Journal of
Cyber Security

Tech Science Press

Doi:10.32604/jcs.2025.073690

ARTICLE

Securing IoT Ecosystems: Experimental Evaluation of Modern Lightweight Cryptographic Algorithms and Their Performance

Mircea Țălu[1,2,*]

[1]Department of Computer Science, Faculty of Automation and Computer Science, The Technical University of Cluj-Napoca, 26–28 George Barițiu St., Cluj-Napoca, 400027, Cluj County, Romania
[2]SC ACCESA IT SYSTEMS SRL, Constanța St., No. 12, Platinia, CP. 400158, Cluj-Napoca, Romania

*Corresponding Author: Mircea Țălu. Email: talu.s.mircea@gmail.com

Received: 23 September 2025; Accepted: 30 October 2025; Published: 11 December 2025

ABSTRACT: The rapid proliferation of Internet of Things (IoT) devices has intensified the demand for cryptographic solutions that balance security, performance, and resource efficiency. However, existing studies often focus on isolated algorithmic families, lacking a comprehensive structural and experimental comparison across diverse lightweight cryptographic designs. This study addresses that gap by providing an integrated analysis of modern lightweight cryptographic algorithms spanning six structural classes—Substitution–Permutation Network (SPN), Feistel Network (FN), Generalized Feistel Network (GFN), Addition–Rotation–XOR (ARX), Nonlinear Feedback Shift Register (NLFSR), and Hybrid models—evaluated on resource-constrained IoT platforms. The key contributions include: (i) establishing a unified benchmarking framework based on standardized evaluation metrics (ROM/RAM usage, throughput, latency, and energy efficiency); (ii) conducting cross-platform experimental validation using Fair Evaluation of Lightweight Cryptographic Systems (FELICS) and the National Institute of Standards and Technology (NIST) reference implementations; and (iii) deriving performance–security trade-offs that map cipher structures to optimal IoT deployment tiers. Results demonstrate that SPN and ARX-based algorithms achieve the best balance between cryptographic strength and implementation efficiency, while Hybrid models exhibit superior adaptability across microcontroller architectures. The findings provide quantitative guidance for selecting lightweight cryptography aligned with hardware capabilities and threat profiles, thereby contributing to the design of scalable and energy-aware IoT security frameworks.

KEYWORDS: Authenticated encryption; Internet of Things (IoT); lightweight cryptography; performance evaluation; resource-constrained devices; security trade-offs

1 Introduction

The Internet of Things (IoT) has revolutionized digital ecosystems by interconnecting billions of diverse devices-from wearable sensors and smart appliances to industrial controllers-via constrained networks such as Bluetooth Low Energy (BLE), Zigbee (a low-power, short-range wireless protocol for IoT devices), Long Range Wide Area Network (LoRaWAN), and Fifth-Generation Mobile Network (5G) [1–3]. Predictions estimate global IoT deployments surpassing 19.8 billion devices by 2025, generating vast volumes of sensitive data requiring robust protection [4,5]. Many IoT devices operate under severe resource constraints, featuring limited processing capabilities, restricted memory sizes (flash and RAM often below 32 KB), and stringent energy budgets, typically relying on batteries or energy harvesting for power [6,7]. As a result, deploying conventional cryptographic primitives (e.g., Advanced Encryption Standard (AES), Rivest–Shamir–Adleman (RSA), Elliptic Curve Cryptography (ECC)) incurs unacceptable overhead in terms of latency, memory usage, and power consumption [8]. Consequently, the field of lightweight cryptography has emerged, focusing on compact, energy-aware algorithms that uphold confidentiality, integrity, and authenticity while meeting stringent IoT device limitations [9,10].

Lightweight cryptography includes five main categories: block ciphers, stream ciphers, hash functions, message authentication codes (MACs), and authenticated encryption schemes. These algorithms are carefully optimized to operate efficiently across different architectural platforms, including both hardware and software environments [2,10–12].

Implementations of lightweight cryptographic (LWC) algorithms are stratified according to the resource constraints and computational capabilities of the target platforms, typically divided into three primary classes: ultra-lightweight, low-cost, and lightweight systems [10].

