher capabilities. As a cryptographic algorithm, Trivium does not support or rely on specific operating systems [256]. Trivium has undergone extensive cryptanalysis since its inception and, as of 2015, no attacks more effective than brute force have been identified against the full version, indicating a strong security profile [257]. Trivium can be used on various operating systems, including Windows, Linux, and macOS, as long as the programming language and cryptographic libraries are compatible with the target platform. Salsa20 has the availability of Raspberry-Pi for implementation on lightweight devices [258].

4.2.13 WG8. It is an improved version of WG-7 that counters and improves on the assault. It employs 80-bit keys and IVs, with the characteristic polynomial LFSR consisting of eight tap positions (F28) [262]. WG-8 inherits the strong randomization qualities of the WG cipher family and is immune against most typical stream cipher attacks. The cipher works well with embedded applications and consumes little energy [262]. Studies have identified potential vulnerabilities, such as the susceptibility to distinguish attacks under certain conditions that could compromise

Table 7. Lightweight Cryptographic Algorithms Using Stream-Cipher Encryption/Decryption Mechanism

Algorithm	Key Length (bits)	IV/Nonce (bits)	Rounds	GE Area	IS	Max. Keystream bits per (key,IV/nonce)	Technology Value (nm)
A2U2	61	64	var.	283	95	-	-
BEAN	80	64	81	-	160	-	-
CAR30	128	120	160	-	256	> 2^{122}	-
CAvium	80	80	144	-	288	-	-
ChaCha20	128/256	96	20	13600	128-256	2^{32}	-
Enocoro	80/128	64	40/96	2700/4100	176/272	2^{35} / 2^{67}	0.18/0.09
Grain 128a	80/128	64/96	160	1294/1857	160/256	2^{43}	0.13
MICKEY 2.0	80/128	80/128	260/416	3188/5039	200/320	2^{40} / 2^{64}	0.13
Quavium	80	80	1152	3496	288	264	-
Rabbit	128	64	4 + 4	3800	513	2^{71}	0.18
Salsa20	128/256	64	20	9970	512	2^{73}	0.18
Trivium	80	80	1152	749	288	2^{64}	0.35
WG-8	80	80	40	1786	160	2^{160}	0.065

its security. However, WG-8 is 2-15 times quicker than comparable lightweight stream ciphers, while spending 2-220 times less energy. For specified key-IV combinations, a key recovery attack requires 253 chosen IVs to recover the key with 253.32 complexity. When these key-IV pairings are avoided, the security level supplied is still suitable for wireless sensor networks (WSN) and RFID tags [262]. WG-8 has been designed for encryption in resource-restricted environments such as RFID applications and smart cards. WG-8 has the availability of Raspberry-Pi for implementation on lightweight devices [261].

Summary of Section 4.2. This section explored and examined stream cipher algorithms that might serve as lightweight encryption mechanisms in IIoT systems. The need for secure communication and data integrity in IIoT contexts with resource-constrained devices required a detailed evaluation of stream ciphers. Notable algorithms, such as Salsa20, Rabbit, Trivium, and others, were investigated for their ability to provide strong cryptographic security while accommodating the limits inherent in IIoT devices. Table 7 summarizes the known lightweight stream ciphers, listed alphabetically, with their Design patterns, key Length, IV/Nonce, number of rounds, number of GE's if known, Internal State (IS), Maximum Key stream bits per IV/Nonce, and Technology Value. Table 8 summarizes the standardization and available implementation of stream ciphers on Arduino and Raspberry-Pi. The findings provide valuable information on the selection of optimal stream cipher solutions adapted to the specific requirements of lightweight encryption in IIoT applications, thus improving the overall security posture of industrial ecosystems.

5 Research Challenges and Future Directions

As IIoT systems advance, implementing LWC presents critical challenges due to device constraints, varying security demands, and scalability limitations. Ensuring robust and future-proof security requires addressing these complexities. The key research challenges include:

- Implementing LWC in IIoT is challenging due to the heterogeneous nature of industrial systems, where security needs vary between resource-constrained edge devices (e.g., sensors, actuators), mid-tier controllers (e.g., PLCs), and high-performance enterprise systems. Cryptographic solutions must ensure efficiency, interoperability, and resilience while supporting

Table 8. Available Standardization and Implementation of Stream Ciphers on Arduino and Raspberry-Pi

Algorithm	**Arduino**	**Raspberry-Pi**	**Standards**
A2U2	No	No	-
BEAN	No	No	-
CAR30	No	No	-
CAvium	No	No	-
ChaCha20	Yes[231]	Yes [232]	RFC 7539 [233]
Enocoro	No	Yes [237]	ISO/IEC 2912-3:2012[259]
Grain 128a	No	Yes [242]	
MICKEY 2.0	No	Yes [244]	-
Quavium	No	No	-
Rabbit	Yes [249]	Yes [250]	ISO/IEC 18033-4:2011 [260]
Salsa20	Yes [254]	Yes [253]	-
Trivium	No	Yes [258]	ISO/IEC 29192-3:2012 [259]
WG-8	No	Yes [261]	-

large-scale deployments. In addition, scalable key management mechanisms must minimize computational overhead without compromising security. Addressing these complexities requires adaptable LWC frameworks capable of maintaining security across diverse IIoT environments, legacy systems, and other IIoT devices manufactured by third parties.

— Establishing open standards, along with standardized best practices and guidelines specifically to implement LWC solutions in IIoT, is essential to overcome the challenges of fragmentation and interoperability. Incorporating clear and uniform requirements LWC into procurement processes will ensure compliance, scalability, and consistency across resource-constrained industrial devices. This approach strengthens overall IIoT security by promoting efficient, robust, and standardized cryptographic deployments.

— Implementing security-by-default and security-by-design principles in LWC for IIoT remains challenging due to resource constraints and real-time operational requirements. Traditional security-by-design practices from conventional Information Technology (IT) domains are difficult to directly apply in IIoT, necessitating context-specific adaptations that prioritize minimal computational overhead, energy efficiency, and low latency. Thus, researchers must focus on effectively integrating these security-by-design principles into lightweight cryptographic implementations tailored specifically for the unique operational characteristics and constraints of IIoT ecosystems.

— Symmetric cryptographic algorithms are generally less vulnerable to quantum computing than asymmetric ones, primarily due to the Grover search algorithm, which accelerates brute-force attacks but does not render symmetric encryption obsolete. To remain secure against quantum threats, symmetric encryption algorithms should employ key lengths that exceed 112 bits. For instance, AES-256 is classified in security category 5 (maximum level), as documented by NIST IR8547 [263]. However, given the extended lifespan of IIoT infrastructures, which often exceed decades, it is critical to develop and deploy lightweight cryptographic solutions resilient to future quantum threats. Although NIST has initiated standardization efforts for quantum-resistant public-key cryptography, integrating these into resource-constrained IIoT environments remains challenging [264–266]. The primary research challenge lies in effectively adapting post-quantum cryptographic schemes to ensure sustainable quantum-safe security while operating within the computational, storage, and energy limitations of IIoT ecosystems.

- Ensuring secure and resilient firmware updates poses a critical challenge in IIoT, especially when lightweight cryptographic algorithms deployed become compromised or obsolete. The primary research challenge is to design lightweight and robust firmware update mechanisms that are capable of securely delivering algorithm replacements, preventing unauthorized access, preserving system integrity, and minimizing downtime in industrial operations. This requires incorporating secure boot procedures, cryptographic authenticity verification, and efficient protocols specifically tailored for resource-constrained IIoT environments.
- ICS follow a hierarchical architecture, with the Field Device Layer as the foundation, consisting of sensors and actuators responsible for the acquisition and control of real-time data. However, legacy security mechanisms often fail to protect this layer, making IIoT driven ICS susceptible to cyber threats from the beginning of the data life cycle. Adversaries can exploit vulnerabilities in this layer to manipulate sensor data, disrupt processes, or gain unauthorized access to critical infrastructure. To mitigate these risks, a hybrid cryptographic approach must be developed, integrating lightweight encryption mechanisms at the Field Device Layer with stronger post-quantum cryptographic techniques at higher layers. This approach ensures end-to-end security, protecting the integrity, confidentiality, and authenticity of industrial data while enhancing resilience against both classical and quantum cyber threats.
- Another critical research challenge in deploying LWC within IIoT-based ICS is overcoming industry reluctance arising from proprietary concerns, competitive pressures, and cost constraints. Addressing this requires developing strategies that foster trust, transparency, and economically feasible adoption of secure LWC solutions tailored specifically for ICS environments.
- Many industrial systems still rely on old legacy devices that were not built with modern cryptography in mind. Replacing these systems entirely is often too expensive, risky, or simply not possible. To make LWC adoption realistic, we need hybrid solutions that allow legacy systems to be upgraded step by step without interrupting ongoing operations. This includes using secure gateways or lightweight wrappers that can work with older protocols while still improving security. The challenge is to ensure these solutions are compatible, efficient, and easy to deploy across existing infrastructure.

6 Conclusion

This comprehensive study emphasizes the implementation of simple yet highly effective lightweight cryptographic algorithms to improve security paradigm within IIoT systems. Throughout this research, we have methodically defined critical concepts and investigated several approaches, including evaluating symmetric algorithms (block and stream ciphers) and their application to the area of IIoT systems. The core objective of our research is to provide a detailed review of existing lightweight cryptographic algorithms for IIoT systems, presenting their important attributes for security in the IIoT environment. We also identify the vulnerability landscape of lightweight cryptographic algorithms in IIoT systems. Through our well-structured systematic review, we categorized different characteristics related to the deployment of lightweight cryptographic algorithms in IIoT-driven ICS. The findings of this study emphasize the critical need to address security challenges in IIoT environments characterized by resource constraints, real-time processing demands, and scalability issues. We have underscored the necessity for secure, efficient and practical implementation of lightweight cryptographic solutions to ensure data integrity, confidentiality, and resilience in IIoT-driven ICS. Moreover, while lightweight cryptographic solutions have been extensively discussed in the general IoT domain, their specific applicability and challenges in IIoT environments remain underexplored. The lack of a systematic literature review in this field motivated us to conduct this research, which serves as a valuable resource for researchers. We have also addressed the following research questions: (1) How can ICS security practitioners identify the most suitable lightweight cryptographic algorithm for their IIoT environment?; (2) How robust are lightweight cryptographic algorithms against common cryptographic attacks?; (3) Which security research gaps and challenges are yet to be solved in IIoT systems? In our survey, we present unresolved issues, research challenges, and potential directions for future research. Hence, we have addressed both the significant challenges and the evolving issues with LWC in IIoT-driven ICS, representing a significant step toward strengthening cyber resilience in vital industrial ecosystems.

References

[1] H. Gupta, S. K. Singh, S. Kumar, K. Sharma, H. Saini, B. B. Gupta, V. Arya, and K. T. Chui. 2024. Variance-driven security optimisation in industrial IoT sensors. *IET Networks* 14, 1 (2024). DOI: 10.1049/ntw2.12139

[2] M. Ammar, G. Russello, and B. Crispo. 2018. Internet of things: A survey on the security of IoT frameworks. *Journal of Information Security and Applications* 38, 2 (2018), 8–27. DOI: 10.1016/j.jisa.2017.11.002

[3] Brij B. Gupta and Megha Quamara. 2020. An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols. *Concurrency and Computation: Practice and Experience* 32, 21 (2020), e4946.

[4] J. M. A. Bothos and V. Vlachos. 2023. Cybersecurity vulnerability and risk of industrial control systems. In *Hybrid Threats, Cyberterrorism and Cyberwarfare*, M. A. Ferrag, I. Kantzavelou, L. Maglaras, and H. Janicke (Eds.). Boca Raton: CRC Press, 148–165. DOI: 10.1201/9781003314721-8

[5] Suk Kyu Lee, Mungyu Bae, and Hwangnam Kim. 2017. Future of IoT networks: A survey. *Applied Sciences* 7, 10 (2017), 1072.

[6] Tarun Kumar Goyal and Vineet Sahula. 2016. Lightweight security algorithm for low power IoT devices. In *Proceedings of the 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI).* 1725–1729. DOI: http://doi.org/10.1109/ICACCI.2016.7732296

[7] M. M. Aslam, A. Tufail, R. A. A. H. M. Apong, L. C. De Silva, and M. T. Raza. 2024. Scrutinizing security in industrial control systems: An architectural vulnerabilities and communication network perspective. *IEEE Access* 12 (2024), 67537–67573. DOI: 10.1109/ACCESS.2024.3394848

[8] arm.com. 2024. Cortex-M0. Retrieved March 28, 2025 from https://www.arm.com/products/silicon-ip-cpu/cortex-m/cortex-m33. (2024).

[9] microchip.com. 2024. PIC32A Family of Microcontrollers (MCUs). Retrieved March 28, 2025 from https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/32-bit-mcus/pic32a. (2024).

[10] Phithak Thaenkaew, Bruno Quoitin, and Ahmed Meddahi. 2023. Leveraging larger AES keys in LoRaWAN: A practical evaluation of energy and time costs. *Sensors* 23, 22 (2023). DOI: http://doi.org/10.3390/s23229172

[11] Ahmed J. Hintaw, Selvakumar Manickam, Shankar Karuppayah, and Mohammed Faiz Aboalmaaly. 2019. A brief review on MQTT's security issues within the Internet of Things (IoT). *Journal of Communications* 14, 6 (2019), 463–469.

[12] M. Devi and A. Majumder. 2021. Side-channel attack in Internet of Things: A survey. In *Applications of Internet of Things*, J. K. Mandal, S. Mukhopadhyay, and A. Roy (Eds.). Lecture Notes in Networks and Systems, Vol. 137. Springer, 213–222. DOI: 10.1007/978-981-15-6198-6_20

[13] Hamidreza Fereidouni, Olga Fadeitcheva, and Mehdi Zalai. 2025. IoT and man-in-the-middle attacks. *Security and Privacy* 8, 2 (2025), e70016.

[14] Sunil Kumar, Dilip Kumar, Ramraj Dangi, Gaurav Choudhary, Nicola Dragoni, and Ilsun You. 2024. A review of lightweight security and privacy for resource-constrained IoT devices. *Computers, Materials and Continua* 78, 1 (2024), 31–63.

[15] NIST. 2024. Lightweight Cryptography. (Feb 2024). Retrieved March 29, 2025 from https://csrc.nist.gov/projects/lightweight-cryptography

[16] ISO/IEC. 2012. 29192-1:2012, Information technology — Security techniques — Lightweight cryptography. Retrieved February 13, 2025 from https://www.iso.org/standard/56425.html. (Feb 2012).

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

[18] M. Rana, Q. Mamun, and R. Islam. 2022. Lightweight cryptography in IoT networks: A survey. *Future Generation Computer Systems* 129 (2022), 77–89. DOI: 10.1016/j.future.2021.11.011

[19] M. Agrawal, J. Zhou, and D. Chang. 2019. A survey on lightweight authenticated encryption and challenges for securing industrial IoT. In *Security and Privacy Trends in the Industrial Internet of Things*, C. Alcaraz (Ed.). Springer International Publishing, 71–94.

[20] J. H. Kong, L.-M. Ang, and K. P. Seng. 2015. A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments. *Journal of Network and Computer Applications* 49 (2015), 15–50. DOI: 10.1016/j.jnca.2014.09.006

[21] C. P. Ekwueme, I. H. Adam, A. Dwivedi, et al. 2024. Lightweight cryptography for Internet of Things: A review. *EAI Endorsed Transactions on Internet of Things* 10, 1 (2024), 1–9.

[22] Seyyed Keyvan Mousavi, Ali Ghaffari, Sina Besharat, and Hamed Afshari. 2021. Security of internet of things based on cryptographic algorithms: A survey. *Wireless Networks* 27, 2 (2021), 1515–1555.

[23] Vidya Rao and KV Prema. 2021. A review on lightweight cryptography for Internet-of-Things based applications. *Journal of Ambient Intelligence and Humanized Computing* 12, 9 (2021), 8835–8857.

[24] Noor Maher Naser and Jolan Rokan Naif. 2022. A systematic review of ultra-lightweight encryption algorithms. *International Journal of Nonlinear Analysis and Applications* 13, 1 (2022), 3825–3851.

[25] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G. Altman, and PRISMA Group*. 2009. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *Annals of Internal Medicine* 151, 4 (2009), 264–269.

[26] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson. 2023. NIST special publication NIST SP 800-82r3 guide to operational technology (OT) security. NIST, Gaithersburg, MD, USA.

[27] Keith Stouffer, Joe Falco, Karen Scarfone, et al. 2011. Guide to industrial control systems (ICS) security. *NIST Special Publication* 800, 82 (2011), 16–16.

[28] Wei Xu, Yan Gao, Chunfang Yang, and Huaping Chen. 2022. An improved purdue enterprise reference architecture to enhance cybersecurity. In *Proceedings of the 2022 5th International Conference on Blockchain Technology and Applications*. 104–109.

[29] M. K. Hasan, M. Shafiq, S. Islam, B. Pandey, Y. A. Baker El-Ebiary, N. S. Nafi, R. Rodriguez, and D. Vargas. 2021. Lightweight cryptographic algorithms for guessing attack protection in complex internet of things applications. *Complexity* 2021, 1 (2021), 13 pages.

[30] IEC. 2019. IEC 62443. Online. (Feb. 2019). Retrieved August 14, 2025 from https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards

[31] Ryan Dsouza. 2023. Guidance on using ISA/IEC 62443 for IIoT projects. Retrieved August 14, 2025 from https://aws.amazon.com/blogs/iot/guidance-on-using-isa-iec-62443-for-iiot-projects/. (Mar. 2023).

[32] IEC 62351-3:2023. 2023. Power systems management and associated information exchange - Data and communications security - Part 3: Communication network and system security - Profiles including TCP/IP. *International Electrotechnical Commission.* 2023.