Ultra-lightweight implementations are designed for highly constrained microcontrollers, such as the standard 8-bit Intel 8051 and Atmel's ATtiny45, characterized by very limited processing power and minimal memory resources. Low-cost implementations target moderately capable devices, for example, the ATmega128, which offer a balance between performance and resource availability. The lightweight category comprises more capable devices with comparatively greater memory and computational capacities, accommodating more complex cryptographic operations [2]. From a hardware perspective, classification hinges primarily on the number of logic gates and the associated chip area utilized during implementation. In contrast, software-based categorizations rely on the footprint of program and data memory, particularly the demands on ROM and RAM. These implementation criteria are essential in guiding the optimal selection of lightweight cryptographic schemes aligned with the resource limitations and functional requirements of targeted IoT environments. Table 1 shows a summary of these classifications, delineating the resource thresholds for each category in both hardware and software domains.

Table 1: Comparative classification of lightweight cryptographic implementations [2]

Category	Hardware characteristics	Software characteristics	Typical platforms	CPU frequency range	Power profile
Ultra-Lightweight	- Logic gate count: **≤1000 GE** - Minimal silicon area (~0.5–1 mm^2) - No hardware acceleration	- ROM: ≤4 KB - RAM: ≤256 Bytes - Code size: ≤2 KB - Low instruction complexity	Intel 8051, ATtiny45, PIC10F200	1–8 MHz	~<1 mW (ultra-low power)
Low-Cost	- Logic gate count: ≤2000 GE - Moderate silicon area (~1–2 mm^2) - Optional crypto coprocessors	- ROM: ≤4 KB - RAM: ≤8 KB - Code size: 2–6 KB	ATmega128, MSP430	4–16 MHz	~5–15 mW (low power)
Lightweight	- Logic gate count: ≤3000 GE - Larger area (~2–5 mm^2) - Integrated security modules	- ROM: ≤32 KB - RAM: ≤8 KB - Code size: up to 12 KB	ARM Cortex-M0/M3, STM32F0, RISC-V RV32E	16–80 MHz	~10–50 mW (moderate power efficiency)

Despite the growing number of lightweight cryptographic algorithms, selecting the most suitable solution for resource-constrained IoT devices remains challenging. Existing studies often focus on individual ciphers or limited platforms, lacking comprehensive experimental validation. Motivated by this gap, our work systematically benchmarks and analyzes modern lightweight cryptographic algorithms across multiple structural classes, offering actionable insights for deployment in heterogeneous IoT ecosystems.

The efficiency of a lightweight cryptographic algorithm can be achieved through two primary approaches: either by optimizing existing cryptographic primitives to meet stringent resource constraints, or by designing bespoke lightweight schemes from the ground up. The latter involves deliberate simplifications, such as reduced key lengths, minimized internal states, compact functional components, streamlined round transformations, and simplified key scheduling mechanisms [12].

The development of novel LWC algorithms necessitates a delicate balance among three fundamental attributes: cryptographic strength, implementation efficiency, and operational performance [13]. These core objectives must be quantitatively assessed using a comprehensive set of evaluation metrics that reflect both the algorithm's feasibility for constrained environments and its resilience against potential threats.

Implementation cost is typically expressed in terms of physical hardware area, measured in Gate Equivalents (GE), alongside memory usage (RAM/ROM) and energy consumption metrics—factors that are particularly critical for battery-operated and energy-harvesting IoT nodes [14–16]. Recent studies have addressed these aspects further, proposing hierarchical key management methods [17] and energy-efficient cryptographic architectures for IoT–cloud networks [18].

Nevertheless, these metrics are significantly influenced by the underlying hardware technology, making it difficult to conduct an accurate and meaningful comparison of lightweight algorithm implementations across different platforms [12].

Performance characteristics encompass indicators such as latency, representing the delay in processing a cryptographic operation, and throughput, which quantifies the rate of data encryption or decryption over time [19–21].