[33] ISO/SAE 21434:2021. 2021. Road vehicles – Cybersecurity engineering. *International Organization for Standardization and SAE International.*

[34] IEEE Standard 2030.5-2023. 2023. IEEE standard for smart energy profile application protocol. *Institute of Electrical and Electronics Engineers.*

[35] IMO. 2021. MSC-FAL.1-Circ.3. Online. (FEB 2021). Retrieved August 14, 2025 from https://www.cdn.imo.org/localresources/en/OurWork/Facilitation/Facilitation/MSC-FAL.1-Circ.3-Rev.1%20-%20Guidelines%20On%20Maritime%20Cyber%20Risk%20Management%20(Secretariat).pdf

[36] ENISA. 2020. Guidelines for securing the internet of things. *European Union Agency for Cybersecurity.* https://www.enisa.europa.eu/publications/guidelines-for-securing-the-internet-of-things. Accessed: August 14, 2025.

[37] Kerry McKay, Lawrence Bassham, Meltem Sönmez Turan, and Nicky Mouha. 2016. *Report on Lightweight Cryptography.* Technical Report. National Institute of Standards and Technology.

[38] C. Pei, Y. Xiao, and X. Han. 2018. Trade-off of security and performance of lightweight block ciphers in industrial wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking* 2018, 117 (2018). DOI: http://doi.org/10.1186/s13638-018-1121-6

[39] Chen-Da Liu-Zhang and Ueli Maurer. 2020. Synchronous constructive cryptography. In *Theory of Cryptography: Proceedings of the 18th International Conference, TCC 2020, Part II 18.* Springer, 439–472.

[40] Michael Backes. 2003. Unifying simulatability definitions in cryptographic systems under different timing assumptions. In *Proceedings of the International Conference on Concurrency Theory.* Springer, 350–365.

[41] Sourav Das, Thomas Yurek, Zhuolun Xiang, Andrew Miller, Lefteris Kokoris-Kogias, and Ling Ren. 2022. Practical asynchronous distributed key generation. In *Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP).* IEEE, 2518–2534.

[42] Kiernan Brent George. 2021. *Analysis of Lightweight Cryptographic Primitives.* Ph.D. Dissertation. Virginia Tech.

[43] Amy Demetra Geae Vennos. 2021. *Security of Lightweight Cryptographic Primitives.* Ph.D. Dissertation. Virginia Tech.

[44] NIST-FIPS Standard. 2001. Announcing the advanced encryption standard (AES). *FIPS Publication* 197, 1-51 (2001), 3–3.

[45] Ruby Mishra, Sayantani Dutta, Manish Okade, and Kamalakanta Mahapatra. 2021. Substitution permutation network based lightweight ciphers with improved substitution layers for secure IoT applications. In *Proceedings of the 2nd International Conference on Range Technology (ICORT)*. IEEE, 1–6.

[46] Ramesh Karri, Grigori Kuznetsov, and Michael Goessel. 2003. Parity-based concurrent error detection of substitution-permutation network block ciphers. In *CHES: Proceedings of the 5th International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 113–124.

[47] Isma Norshahila Mohammad Shah, Eddie Shahril Ismail, Faieza Samat, and Normahirah Nek Abd Rahman. 2023. Modified generalized feistel network block cipher for the Internet of Things. *Symmetry* 15, 4 (2023), 900.

[48] JinGyo Song and Seog Chung Seo. 2021. Efficient parallel implementation of CTR mode of ARX-based block ciphers on ARMv8 microcontrollers. *Applied Sciences* 11, 6 (2021), 2548.

[49] Chenqingshui Huang. 2022. AquaMZ: New lightweight authenticated encryption with generalized feistel network based primitive for IoT protocols. In *Proceedings of the 2022 6th International Conference on Computer Science and Artificial Intelligence*. 327–332.

[50] Guangfu Wu, Keke Wang, Jinjun Zhang, and Jiguang He. 2018. A lightweight and efficient encryption scheme based on LFSR. *International Journal of Embedded Systems* 10, 3 (2018), 225–232.

[51] Hatice Kübra Güner, Ceyda Mangır, and Oğuz Yayla. 2023. White-box block cipher implementation based on LS-design. *Cryptology ePrint Archive* (2023).

[52] Aleksandra Mileva, Vesna Dimitrova, Orhun Kara, and Miodrag J. Mihaljević. 2021. *Catalog and Illustrative Examples of Lightweight Cryptographic Primitives*. Springer International Publishing, Cham. DOI: http://doi.org/10.1007/978-3-030-10591-4_2

[53] Elaine Barker and Allen Roginsky. 2018. *Transitioning the use of Cryptographic Algorithms and Key Lengths*. Technical Report. National Institute of Standards and Technology.

[54] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas. 2018. A review of lightweight block ciphers. *Journal of Cryptographic Engineering* 8, 2 (2018), 141–184. DOI: 10.1007/s13389-017-0160-y

[55] J. H. Kong, L.-M. Ang, and K. P. Seng. 2015. A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments. *Journal of Network and Computer Applications* 49 (2015), 15–50. DOI: 10.1016/j.jnca.2014.09.006

[56] Y. Harbi, Z. Aliouat, A. Refoufi, and S. Harous. 2021. Recent security trends in Internet of Things: A comprehensive survey. *IEEE Access* 9 (2021), 113292–113314. DOI: 10.1109/ACCESS.2021.3103725

[57] Hubert Kaeslin. 2008. *Digital Integrated Circuit Design: From VLSI Architectures to CMOS Fabrication*. Cambridge University Press.

[58] R. Kousalya and G. A. Sathish Kumar. 2019. A survey of light-weight cryptographic algorithm for information security and hardware efficiency in resource constrained devices. In *Proceedings of the 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN*). 1–5. DOI: http://doi.org/10.1109/ViTECoN.2019.8899376

[59] L. Diedrich, L. Murati, and A. Wiesmaier. 2015. Stream ciphers in the IoT: Grain and Trivium. *First Workshop on Security and Privacy for Internet of Things and Cyber-Physical Systems (SPIC'15)*.

[60] Abdulrazzaq HA Al-Ahdal et al. 2021. Security analysis of a robust lightweight algorithm for securing data in Internet of Things Networks. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12, 12 (2021), 133–143.

[61] Adrien Benamira, David Gerault, Thomas Peyrin, and Quan Tan. 2021. A deeper look at machine learning-based cryptanalysis. In *EUROCRYPT: Proceedings of the 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 805–835.

[62] D. Stiawan, M. Y. Idris, R. F. Malik, S. Nurmaini, N. Alsharif, R. Budiarto, et al. 2019. Investigating brute force attack patterns in IoT network. *Journal of Electrical and Computer Engineering* 2019 (2019), 1–13. DOI: 10.1155/2019/4568368

[63] Chai Wen, A. Vivegan, L. Samylingam, Irfan Darmawan, P. Siva, Mohd Foozy, Sofia Najwa Ramli, and Janaka Alawatugoda. 2018. Analysis of four historical ciphers against known plaintext frequency statistical attack. *International Journal of Integrated Engineering* 10, 6 (2018), 183–192. DOI: 10.30880/ijie.2018.10.06.026

[64] L. Y. Zhang, Y. Liu, C. Wang, J. Zhou, Y. Zhang, and G. Chen. 2018. Improved known-plaintext attack to permutation-only multimedia ciphers. *Information Sciences* 430 (2018), 228–239.

[65] Achiya Bar-On, Orr Dunkelman, Nathan Keller, and Ariel Weizman. 2019. DLCT: A new tool for differential-linear cryptanalysis. In *EUROCRYPT: Proceedings of the 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Part I 38*. Springer, 313–342.

[66] Mark Randolph and William Diehl. 2020. Power side-channel attack analysis: A review of 20 years of study for the layman. *Cryptography* 4, 2 (2020), 15.

[67] Patrick Karl and Michael Gruber. 2021. A survey on the application of fault analysis on lightweight cryptography. In *Proceedings of the 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 1–3.

[68] Xiaoyang Dong, Jialiang Hua, Siwei Sun, Zheng Li, Xiaoyun Wang, and Lei Hu. 2021. Meet-in-the-middle attacks revisited: Key-recovery, collision, and preimage attacks. In *Advances in Cryptology–CRYPTO 2021: Proceedings of the 41st Annual International Cryptology Conference, CRYPTO 2021, Part III 41*. Springer, 278–308.

[69] S. L. Yeo, D.-P. Le, and K. Khoo. 2021. Improved algebraic attacks on lightweight block ciphers. *Journal of Cryptographic Engineering* 11, 3–4 (2021), 1–19, DOI: 10.1007/s13389-020-00237-4

[70] Ahmed J. Hintaw, Selvakumar Manickam, Mohammed Faiz Aboalmaaly, and Shankar Karuppayah. 2023. MQTT vulnerabilities, attack vectors and solutions in the internet of things (IoT). *IETE Journal of Research* 69, 6 (2023), 3368–3397.

[71] Zhiyong Luo and Xue Zhang. 2020. Research on OPC UA security encryption method. In *Proceedings of the 2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, Vol. 1. IEEE, 287–292.

[72] Indira Kalyan Dutta, Bhaskar Ghosh, and Magdy Bayoumi. 2019. Lightweight cryptography for internet of insecure things: A survey. In *Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 0475–0481.

[73] D. A. F. Saraiva, V. R. Q. Leithardt, D. de Paula, A. S. Mendes, G. V. González, and P. Crocker. 2019. PRISEC: Comparison of symmetric key algorithms for IoT devices. *Sensors* 19, 19 (2019). DOI: 10.3390/s19194312

[74] R. K. Muhammed, R. R. Aziz, A. A. Hassan, A. M. Aladdin, S. J. Saydah, T. A. Rashid, and B. A. Hassan. 2024. Comparative analysis of AES, Blowfish, Twofish, Salsa20, and ChaCha20 for image encryption. *arXiv preprint* arXiv:2407.16274. https://arxiv.org/abs/2407.16274. Accessed: August 14, 2025.

[75] A. Biryukov. 2004. Block ciphers and stream ciphers: The state of the art. *Cryptology ePrint Archive*. Paper 2004/006.

[76] Yi Lu and Serge Vaudenay. 2004. Cryptanalysis of bluetooth keystream generator two-level E0. In *Advances in Cryptology-ASIACRYPT 2004: Proceedings of the 10th International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 483–499.

[77] Yanbin Xu, Yonglin Hao, and Mingxing Wang. 2023. Revisit two memoryless state-recovery cryptanalysis methods on A5/1. *IET Information Security* 17, 4 (2023), 626–638.

[78] M. Dworkin, E. Barker, J. Nechvatal, J. Foti, L. Bassham, E. Roback, and J. Dray. 2001. Advanced Encryption Standard (AES), Federal Inf. Process. Stds.(NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD. (2001).

[79] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. 2007. PRESENT: An ultra-lightweight block cipher. In *CHES 2007: Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 450–466.

[80] Suzan Sallam and Babak D. Beheshti. 2018. A survey on lightweight cryptographic algorithms. In *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE, 1784–1789.

[81] Packetizer. 2023. AES Crypt Downloads. Retrieved March 15, 2025 from https://www.aescrypt.com/download/. (2023).

[82] Rhys Weatherley. 2023. Arduino Cryptography Library,2023a. Retrieved March 16, 2025 from https://rweather.github.io/arduinolibs/crypto.html. (2023).

[83] Matej Sychra. 2023. AES Lab 2023. Retrieved January 16, 2024 from https://docs.arduino.cc/libraries/aeslib/. (2023).

[84] NIST. 2001. Advanced Encryption Standard (AES). Retrieved February 13, 2025 from https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf. (2001).

[85] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. 2016. Ascon v1. 2. *Submission to the CAESAR Competition* 5, 6 (2016), 7.

[86] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. 2015. Cryptanalysis of ASCON. In *Topics in Cryptology—CT-RSA 2015: The Cryptographer's Track at the RSA Conference 2015*. Springer, 371–387.

[87] ASCON. 2023. Ascon-C. Retrieved March 16, 2025 from https://github.com/rweather/ascon-suite. (2023).

[88] Rhys Weatherley. 2024. Ascon-Suite,2023b. Retrieved March 16, 2025 from https://github.com/ascon/ascon-c. (2024).

[89] Mathieu Degré, Patrick Derbez, Lucie Lahaye, and André Schrottenloher. 2024. New models for the cryptanalysis of ASCON. *Cryptology ePrint Archive*, Paper 2024/298. (2024). Retrieved from https://eprint.iacr.org/2024/298. Accessed: August 14, 2025.

[90] C. Jnana Ramakrishna, D. B. K. Reddy, B. K. Priya, P. P. Amritha, and K. V. Lakshmy. 2024. Analysis of lightweight cryptographic algorithms for IoT gateways. *Procedia Computer Science* 233 (2024), 235–242. DOI: 10.1016/j.procs.2024.03.213

[91] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita. 2000. Camellia: A 128-bit block cipher suitable for multiple platforms. *Submission Documents to NESSIE Project*. http://info.isl.ntt.co.jp/camellia. Accessed: August 14, 2025.

[92] NTT and Mitsubishi Electric. 2000. Joint development of next-generation encryption algorithm "Camellia" by NTT and mitsubishi electric-symmetric block cipher achieves high security and world' highest efficiency. (2000). DOI: http://doi.org/en/newsrelease/pdf/news/news00e/0003/000310.pdf. Accessed: August 14, 2025.

[93] M. Matsui, J. Nakajima, and S. Moriai. 2004. RFC 3713: A description of the Camellia encryption algorithm. *Internet Engineering Task Force (IETF)*. https://datatracker.ietf.org/doc/html/rfc3713. Accessed: August 14, 2025.

[94] A. Kato, S. Moriai, and M. Kanda. 2005. RFC 4312: The Camellia cipher and its use with IPsec. *Internet Engineering Task Force (IETF)*. https://datatracker.ietf.org/doc/html/rfc4312. Accessed: August 14, 2025.

[95] Wen-Ling Wu, Wen-Tao Zhang, and Deng-Guo Feng. 2007. Impossible differential cryptanalysis of reduced-round ARIA and camellia. *Journal of Computer Science and Technology* 22, 3 (2007), 449–456. Retrieved from https://jcst.ict.ac.cn/en/article/id/1363

[96] Dongxia Bai and Leibo Li. 2012. New impossible differential attacks on camellia. In *Information Security Practice and Experience*. Mark D. Ryan, Ben Smyth, and Guilin Wang (Eds.), Springer Berlin Heidelberg, Berlin, 80–96.

[97] aead. 2016. Camellia. Retrieved March 23, 2025 from https://github.com/aead/camellia. (2016).

[98] mirkoflchtt. 2017. CamelliaLib. Retrieved March 23, 2025 from https://github.com/mirkoflchtt/CamelliaLib. (2017).

[99] Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata. 2007. The 128-bit blockcipher CLEFIA. In *Proceedings of the 14th International Workshop on Fast Software Encryption, FSE, Revised Selected Papers 14*. Springer, 181–195.

[100] Toru Akishita and Harunaga Hiwatari. 2012. Very compact hardware implementations of the blockcipher CLEFIA. In *Proceedings of the 18th International Workshop on Selected Areas in Cryptography, SAC 2011*. Springer, 278–292.

[101] Alex Biryukov, Adi Shamir, and David Wagner. 2001. Real time cryptanalysis of A5/1 on a PC. In *Proceedings of the 7th International Workshop on Fast Software Encryption, FSE 2000* . Springer, 1–18.

[102] Gaurav Bansod, Nishchal Raval, and Narayan Pisharoty. 2014. Implementation of a new lightweight encryption design for embedded security. *IEEE Transactions on Information Forensics and Security* 10, 1 (2014), 142–151.

[103] Federico Scarpa. 2016. Clefia,2016,. Retrieved February 10, 2025 from https://github.com/fedescarpa/clefia. (2016).

[104] Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bon-Seok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, et al. 2006. HIGHT: A new block cipher suitable for low-resource device. In *CHES 2006: Proceedings of the 8th International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 46–59.

[105] Young-Il Lim, Je-Hoon Lee, Younggap You, and Kyoung-Rok Cho. 2009. Implementation of HIGHT cryptic circuit for RFID tag. *IEICE Electronics Express* 6, 4 (2009), 180–186.

[106] Lilian Bossuet. 2016. Lightweight block cipher implementations,2016. Retrieved March 13, 2025 from https://perso.univ-st-etienne.fr/bl16388h/salware/lightweight_block_cipher.htm. (2016).

[107] Jeffrey Walton. 2023. Cryptopp, 2023. Retrieved March 11, 2025 from https://github.com/weidai11/cryptopp/tree/master. (2023).

[108] Huiju Cheng and Howard M. Heys. 2008. Compact ASIC implementation of the ICEBERG block cipher with concurrent error detection. In *Proceedings of the 2008 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2921–2924.

[109] Yuechuan Wei, Yisheng Rong, and Xu Wang. 2016. Security analysis of cipher ICEBERG against bit-pattern based integral attack. *International Journal of Technology and Human Interaction (IJTHI)* 12, 2 (April 2016), 60–71.

[110] Pass2774. 2023. ESP32IoT-Iceberg. Retrieved March 23, 2025 from https://github.com/pass2774/ESP32IoT-Iceberg. (2023).

[111] Christophe De Canniere, Orr Dunkelman, and Miroslav Knežević. 2009. KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 272–288.

[112] Thomas Eisenbarth et al. 2012. Compact implementation and performance evaluation of block ciphers in ATtiny devices. In *AFRICACRYPT: Proceedings of the 5th International Conference on Cryptology in Africa*. Springer, 172–187.

[113] Lei Wei, Christian Rechberger, Jian Guo, Hongjun Wu, Huaxiong Wang, and San Ling. 2011. Improved meet-in-the-middle cryptanalysis of KTANTAN (poster). In *Proceedings of the 16th Australasian Conference on Information Security and Privacy, ACISP 2011*. Springer, 433–438.

[114] Wenling Wu and Lei Zhang. 2011. LBlock: A lightweight block cipher. In *Proceedings of the 9th International Conference on Applied Cryptography and Network Security, ACNS 2011*. Springer, 327–344.

[115] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. 2011. The LED block cipher. In *CHES: Proceedings of the 13th International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 326–341.