Security, as a cornerstone of algorithmic assessment, is gauged through the security level (e.g., in bits of resistance) and resistance to various cryptanalytic techniques including, but not limited to, linear, differential, algebraic, and side-channel attacks [22–25]. To provide a unified view of an algorithm's viability, integrated indicators such as the efficiency ratio (e.g., throughput per unit of area or energy) and the figure of merit (FoM) are employed. These metrics offer insights into the trade-offs between robustness, speed, and resource utilization, enabling more informed decisions during the selection or design of ciphers for constrained environments [18,26].

A synopsis of widely accepted performance metrics and their mathematical formulations is presented in Table 2, serving as a foundation for objective comparison and standardized benchmarking of LWC implementations.

Table 2: Evaluation metrics for lightweight cryptographic algorithms in IoT systems

Metric	Definition and formula	Practical implementation note	Importance of IoT devices
Security level (bits)	Represents resistance to cryptanalysis; defined as the base-2 logarithm of the estimated effort required to break the cipher (e.g., 2^{128} operations for 128-bit security).	Usually equivalent to key length, unless the algorithm design reduces effective security.	Ensures long-term resilience against brute-force and differential attacks.
Silicon area (GE)	Total physical chip space occupied by the implementation, measured in Gate Equivalents. Calculated as: $GE = (A_{total})/(A_{NAND})$ (1) where: total A_{total} is the total silicon area of the circuit (in μm^2), and A_{NAND} is the area of a two-input NAND gate (in μm^2).	Lower values imply compact hardware, suitable for space-constrained embedded systems.	Vital for integration into RFID tags, sensors, and wearables.
Processing throughput	Speed of data encryption/decryption, typically measured in kilobits per second (kbps). $Throughput = \frac{(Bc)\cdot(Cf)}{Cc}$ (2) where: Bc is the block size; Cf is the clock frequency; and Cc is the cycle count.	Common frequencies: 100 kHz (hardware), 4 MHz (software).	Influences system responsiveness, especially in real-time applications.
Latency (clock cycles)	The number of processor cycles required to complete one cryptographic operation on a block of data.	Lower latency means faster per-block processing time.	Determines performance in time-critical protocols (e.g., handshake, authentication).
Power consumption (μW)	Electrical power drawn during operation. Depends on hardware platform and design size (GE).	Average microcontroller power: 0.004 μW (8-bit), 0.00135 μW (16-bit).	Impacts battery usage, thermal limits, and environmental sustainability.
Energy per bit (μJ/bit)	Energy needed to process a single bit, (in μJ). $Energy = \frac{(Latency)\cdot(Power)}{Bc}$ (3) where: Bc is the block size.	Lower energy is better for long-term deployments, energy-harvesting systems.	Critical for IoT nodes with energy constraints or no recharge capability.

(Continued)

Table 2 (continued)

Metric	Definition and formula	Practical implementation note	Importance of IoT devices
Memory footprint (RAM/ROM)	Memory used during execution (RAM) and code storage size (ROM), both in bytes or kilobytes.	RAM holds intermediate values; ROM holds static code.	Affects compatibility with low-memory microcontrollers and edge devices.
Hardware efficiency	Measures throughput relative to implementation size. $Efficiency = \frac{Throughput}{Area}$ (4) where: *Area* is in kGE.	Higher values suggest better resource utilization on silicon.	Facilitates optimal hardware deployment in low-cost IoT applications.
Software efficiency	Indicates the balance between throughput and software code size. $Efficiency = \frac{Throughput}{Codesize}$ (5) where: *Code size* is in KB.	Relevant when evaluating portable software implementations.	Crucial for constrained OS-based platforms like TinyOS or Contiki.
Adaptability and modularity	Ability of the cipher to adjust for different use-cases, key/block sizes, or operational modes.	Configurable ciphers are more reusable across varied deployments.	Supports customization and future-proofing in IoT ecosystems.
Side-channel resistance	Resistance against implementation-specific attacks such as power analysis, timing analysis, EM leaks, or fault injection.	Requires dedicated countermeasures (masking, constant-time logic, etc.).	Essential in physically accessible devices like smart locks or industrial nodes.
Figure of merit (FOM)	Composite metric reflecting performance vs area. $FOM = \frac{Throughput}{(Area)^2}$ (6)	Independent of specific technology; allows cross-platform comparison.	Useful for selecting optimal algorithms under strict design constraints.