[116] Chae Hoon Lim and Tymur Korkishko. 2005. mCrypton–a lightweight block cipher for security of low-cost RFID tags and sensors. In *Proceedings of the International Workshop on Information Security Applications*. Springer, 243–258.

[117] Chae Hoon Lim and Tymur Korkishko. 2006. mCrypton – a lightweight block cipher for security of low-cost RFID tags and sensors. In *Information Security Applications*. Joo-Seok Song, Taekyoung Kwon, and Moti Yung (Eds.), Springer Berlin Heidelberg, Berlin, 243–258.

[118] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. 2011. Biclique cryptanalysis of the full AES. In *Advances in Cryptology–ASIACRYPT 2011: Proceedings of the 17th International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 344–371.

[119] F. Zhang, Y. Zhang, H. Jiang, X. Zhu, S. Bhasin, X. Zhao, Z. Liu, D. Gu, and K. Ren. 2020. Persistent fault attack in practice. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020, 2 (2020), 172–195. DOI: 10.13154/tches.v2020.i2.172-195

[120] S. Bhasin, J. Breier, X. Hou, D. Jap, R. Poussier, and S. M. Sim. 2020. SITM: See-in-the-middle side-channel assisted middle round differential cryptanalysis on SPN block ciphers. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020, 3 (2020), 95–122.

[121] K. Keerthi, I. Roy, C. Rebeiro, S. Hazra, and S. Bhunia. 2020. FEDS: Comprehensive fault attack exploitability detection for software implementations of block ciphers. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020, 4 (2020), 272–299.

[122] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas. 2018. A review of lightweight block ciphers. *Journal of Cryptographic Engineering* 8, 2 (2018), 141–184. DOI: 10.1007/s13389-017-0160-y

[123] Cihangir Tezcan. 2020. Analysis of ascon, drygascon, and shamash permutations. *International Journal of Information Security Science* 9, 3 (2020), 172–187.

[124] Sinian Luo, Weibin Wu, Yanbin Li, Ruyun Zhang, and Zhe Liu. 2022. An efficient soft analytical side-channel attack on ascon. In *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 389–400.

[125] R. Rohit, K. Hu, S. Sarkar, and S. Sun. 2021. Misuse-free key-recovery and distinguishing attacks on 7-round ASCON. *Cryptology ePrint Archive*, Paper 2021/1629.

[126] André Schrottenloher and Marc Stevens. 2022. Simplified MITM modeling for permutations: New (quantum) attacks. In *Proceedings of the Annual International Cryptology Conference*. Springer, 717–747.

[127] Wei Li, Dawu Gu, Juanru Li, Zhiqiang Liu, and Ya Liu. 2010. Differential fault analysis on camellia. *Journal of Systems and Software* 83, 5 (2010), 844–851. DOI: http://doi.org/10.1016/j.jss.2009.12.019

[128] Marco Macchetti. 2013. Cryptanalysis of AES and camellia with related s-boxes. In *AFRICACRYPT 2013*. Amr Youssef, Abderrahmane Nitaj, and Aboul Ella Hassanien (Eds.), Springer Berlin Heidelberg, Berlin, 208–221.

[129] Y. Li, H. Lin, M. Liang, and Y. Sun. 2021. A new quantum cryptanalysis method on block cipher camellia. *IET Information Security* 15, 6 (2021), 487–495. DOI: http://doi.org/10.1049/ise2.12037

[130] Leibo Li and Keting Jia. 2014. Improved meet-in-the-middle attacks on reduced-round camellia-192/256. *Cryptology ePrint Archive*, Paper 2014/292. (2014). Retrieved from https://eprint.iacr.org/2014/292. Accessed: August 14, 2025.

[131] Onur Özen, Kerem Varıcı, Cihangir Tezcan, and Çelebi Kocair. 2009. Lightweight block ciphers revisited: Cryptanalysis of reduced round PRESENT and HIGHT. In *Proceedings of the 14th Australasian Conference on Information Security and Privacy, ACISP 2009* . Springer, 90–107.

[132] Bonwook Koo, Deukjo Hong, and Daesung Kwon. 2011. Related-key attack on the full HIGHT. In *ICISC 2010: Proceedings of the 13th International Conference on Information Security and Cryptology, Revised Selected Papers 13*. Springer, 49–67.

[133] Junghwan Song, Kwanhyung Lee, and Hwanjin Lee. 2013. Biclique cryptanalysis on lightweight block cipher: HIGHT and Piccolo. *International Journal of Computer Mathematics* 90, 12 (2013), 2564–2580.

[134] Long Wen, Meiqin Wang, Andrey Bogdanov, and Huaifeng Chen. 2014. Multidimensional zero-correlation attacks on lightweight block cipher HIGHT: Improved cryptanalysis of an ISO standard. *Information Processing Letters* 114, 6 (2014), 322–330.

[135] Yue Sun, Meiqin Wang, Shujia Jiang, and Qiumei Sun. 2012. Differential cryptanalysis of reduced-round ICEBERG. In *AFRICACRYPT 2012: Proceedings of the 5th International Conference on Cryptology in Africa*. Springer, 155–171.

[136] Martin Ågren. 2012. Some instant-and practical-time related-key attacks on KTANTAN32/48/64. In *Proceedings of the 18th International Workshop on Selected Areas in Cryptography, SAC 2011, Revised Selected Papers 18*. Springer, 213–229.

[137] Kitae Jeong, Changhoon Lee, and Jong In Lim. 2013. Improved differential fault analysis on lightweight block cipher LBlock for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking* 2013, 1 (2013), 1–9.

[138] Hideki Yoshikawa, Masahiro Kaminaga, Arimitsu Shikoda, and Toshinori Suzuki. 2014. Secret key reconstruction method using round addition DFA on lightweight block cipher LBlock. In *Proceedings of the 2014 International Symposium on Information Theory and Its Applications*. IEEE, 493–496.

[139] Yanfeng Wang, Wenling Wu, Xiaoli Yu, and Lei Zhang. 2012. Security on LBlock against biclique cryptanalysis. In *Proceedings of the 13th International Workshop on Information Security Applications, WISA, Revised Selected Papers 13*. Springer, 1–14.

[140] Guangyao Zhao, Ruilin Li, Lei Cheng, Chao Li, and Bing Sun. 2015. Differential fault analysis on LED using super-sbox. *IET Information Security* 9, 4 (2015), 209–218.

[141] Y. Kim and H. Yoon. 2014. First experimental result of power analysis attacks on a FPGA implementation of LEA. *IACR Cryptology ePrint Archive*. Report 2014/999. https://eprint.iacr.org/2014/999. Accessed: August 14, 2025.

[142] E. Biham. 1994. New types of cryptanalytic attacks using related keys. *Journal of Cryptology* 7, 4 (1994), 229–246. DOI: 10.1007/BF00203965

[143] Junghwan Song, Kwanhyung Lee, and Younghoon Jung. 2014. The security weakness of block cipher piccolo against fault analysis. *International Journal of Distributed Sensor Networks* 10, 3 (2014), 842675. DOI: http://doi.org/10.1155/2014/842675

[144] Jong Hyuk Park. 2009. Security analysis of mCrypton proper to low-cost ubiquitous computing devices and applications. *International Journal of Communication Systems* 22, 8 (2009), 959–969.

[145] Seyyed Arash Azimi, Zahra Ahmadian, Javad Mohajeri, and Mohammad Reza Aref. 2014. Impossible differential cryptanalysis of piccolo lightweight block cipher. In *Proceedings of the 2014 11th International ISC Conference on Information Security and Cryptology*. IEEE, 89–94.

[146] Kitae Jeong, Yuseop Lee, Jaechul Sung, and Seokhie Hong. 2013. Improved differential fault analysis on PRESENT-80/128. *International Journal of Computer Mathematics* 90, 12 (2013), 2553–2563.

[147] Céline Blondeau and Kaisa Nyberg. 2014. Links between truncated differential and multidimensional linear properties of block ciphers and underlying attack complexities. In *Proceedings of the 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 165–182.

[148] Y. Kim and H. Yoon. 2014. First experimental result of power analysis attacks on a FPGA implementation of LEA. *IACR Cryptology ePrint Archive*. Report 2014/999. https://eprint.iacr.org/2014/999. Accessed: August 14, 2025.

[149] Runa Chatterjee and Rajdeep Chakraborty. 2020. A modified lightweight PRESENT cipher for IoT security. In *Proceedings of the 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*. IEEE, 1–6.

[150] Guangyao Zhao, Bing Sun, Chao Li, and Jinshu Su. 2015. Truncated differential cryptanalysis of PRINCE. *Security and Communication Networks* 8, 16 (2015), 2875–2887.

[151] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knežević, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, et al. 2012. PRINCE - A low-latency block cipher for pervasive computing applications (full version). *IACR Cryptology ePrint Archive*. Report 2012/529. https://eprint.iacr.org/2012/529. Accessed: August 14, 2025.

[152] Y. Liu, T. Zang, D. Gu, F. Zhao, W. Li, and Z. Liu. 2020. Improved cryptanalysis of reduced-version QARMA-64/128. *IEEE Access* 8 (2020), 8361–8370. DOI: 10.1109/ACCESS.2020.2964259

[153] Jasmin Kaur, Mehran Mozaffari Kermani, and Reza Azarderakhsh. 2020. Hardware constructions for lightweight cryptographic block cipher QARMA with error detection mechanisms. *IEEE Transactions on Emerging Topics in Computing* 10, 1 (2020), 514–519.

[154] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede. 2014. RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms. *IACR Cryptology ePrint Archive*. Report 2014/084. https://eprint.iacr.org/2014/084. Accessed: August 14, 2025.

[155] S. Sinha and S. Karmakar. 2018. Differential fault analysis of RECTANGLE-80. *IACR Cryptology ePrint Archive*. Report 2018/428, 2018. https://eprint.iacr.org/2018/428. Accessed: August 14, 2025.

[156] R Shantha Mary Joshitta, Lawrence Arockiam, and PD Sheba Kezia Malarchelvi. 2019. Security analysis of SAT_Jo lightweight block cipher for data security in healthcare IoT. In *Proceedings of the 2019 3rd International Conference on Cloud and Big Data Computing*. 111–116.

[157] François-Xavier Standaert, Gilles Piret, Neil Gershenfeld, and Jean-Jacques Quisquater. 2006. SEA: A scalable encryption algorithm for small embedded applications. 222–236. DOI: http://doi.org/10.1007/11733447_16

[158] S. Sadeghi, M. Mahmoudzadeh Niknam, N. Bagheri, and M. R. Aref. 2025. Cryptanalysis of full-round SFN block cipher a lightweight block cipher, targeting IoT systems. *Scientia Iranica* 32, 1 (2025), 9. DOI: 10.24200/sci.2023.59581.6319

[159] S. Sadeghi and N. Bagheri. 2018. Cryptanalysis of SFN block cipher. *IACR Cryptology ePrint Archive*. Report 2018/594. https://eprint.iacr.org/2018/594. Accessed: August 14, 2025.

[160] Kitae Jeong, Yuseop Lee, Jaechul Sung, and Seokhie Hong. 2012. Differential fault analysis on block cipher SEED. *Mathematical and Computer Modelling* 55, 1 (2012), 26–34. DOI: http://doi.org/10.1016/j.mcm.2011.01.008

[161] Tae Hyun Kim, Changkyun Kim, and Ilhwan Park. 2012. Side channel analysis attacks using AM demodulation on commercial smart cards with SEED. *Journal of Systems and Software* 85, 12 (2012), 2899–2908.

[162] H. A. Alkhzaimi and M. M. Lauridsen. 2013. Cryptanalysis of the SIMON family of block ciphers. *IACR Cryptology ePrint Archive*. Report 2013/543. https://eprint.iacr.org/2013/543. Accessed: August 14, 2025.

[163] Harshal Tupsamudre, Shikha Bisht, and Debdeep Mukhopadhyay. 2014. Differential fault analysis on the families of SIMON and SPECK ciphers. In *Proceedings of the 2014 Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE, 40–48.

[164] Reihaneh Rabbaninejad, Zahra Ahmadian, Mahmoud Salmasizadeh, and Mohammad Reza Aref. 2014. Cube and dynamic cube attacks on SIMON32/64. In *Proceedings of the 2014 11th International ISC Conference on Information Security and Cryptology*. IEEE, 98–103.

[165] Yen Yee Chan, Cher-Yin Khor, Je Sen Teh, Wei Jian Teng, and Norziana Jamil. 2022. Differential cryptanalysis of lightweight block ciphers SLIM and LCB. In *Proceedings of the International Symposium on Emerging Information Security and Applications*. Springer, 55–67.

[166] Gautham Sekar, Nicky Mouha, Vesselin Velichkov, and Bart Preneel. 2011. Meet-in-the-middle attacks on reduced-round XTEA. In *Cryptographers' Track at the RSA Conference*. Springer, 250–267.

[167] Mohamed Ahmed Abdelraheem, Julia Borghoff, Erik Zenner, and Mathieu David. 2011. Cryptanalysis of the lightweight cipher A2U2. In *Proceedings of the 13th IMA International Conference on Cryptography and Coding, IMACC 2011.* Springer, 375–390.

[168] A. Kumar and A. Aggarwal. 2012. Lightweight cryptographic primitives for mobile ad hoc networks. In *Recent Trends in Computer Networks and Distributed Systems Security*, S. M. Thampi, A. Y. Zomaya, T. Strufe, J. M. Alcaraz-Calero, and T. Thomas (Eds.). Communications in Computer and Information Science, Vol. 335. Heidelberg: Springer, 240–251. DOI: 10.1007/978-3-642-34135-9_25

[169] Martin Ågren and Martin Hell. 2011. Cryptanalysis of the stream cipher BEAN. In *Proceedings of the 4th International Conference on Security of Information and Networks.* 21–28.

[170] Hui Wang, Martin Hell, Thomas Johansson, and Martin Ågren. 2013. Improved key recovery attack on the BEAN stream cipher. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 96, 6 (2013), 1437–1444.

[171] S. Das and D. RoyChowdhury. 2013. CAR30: A new scalable stream cipher with rule 30. *Cryptography and Communications* 5, 2 (2013), 137–162. DOI: 10.1007/s12095-012-0079-1

[172] Sandip Karmakar, Debdeep Mukhopadhyay, and Dipanwita Roy Chowdhury. 2012. CAvium-strengthening trivium stream cipher using cellular automata. *Journal of Cellular Automata* 7, 2 (2012), 179–197.

[173] A. R. Choudhuri and S. Maitra. 2016. Differential cryptanalysis of Salsa and ChaCha - an evaluation with a hybrid model. *IACR Cryptology ePrint Archive*, 2016/377. https://eprint.iacr.org/2016/377

[174] S. Dey and S. Sarkar. 2021. A theoretical investigation on the distinguishers of Salsa and ChaCha. *Discrete Applied Mathematics* 302 (2021), 147–162. DOI: 10.1016/j.dam.2021.06.017

[175] Bernhard Jungk and Shivam Bhasin. 2017. Don't fall into a trap: Physical side-channel analysis of ChaCha20-Poly1305. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017.* IEEE, 1110–1115.

[176] Zakaria Najm, Dirmanto Jap, Bernhard Jungk, Stjepan Picek, and Shivam Bhasin. 2018. On comparing side-channel properties of AES and ChaCha20 on microcontrollers. In *Proceedings of the 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS).* IEEE, 552–555.

[177] Danang Enggar Risyaf Alam, Habib Al Fitrah, Malika Ayunasari, and Dion Ogi. 2023. Implementation of ChaCha20 algorithm with elliptic-curve-Diffie-Hellman in server room monitoring system to prevent MITM and reused key attack. In *Proceedings of the 2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs).* IEEE, 71–76.

[178] N. Ghafoori and A. Miyaji. 2024. Higher-order differential-linear cryptanalysis of ChaCha stream cipher. *IEEE Access* 12 (2024), 13386–13399.

[179] Martin Hell and Thomas Johansson. 2010. Security evaluation of stream cipher enocoro-128v2. (2010). Retrieved August 14, 2025 from https://lucris.lub.lu.se/ws/portalfiles/portal/5976181/2433492.pdf

[180] Santanu Sarkar, Subhadeep Banik, and Subhamoy Maitra. 2014. Differential fault attack against grain family with very few faults and minimal assumptions. *IEEE Transactions on Computers* 64, 6 (2014), 1647–1657.

[181] Yosuke Todo, Takanori Isobe, Willi Meier, Kazumaro Aoki, and Bin Zhang. 2018. Fast correlation attack revisited: Cryptanalysis on full grain-128a, grain-128, and grain-v1. In *Advances in Cryptology–CRYPTO 2018: Proceedings of the 38th Annual International Cryptology Conference, Part II 38.* Springer, 129–159.

[182] Lin Ding and Jie Guan. 2013. Cryptanalysis of MICKEY family of stream ciphers. *Security and Communication Networks* 6, 8 (2013), 936–941.

[183] Steve Babbage and Matthew Dodd. 2006. The stream cipher MICKEY 2.0. *ECRYPT Stream Cipher* (2006), 191–209.

[184] Shiyong Zhang, Gongliang Chen, and Jianhua Li. 2016. State recovering attack against quavium. In *Proceedings of the 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA).* IEEE, 361–364.

[185] Jonathan A. P. Marpaung, Bruce Ndibanje, and Hoon Jae Lee. 2014. Higher-order countermeasures against side-channel cryptanalysis on rabbit stream cipher. *Journal of Information and Communication Convergence Engineering* 12, 4 (2014), 237–245.

[186] S. Maitra, G. Paul, and W. Meier. 2015. Salsa20 cryptanalysis: New moves and revisiting old styles. *IACR Cryptology ePrint Archive.* https://eprint.iacr.org/2015/217. Accessed: August 14, 2025.

[187] Emam Mohamed, Stanislav Bulygin, and Johannes Buchmann. 2011. Using SAT solving to improve differential fault analysis of trivium. In *Proceedings of the International Conference on Information Security and Assurance, ISA.* Springer, 62–71.

[188] S. Rostami, E. Shakour, M. A. Orumiehchiha, and J. Pieprzyk. 2019. Cryptanalysis of WG-8 and WG-16 stream ciphers. *Cryptography and Communications* 11, 2 (2019), 351–362.