To be deemed robust and suitable for deployment in constrained environments, any lightweight cryptographic algorithm must uphold four fundamental security objectives: data confidentiality, message integrity, entity authentication, and non-repudiation. These properties collectively ensure that information remains protected from unauthorized access, tampering, impersonation, and denial of origin. From an architectural perspective, lightweight block ciphers are designed with efficiency as a core principle. This is often achieved by reducing the size of the data blocks and cryptographic keys, simplifying the structure of encryption rounds, and minimizing the complexity of the key schedule mechanism. Unlike standard ciphers such as AES or DES, which prioritize high-security margins, lightweight alternatives focus on achieving a balance between security and the stringent resource limitations of embedded or IoT platforms [27].

This study provides a focused performance assessment of LWC algorithms on resource-constrained IoT platforms, using benchmarking to evaluate execution time, memory usage, and power efficiency. The analysis supports informed decisions by researchers and practitioners, ensuring that cryptographic designs align with the practical limitations and security demands of modern IoT environments.

2 Analytical Foundations of Lightweight Cryptography for IoT Devices

2.1 Structural and Key-Based Classification of Lightweight Cryptography

In IoT ecosystems, LWC algorithms are carefully selected based on two fundamental criteria: key management (symmetric vs. asymmetric) and structural design (cryptographic primitives) [2,7,28]. Together, these classifications inform the choice of suitable algorithms based on memory footprint, energy efficiency, latency, and resistance to cryptanalysis.

A. Key-oriented classification

Symmetric cryptography involves a shared secret key for both encryption and decryption. These algorithms (e.g., AES, PRESENT) offer high performance and are computationally inexpensive, making them ideal for low-power IoT nodes. The key challenge lies in secure key distribution. This method is renowned for its speed and efficiency; however, it requires secure key distribution—a challenge often solved by pre-sharing keys via trusted intermediaries or secure channels. When implemented with authenticated modes, symmetric ciphers can also guarantee confidentiality, integrity, and authentication.

Asymmetric cryptography utilizes a public/private key pair and provides authentication and non-repudiation, commonly via digital signatures. This approach simplifies secure key exchange and authentication, though it is slower and requires longer keys, making it less suitable for environments with limited computational power. Algorithms like RSA, ECC, and Kyber offer robust security but are costly in terms of computation and memory, rendering them less favorable in constrained IoT environments (NIST SP 800-232).

B. Structural classification of LWC algorithms

LWC algorithms can be structurally classified into four major categories: Substitution-Permutation Networks (SPNs), sponge-based constructions, ARX ciphers, and Authenticated Encryption with Associated Data (AEAD) algorithms. Each of these structural types offers unique design characteristics optimized for constrained environments such as those found in IoT systems [29–31].

- Substitution-Permutation Networks (SPNs): SPNs, exemplified by algorithms like AES, PRESENT, and SPECK, are structured around iterative rounds of substitution (S-boxes) and permutation (P-boxes). These operations introduce confusion and diffusion—two essential principles for strong encryption. SPNs are highly suitable for lightweight applications due to their simple hardware implementation, strong security properties, and efficient use of memory and processing resources.
- Sponge-Based Constructions: Originating from the Keccak algorithm—selected for the SHA-3 standard—sponge-based designs are adaptable structures that absorb input data and squeeze out output through a single permutation function. These constructions provide both hashing and encryption capabilities, making them versatile for IoT use cases. Sponge functions also exhibit strong resistance to cryptanalytic attacks and allow for configurable parameters to balance performance and security based on the application.
- ARX Ciphers (Addition, Rotation, XOR): ARX-based algorithms, including ChaCha20, Salsa20, HIGHT, and TEA, employ only three simple operations—modular addition, bitwise rotation, and XOR. These operations are highly efficient on low-power processors and offer straightforward software and hardware implementation. Although they may lack the confusion benefits of substitution

boxes, ARX ciphers are valued for their minimal code size, high speed, and suitability for real-time embedded systems.