[189] Lin Ding, Chenhui Jin, Jie Guan, and Qiuyan Wang. 2014. Cryptanalysis of lightweight WG-8 stream cipher. *IEEE Transactions on Information Forensics and Security* 9, 4 (2014), 645–652. DOI: http://doi.org/10.1109/TIFS.2014.2307202

[190] Kyoji Shibutani et al. 2011. Piccolo: An ultra-lightweight blockcipher. In *CHES: Proceedings of the 13th International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 342–357.

[191] Philipp Jovanovic. 2012. Piccolo,2012. Retrieved February 8, 2025 from https://github.com/Daeinar/piccolo. (2012).

[192] aczid. 2013. PRESENT arduion. Online. (2013). Retrieved March 31, 2025 from https://github.com/aczid/PRESENT_arduino

[193] Lejla Batina, Amitabh Das, Barış Ege, Elif Bilge Kavun, Nele Mentens, Christof Paar, Ingrid Verbauwhede, and Tolga Yalçın. 2013. Dietary recommendations for lightweight block ciphers: Power, energy and area analysis of recently developed architectures. In *Radio Frequency Identification: Security and Privacy Issues 9th International Workshop, RFIDsec 2013*. Springer, 103–112.

[194] Martin R. Albrecht et al. 2014. Block ciphers–focus on the linear layer (feat. PRIDE). In *CRYPTO: Proceedings of the 34th Annual Cryptology Conference, Part I 34*. Springer, 57–76.

[195] R. Avanzi, S. Banik, O. Dunkelman, M. Eichlseder, S. Ghosh, M. Nageler, and F. Regazzoni. 2023. The QARMAv2 family of tweakable block ciphers. *IACR Transactions on Symmetric Cryptology* 2023, 3 (2023), 25–73. DOI: 10.46586/tosc.v2023.i3.25-73

[196] R. Avanzi. 2017. The QARMA block cipher family. Almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes. *IACR Transactions on Symmetric Cryptology* 2017, 1 (2017), 4–44. DOI: 10.13154/tosc.v2017.i1

[197] Arm Community. 2016. Armv8-A architecture: 2016 additions. (Oct 2016). Retrieved August 14, 2025 from https://community.arm.com/arm-community-blogs/b/architectures-and-processors-blog/posts/armv8-a-architecture-2016-additions

[198] R. Zong and X. Dong. 2016. Meet-in-the-middle attack on QARMA block cipher. *IACR Cryptology ePrint Archive*. Report 2016/1160. https://eprint.iacr.org/2016/1160. Accessed: August 14, 2025.

[199] Jinyan Xu. 2022. Qarma64,2022b. Retrieved March 12, 2025 from https://github.com/Phantom1003/QARMA64. (2022).

[200] Mary Joshitta R. Shantha and L. Arockiam. 2018. Sat_Jo: An enhanced lightweight block cipher for the internet of things. In *Proceedings of the 2018 2nd International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 1146–1150.

[201] Ta Thi Kim Hue, Thang Manh Hoang, and Dat Tran. 2014. Chaos-based S-box for lightweight block cipher. In *Proceedings of the 2014 IEEE 5th International Conference on Communications and Electronics (ICCE)*. IEEE, 572–577.

[202] Franois Mace, Franois-Xavier Standaert, Jean-Jacques Quisquater, et al. 2007. ASIC implementations of the block cipher sea for constrained applications. In *Proceedings of the 3rd International Conference on RFID Security-RFIDSec*, Vol. 2007. 103–114.

[203] H. J. Lee, Sung Jae Lee, J. H. Yoon, Dong Hyeon Cheon, and J. I. Lee. 2005. *The SEED Encryption Algorithm*. Technical Report. KISA.

[204] ISO/IEC. 2010. ISO/IEC 18033-3:2010 Information technology — Security techniques — Encryption algorithms, Part 3: Block ciphers. (Feb 2010). Retrieved February 13, 2025 from https://www.iso.org/standard/54531.html

[205] Jaeho Yoon, Sungjae Lee, D. H. Cheon, Jaeil Lee, and Hyangjin Lee. 2005. The SEED Encryption Algorithm. RFC 4269. (Dec. 2005). DOI: http://doi.org/10.17487/RFC4269

[206] KISA. 2005. SEED encryption algorithm. online. (2005). Retrieved August 14, 2025 from https://seed.kisa.or.kr/kisa/algorithm/EgovSeedInfo.do

[207] openssl. 2018. EVP_CIPHER-SEED. Retrieved March 23, 2025 from https://docs.openssl.org/master/man7/EVP_CIPHER-SEED/#name. (2018).

[208] L. Li, B. Liu, Y. Zhou, and Y. Zou. 2018. SFN: A new lightweight block cipher. *Microprocessors and Microsystems* 60 (2018), 138–150. DOI: 10.1016/j.micpro.2018.04.009

[209] ISO/IEC 29167-21:2018. 2018. Information technology — Automatic identification and data capture techniques Part 21: Crypto suite SIMON security services for air interface communications. (Apr 2018). Retrieved February 13, 2025 from https://www.iso.org/standard/70388.html

[210] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. 2013. The SIMON and SPECK families of lightweight block ciphers. *IACR Cryptology ePrint Archive*. Report 2013/404. https://eprint.iacr.org/2013/404. Accessed: August 14, 2025.

[211] Calvin McCoy. 2018. Simon Speck Ciphers,2018. Retrieved March 14, 2025 from https://github.com/inmcm/Simon_Speck_Ciphers. (2018).

[212] B. Aboushosha, R. A. Ramadan, A. D. Dwivedi, A. El-Sayed, and M. M. Dessouky. 2020. SLIM: A lightweight block cipher for internet of health things. *IEEE Access* 8 (2020), 203747–203757. DOI: 10.1109/ACCESS.2020.3036589

[213] Je Sen. 2022. Slim-cipher,2022. Retrieved March 14, 2025 from https://github.com/CryptoUSM/slim-cipher. (2022).

[214] Michael Appel, Andreas Bossert, Steven Cooper, Tobias Kußmaul, Johannes Löffler, Christof Pauer, and Alexander Wiesmaier. 2016. Block ciphers for the IoT–SIMON, SPECK, KATAN, LED, TEA, PRESENT, and SEA compared. *Proc. Appel Block CF* (2016), 1–37.

[215] Denhart. 2010. Tea Encryption on Arduino,2010. Retrieved February 14, 2025 from https://github.com/dkobia/TEA-encryption-on-arduino. (2010).

[216] Jens-Peter Kaps. 2008. Chai-tea, cryptographic hardware implementations of XTEA. In *Progress in Cryptology-INDOCRYPT 2008: Proceedings of the 9th International Conference on Cryptology in India*. Springer, 363–375.

[217] Michal Protasowicki. 2021. XTEA Cipher,2021. Retrieved February 19, 2025 from https://docs.arduino.cc/libraries/xtea-cipher/?queryID=undefined. (2021).

[218] J. Lu. 2009. Related-key rectangle attack on 36 rounds of the XTEA block cipher. *International Journal of Information Security* 8, 1 (2009), 1–11.

[219] NIST SP 800-232 IPD. 2024. Ascon-Based Lightweight Cryptography Standards for Constrained Devices: Authenticated Encryption, Hash, and Extendable Output Functions. (Nov 2024). Retrieved August 14, 2025 from https://csrc.nist.gov/pubs/sp/800/232/ipd

[220] ISO/IEC 29192-2. 2019. Information security — Lightweight cryptography Part 2: Block ciphers. Online. (Feb 2019). Retrieved August 14, 2025 from https://www.iso.org/standard/78477.html

[221] Julian Harttung. 2017. Mcrypton-vhdl,2017. Retrieved March 10, 2025 from https://github.com/huljar/mcrypton-vhdl. (2017).

[222] Adithya Pokala. 2020. Piccolo Cipher,2020. Retrieved February 12, 2025 from https://github.com/adipokala/piccolo-cipher. (2020).

[223] Julian Harttung. 2016. Prince-vhdl. Retrieved March 02, 2025 from https://github.com/huljar/prince-vhdl. (2016).

[224] Mathieu David, Damith C. Ranasinghe, and Torben Larsen. 2011. A2U2: A stream cipher for printed electronics RFID tags. In *Proceedings of the 2011 IEEE International Conference on RFID*. IEEE, 176–183.

[225] Q. Chai, X. Fan, and G. Gong. 2011. An ultra-efficient key recovery attack on the lightweight stream cipher A2U2. *IACR Cryptology ePrint Archive*. Report 2011/247. https://eprint.iacr.org/2011/247. Accessed: August 14, 2025.

[226] Naveen Kumar, Shrikant Ojha, Kritika Jain, and Sangeeta Lal. 2009. BEAN: A lightweight stream cipher. In *Proceedings of the 2nd International Conference on Security of Information and Networks*. 168–171.

[227] M. Ågren. 2012. On some symmetric lightweight cryptographic designs. *Department of Electrical and Information Technology*. Lund University, Sweden.