- Authenticated Encryption with Associated Data (AEAD): AEAD algorithms such as AES-GCM, ChaCha20-Poly1305, and OCB provide both confidentiality and message integrity in a single operation. These schemes encrypt the payload while simultaneously producing an authentication tag that ensures the data has not been tampered with. AEAD constructions are essential in environments requiring secure, authenticated communication and are often optimized to support parallel and streaming modes for higher throughput.

The diversity of these structural paradigms enables designers to tailor cryptographic solutions to specific IoT requirements, considering trade-offs between security, performance, memory consumption, and power efficiency. Structural decisions in LWC design are further influenced by implementation contexts—whether software-driven on microcontrollers or hardware-focused on ASIC/FPGA platforms. Each structure addresses different dimensions of the cryptographic standards.

Lightweight symmetric ciphers are essential for securing IoT systems, where efficiency and low overhead are critical. These algorithms are categorized by their internal structural design, which directly influences performance, implementation cost, and resistance to attacks. Table 3 summarizes the main structural classes of lightweight block ciphers, highlighting representative algorithms and their design trade-offs.

Table 3: Structural classification of lightweight block ciphers

Structure type	**Description**	**Representative algorithms**	**Remarks**
SPN (Substitution–Permutation Network)	Utilizes substitution and permutation layers for high confusion and diffusion. Optimized for hardware.	AES, PRESENT, GIFT, SKINNY, LED, PRINCE, Midori, Rectangle	Offers strong security and parallelism; widely used in block cipher designs.
FN (Feistel Network)	Processes data in two halves, with one side modified and the other retained per round.	DESL, Simon, LBlock, MIBS, TEA, KASUMI	Flexible and efficient; decryption uses the same logic as encryption.
GFN (Generalized Feistel Network)	An extension of the Feistel structure with more than two branches, improving diffusion and scalability.	CLEFIA, Piccolo, TWINE, MARS	Flexible architecture; suitable for both lightweight and high-throughput designs.
ARX (Addition–Rotation–XOR)	Combines simple arithmetic and logical operations without S-boxes; ideal for software.	SPECK, HIGHT, LEA, IDEA	Highly efficient; software-friendly, though some designs are controversial.
NLFSR (Nonlinear Feedback Shift Register)	Uses nonlinear shift registers, often for stream ciphers; minimal hardware.	KATAN, KTANTAN, Grain, Trivium, Achterbahn	Ultra-low area usage; popular in passive RFID and sensor devices.
Hybrid	Combines multiple structural principles (e.g., SPN + LFSR) to balance efficiency and security.	Hummingbird, Hummingbird-2, Enocoro	Custom-designed for specific constraints; aims at trade-offs in design.

C. Core primitives of symmetric lightweight cryptographic algorithms

In the realm of symmetric cryptography—widely adopted for resource-constrained IoT devices—algorithms are typically grouped by how they process input data: block ciphers, stream ciphers, and cryptographic hash functions [32–34].

Block ciphers operate on data in fixed-length segments, transforming each block through a series of operations governed by a secret key. Their design hinges on two essential principles: confusion, which obscures the relationship between the ciphertext and the key, and diffusion, which spreads the influence of each input bit across the output. Together, these mechanisms ensure robustness against pattern recognition and statistical inference attacks.

In contrast, stream ciphers handle plaintext in a bitwise or bytewise manner, encrypting data in a continuous flow rather than segmented blocks. These ciphers typically rely on confusion mechanisms, employing pseudorandom keystreams to mask the input. While stream ciphers can offer higher performance in certain environments due to lower computational overhead, they often provide reduced resistance against certain attack vectors compared to block ciphers, especially when keys or initialization vectors are reused.

A distinct class within symmetric cryptography is represented by hash functions. These algorithms map arbitrary-length input data to a fixed-length string, known as the hash or digest. As one-way functions, they are designed to be irreversible—making it computationally infeasible to recover the original input from its digest. Hash functions are not used for encryption but are critical for ensuring data integrity and supporting authentication mechanisms, such as digital signatures or message authentication codes (MACs), particularly in IoT protocols.

Each of these symmetric primitives contributes to the lightweight cryptographic landscape by balancing security, efficiency, and hardware adaptability, making them indispensable for modern IoT applications.

2.2 Evaluation Frameworks and Standardization Efforts in Lightweight Cryptography

To ensure the robustness, efficiency, and real-world applicability of LWC algorithms, structured evaluation using standardized benchmarking tools is indispensable. These tools assess algorithmic performance under the stringent constraints typical of IoT and embedded systems, offering reproducible metrics on memory usage, throughput, latency, energy consumption, and resistance to side-channel attacks.

2.2.1 Software Evaluation Frameworks

Early initiatives such as FELICS (Fair Evaluation of Lightweight Cryptographic Systems) [35] laid the foundation for performance evaluation on embedded microcontrollers. Supporting 8-bit AVR, 16-bit MSP430, and 32-bit ARM Cortex-M platforms, FELICS offers standardized metrics including ROM/RAM footprint, execution cycles, and energy consumption. Optimized implementations for ciphers such as LEA, SIMON, and SPECK have been integrated to exploit efficient bitwise operations [36,37].

The BLOC project [38], targeting MSP430 platforms, evaluated 12 lightweight ciphers, revealing key trade-offs in speed, memory footprint, and security margins. These early efforts emphasized that no algorithm is universally optimal across all IoT scenarios.

XBX (eXternal Benchmarking eXtension), extending SUPERCOP, brings benchmarking to non-POSIX environments such as smartcards and constrained embedded systems, ensuring broader platform support and cross-comparability.

2.2.2 Hardware Evaluation Tools

For evaluating Application-Specific Integrated Circuit (ASIC) and Field-Programmable Gate Array (FPGA) implementations, ATHENA provides a complete hardware evaluation suite. It assesses parameters such as GE, energy usage, and clock frequency, aiding design decisions for resource-limited embedded systems. ATHENA has been utilized by both academic researchers and standardization bodies, such as the Cryptographic Evaluation Center (CRYPTREC), for evaluating the physical footprint of LWC candidates. The Cryptographic Engineering Research Group (CERG) at George Mason University [39] has further enriched the landscape by releasing side-channel attack (SCA)-protected hardware description language (HDL) implementations of National Institute of Standards and Technology (NIST) finalists on FPGA platforms, such as the Artix-7 model. Their work measured area, latency, throughput, and cost efficiency under both unprotected and hardened configurations, contributing directly to the NIST LWC evaluation corpus.

2.2.3 NIST Standardization Benchmarks

During the NIST LWC Standardization Process (2018–2023), eight finalist ciphers were benchmarked on AVR (a family of microcontrollers developed by Atmel, now part of Microchip Technology), ARM Cortex-M (a family of microprocessor cores designed by ARM Holdings for low-power embedded systems), and Reduced Instruction Set Computing-Five (an open-source instruction set architecture for CPUs) RISC-V platforms [40]. These assessments encompassed both functional performance and side-channel resistance, incorporating timing noise, energy profiling, and implementation security metrics. ASCON, selected as the NIST standard, was rigorously evaluated in both software and hardware environments [41].

Parallel analyses, including the fault and SCA studies by Mohajerani et al. [39], demonstrated that candidates exhibit distinct performance-security trade-offs depending on optimization targets (e.g., latency, side-channel leakage, gate count).

2.2.4 Comparative Surveys and Meta-Analyses

Recent surveys provide a panoramic view of LWC performance across over 50 cipher implementations. Kaur et al. [41] delivered a comprehensive summary of ASCON, including energy efficiency, gate count, and resistance to power/fault attacks. Aagaard et al. explored sponge-based primitives such as Wide-Area Group Encryption (WAGE) [42] and ACE [43], focusing on Application-Specific Integrated Circuits (ASICs)-level performance and authenticated encryption properties. Hosseinzadeh and Bafghi [44] highlighted software implementation performance from energy and memory perspectives. Broader bibliometric and empirical reviews by Dewamuni et al. [45], Sarker [46], and Buchanan & Maglaras [47] synthesized data across platforms, emphasizing sustainable cryptography for real-world IoT applications. These collective insights reveal that optimal algorithm selection depends on specific deployment priorities—whether minimal memory usage, low latency, energy efficiency, or robust protection against cryptanalytic and side-channel attacks.