[228] J. Jose, S. Das, and D. Roy Chowdhury. 2016. Prevention of fault attacks in cellular automata based stream ciphers. *Journal of Cellular Automata* 12, 1/2 (2016), 141–157.

[229] Charalampos Manifavas, George Hatzivasilis, Konstantinos Fysarakis, and Yannis Papaefstathiou. 2016. A survey of lightweight stream ciphers for embedded systems. *Security and Communication Networks* 9, 10 (2016), 1226–1246.

[230] A. John, S. K. M. Reddy, and J. Jose. 2022. Fault resistant Trivium-like stream cipher using higher radii cellular automata. *Journal of Cellular Automata* 16, 5–6 (2022), 463–490.

[231] marcizhu. 2020. ChaCha20. Retrieved March 23, 2025 from https://github.com/marcizhu/ChaCha20. (2020).

[232] haiwen. 2020. Allow ChaCha20 type ciphers to avoid crippling servers without AES hardware acceleration. Retrieved March 23, 2025 from https://github.com/haiwen/seadroid/issues/903. (2020).

[233] Y. Nir and A. Langley. 2015. ChaCha20 and Poly1305 for IETF Protocols. *RFC 8439*, RFC Editor. DOI: 10.17487/RFC8439

[234] Dai Watanabe, Toru Owada, Kazuto Okamoto, Yasutaka Igarashi, and Toshinobu Kaneko. 2010. Update on enocoro stream cipher. In *Proceedings of the 2010 International Symposium on Information Theory & Its Applications*. IEEE, 778–783.

[235] M. Robshaw. 2008. The eSTREAM project. In *New Stream Cipher Designs: The eSTREAM Finalists*, M. Robshaw and O. Billet (Eds.). Lecture Notes in Computer Science, Vol. 4986. 1–6. DOI: 10.1007/978-3-540-68351-3_1

[236] Saeed Rostami, Mohammad Ali Orumiehchiha, Elham Shakour, and Sadegh Alizadeh. 2025. Fault attack on enocoro stream cipher family. *Journal of Cryptographic Engineering* 15, 1 (2025), 3.

[237] Tiemoko Ballo. 2021. Enocoro128 v2. Online. (2021). Retrieved March 12, 2025 from https://github.com/entropic-security/enocoro128v2

[238] Martin Ågren, Martin Hell, Thomas Johansson, and Willi Meier. 2011. Grain-128a: A new version of grain-128 with optional authentication. *International Journal of Wireless and Mobile Computing* 5, 1 (2011), 48–59.

[239] M. Wroński, E. Burek, and M. Leśniak. 2023. (In) security of stream ciphers against quantum annealing attacks on the example of the Grain-128 and Grain-128a ciphers. *IACR Cryptology ePrint Archive*. Report 2023/1502, 2023. https://eprint.iacr.org/2023/1502. Accessed: August 14, 2025.

[240] Lin Ding and Jie Guan. 2013. Related key chosen IV attack on grain-128a stream cipher. *IEEE Transactions on Information Forensics and Security* 8, 5 (2013), 803–809. DOI: http://doi.org/10.1109/TIFS.2013.2256419

[241] Jonathan Sönnerup, Martin Hell, Mattias Sönnerup, and Ripudaman Khattar. 2019. Efficient hardware implementations of grain-128aead. In *Proceedings of the International Conference on Cryptology in India*. Springer, 495–513.

[242] Jonathan Sönnerup. 2021. Grain128 AEAD. Online. (2021). Retrieved March 12, 2025 from https://github.com/Noxet/Grain-128AEAD

[243] Tim Good and Mohammed Benaissa. 2008. Hardware performance of eStream phase-III stream cipher candidates. In *Proceedings of the Workshop on the State of the Art of Stream Ciphers (SACS'08).*

[244] Mridul Tuteja. 2018. Crypto Mickey 2.0. Online. (2018). Retrieved March 12, 2025 from https://github.com/mridultuteja/crypto-mickey-2.0

[245] Yun Tian, Gongliang Chen, and Jianhua Li. 2012. Quavium-A new stream cipher inspired by trivium. *Journal of Computers* 7, 5 (2012), 1278–1283.

[246] Shiyong Zhang, Gongliang Chen, and Jianhua Li. 2015. Cube attack on reduced-round quavium. In *Proceedings of the 2015 3rd International Conference on Mechatronics and Industrial Informatics (ICMII 2015).* Atlantis Press, 135–139.

[247] A. A. Kuznetsov, O. V. Potii, N. A. Poluyanenko, Y. I. Gorbenko, and N. Kryvinska. 2022. Analysis of standardized algorithms for streaming cryptographic convention, defined in ISO/IEC 18033-4. In *Stream Ciphers in Modern Real-time IT Systems: Analysis, Design and Comparative Studies*, A. A. Kuznetsov, O. V. Potii, N. A. Poluyanenko, Y. I. Gorbenko, and N. Kryvinska (Eds.). Studies in Systems, Decision and Control, Vol. 375. Cham: Springer, 2022, 111–163. DOI: 10.1007/978-3-030-79770-6_8

[248] M. Boesgaard, M. Vesterager, T. Christensen, and E. Zenner. 2005. The stream cipher Rabbit. *ECRYPT Stream Cipher Project Report*, Report 2005/006. 28.

[249] Vaibhav Daga. 2016. Rabbit Cipher. Online. (2016). Retrieved March 13, 2025 from https://github.com/vbdaga/Rabbit-Cipher

[250] Mridul Tuteja. 2024. Rabbit Light Weight Stream Cipher. Online. (2024). Retrieved March 10, 2025 from https://asecuritysite.com/stream/rabbit

[251] D. J. Bernstein. 2008. The Salsa20 family of stream ciphers. In *New Stream Cipher Designs: The eSTREAM Finalists*, M. Robshaw and O. Billet (Eds.). Lecture Notes in Computer Science, Vol. 4986. Heidelberg: Springer. 84–97. DOI: 10.1007/978-3-540-68351-3_8

[252] Diyana Afdhila, Surya Michrandi Nasution, and Fairuz Azmi. 2016. Implementation of stream cipher Salsa20 algorithm to secure voice on push to talk application. In *Proceedings of the 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob).* IEEE, 137–141.

[253] Alex Weber. 2016. Salsa20. Online. (2016). Retrieved March 15, 2025 from https://github.com/alexwebr/salsa20

[254] Nezametdinov Ildus. 2015. Salsa20 C++. Online. (2015). Retrieved March 16, 2025 from https://github.com/everard/Salsa20

[255] N. Mentens, J. Genoe, L. Batina, and I. Verbauwhede. 2008. A low-cost implementation of Trivium. In *Pre-proceedings of SASC 2008 - The State of the Art of Stream Ciphers, K.U. Leuven and IBBT.* 197–204.

[256] José Miguel Mora-Gutierrez et al. 2013. Low power implementation of trivium stream cipher. In *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation: 22nd International Workshop, PATMOS.* Springer, 113–120.

[257] F. E. Potestad-Ordóñez, M. Valencia-Barrero, C. Baena-Oliva, P. Parra-Fernández, and C. J. Jiménez-Fernández. 2020. Breaking Trivium stream cipher implemented in ASIC using experimental attacks and DFA. *Sensors* 20, 23 (2020), 6909. DOI: 10.3390/s20236909

[258] William J. Buchanan. 2024. Trivium Light Weight Stream Cipher. Online. (2024). Retrieved January 28, 2025 from https://asecuritysite.com/encryption/trivium

[259] ISO/IEC. 2012. ISO/IEC 29192-3:2012. Online. (Feb 2012). Retrieved August 14, 2025 from https://www.iso.org/standard/56426.html

[260] ISO/IEC. 2011. ISO/IEC 18033-4:2011, Information technology — Security techniques — Encryption algorithms - Part 4: Stream ciphers. (Feb 2011). Retrieved February 13, 2025 from https://www.iso.org/standard/54532.html

[261] Shih-Hao Chang and Ping-Tsai Chung. 2020. A lightweight authentication stream cypher mechanism for industrial internet of things. In *Proceedings of the 3rd International Conference on SICBS.* Springer, 27–34.

[262] Lin Ding, Chenhui Jin, Jie Guan, and Qiuyan Wang. 2014. Cryptanalysis of lightweight WG-8 stream cipher. *IEEE Transactions on Information Forensics and Security* 9, 4 (2014), 645–652.

[263] NIST . 2024. NIST IR 8547 - Transition to Post-Quantum Cryptography Standards. (2024). DOI: http://doi.org/10.6028/NIST.IR.8547.ipd. Accessed date: August 14, 2025.

[264] NIST. 2024. FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard. (Feb 2024). Retrieved August 14, 2025 from https://csrc.nist.gov/pubs/fips/203/final

[265] NIST. 2024. FIPS 204: Module-Lattice-Based Digital Signature Standard. (Aug 2024). Retrieved August 14, 2025 from https://csrc.nist.gov/pubs/fips/204/ipd

[266] NIST. 2023. FIPS 205: Stateless Hash-Based Digital Signature Standard. (Aug 2023). Retrieved August 14, 2025 from https://csrc.nist.gov/pubs/fips/205/ipd

Received 17 April 2024; revised 11 June 2025; accepted 16 June 2025