2.2.5 Role of Standardization and Industrial Participation

A global consortium of academic institutions, standardization bodies (e.g., NIST, International Organization for Standardization/International Electrotechnical Commission (ISO/IEC), Cryptographic Evaluation Center (CRYPTREC), European Network of Excellence in Cryptography (ECRYPT-NET)), and technology companies (e.g., Google, Sony, Intel) continues to advance the field. These stakeholders have driven the development of unified evaluation criteria and contributed to the evolution of cryptographic protocols tailored to the operational realities of IoT ecosystems [48].

2.3 Cryptanalytic Threat Vectors Targeting Lightweight Cipher Designs

While lightweight cryptographic primitives are tailored for environments with severe resource constraints, they are not immune to sophisticated cryptanalytic threats.

1. Key space exhaustion (brute-force) attacks. These attacks attempt all possible keys to find the correct one. While computationally impractical for keys ≥128 bits, reduced-round or atypical key setups may remain susceptible in certain cases [49].
2. Precomputation and lookup attacks. Techniques like dictionary and rainbow-table attacks leverage precomputed plaintext-ciphertext pairs to expedite key recovery, especially when keys are reused or plaintext is repetitive. Despite high memory demands, they pose threats when key rotation is rare.
3. Differential cryptanalysis. This method exploits how input differences affect ciphertext differences to statistically reveal key information. Advanced variants like impossible differential and boomerang attacks effectively target reduced-round lightweight ciphers such as GIFT and SKINNY [50].
4. Algebraic and linear attacks. Algebraic and linear attacks: Linear cryptanalysis uses linear approximations of nonlinear components, while algebraic attacks represent ciphers as systems of nonlinear multivariate equations. Though solving these is Non-deterministic Polynomial-time hard (NP-hard), methods involving Gröbner bases and Satisfiability (SAT) solvers can break weak designs, especially when round functions lack strong nonlinearity or diffusion [51].
5. Side-channel and fault injection attacks. Side-channel attacks (e.g., Differential Power Analysis (DPA), Electromagnetic Analysis (EMA), timing) exploit physical leakage during encryption to extract secrets. Fault injection attacks cause intentional errors to disrupt internal states. Even NIST LWC finalists like ASCON and GIFT-COFB have demonstrated vulnerabilities, highlighting the need for hardware-level countermeasures [52].

To resist a wide range of security attacks, LWC systems rely on six fundamental design strategies: (SPN, FN, GFN, ARX, NLFSR, and Hybrid)—each offering a unique internal structure that plays a crucial role in maintaining the overall security of the algorithm [48].

3 Historical Trends of LWC Algorithms for IoT Devices

The field of lightweight cryptography has evolved significantly in response to the growing need for securing pervasive, resource-constrained devices, especially in the context of the IoT. As IoT devices proliferate, they face challenges that traditional cryptographic algorithms were not designed to address—specifically, the constraints imposed by limited computational power, memory, and energy resources. This has prompted the development of cryptographic schemes that are tailored to meet the demands of IoT environments, offering robust security with minimal resource consumption.

3.1 Early Adaptation of Conventional Ciphers

In the initial stages of integrating cryptography into resource-constrained environments, conventional ciphers like AES were adapted to fit the needs of smaller, low-power devices. AES-128, a widely trusted standard, became a security benchmark for many systems. However, its computational complexity and memory overhead posed significant challenges, especially for devices running on 8-bit or 16-bit microcontrollers with extremely limited resources. This led to an ongoing search for alternatives that could balance security with efficiency. As such, many IoT platforms began adapting AES in less resource-intensive ways, including reducing key sizes, simplifying block structures, or relying on hardware acceleration to make the algorithm more feasible. However, despite these optimizations, AES was still too resource-heavy for many constrained systems, further catalyzing the search for LWC solutions [32].

3.2 Emergence of Native Lightweight Block Ciphers

A significant milestone in lightweight cryptography was the introduction of PRESENT in 2007, developed by Bogdanov et al. [53] from Orange Labs, Ruhr-University Bochum, and TU Denmark. PRESENT uses a substitution-permutation network (SPN) structure with a 64-bit block and 80-/128-bit keys. Its silicon area requirement is approximately 1570 GE, significantly smaller than AES, making it ideal for hardware-constrained applications. PRESENT is standardized in ISO/IEC 29192-2:2019 [53,54].

In 2013, the US National Security Agency (NSA) released two lightweight block ciphers: SIMON (optimized for hardware) and SPECK (optimized for software). Both support various block/key sizes and are designed for minimal gate count, fast execution, and adaptability to constrained environments. The design philosophy behind both ciphers was to strike a balance between security and efficiency, making them valuable for IoT and embedded systems where both performance and resource efficiency are paramount [55].

3.3 Authenticated Encryption and the Rise of ASCON

To address the growing needs for data authentication, ASCON was introduced in 2014 by researchers from TU Graz, Infineon, and Radboud University. It is based on a sponge construction and supports both AEAD and hashing, offering strong resistance against cryptanalytic attacks [56]. ASCON was selected as a finalist in the Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR) and became the official NIST standard for LWC in 2023 [49].

4 Experimental Validation of LWC Algorithms on Embedded Platforms

To complement theoretical models and cryptanalytic assessments, various researchers have performed comprehensive experimental studies to evaluate the real-world performance of lightweight cryptographic algorithms on a range of embedded platforms.

Choubey et al. [57] show a compact, high-frequency hardware architecture for the GIFT-COFB cipher (a lightweight cryptographic algorithm optimized for resource-constrained environments), specifically tailored for IoT devices, where lightweight cryptography is essential due to stringent resource constraints such as limited power, memory, and processing capabilities. The architecture demonstrates superior performance in terms of frequency (operating speed), latency (delay in processing), and area (the physical space required on the chip) when implemented on Xilinx Vertex-7 and Kintex-7 FPGA boards, which are customizable hardware platforms that allow for efficient hardware-based cryptographic operations. Abdel-Halim and Zayan [58] evaluated the performance of lightweight block ciphers GIFT-COFB, Romulus (a lightweight cipher designed for both security and efficiency), and TinyJAMBU (a highly efficient lightweight block cipher for IoT devices) on the Arduino Due (a microcontroller board based on the ARM Cortex-M3 processor), highlighting TinyJAMBU and GIFT-COFB as optimal candidates for IoT devices based on execution time (how fast the algorithm processes data), throughput (the rate at which data is processed), power consumption (the amount of energy used), and memory efficiency (the amount of memory required for the algorithm).

Beg et al. [59] categorize lightweight cryptographic algorithms and simulate selected candidates using IoT-relevant metrics, addressing the challenge of identifying suitable security solutions for resource-constrained IoT applications in industrial and military domains. Regla and Festijo [60] conduct a systematic review of lightweight cryptography algorithms for IoT, comparing their security levels, performance metrics, and resource usage to provide insights into their suitability for smart devices and to guide future research in the field. Singh et al. [61] show an analysis of lightweight cryptographic primitives—including block ciphers, stream ciphers, and hash functions—evaluating their structural parameters and security suitability for constrained IoT environments, while also proposing a security scheme to address ongoing challenges in power, memory, and performance efficiency. Desai [62] provides a comprehensive review of lightweight cryptographic techniques—including symmetric ciphers (PRESENT, SPECK, SIMON), asymmetric schemes (ECC, NTRUEncrypt), and hash functions (SPONGENT, PHOTON, QUARK)—evaluating their performance and security in IoT environments, while addressing implementation challenges and proposing future directions such as hybrid models and AI-driven optimizations. Ramakrishna et al. [63] investigated lightweight encryption algorithms in IoT settings using real-time data and the Message Queuing Telemetry Transport (MQTT) protocol, evaluating encryption time, memory, and CPU usage to identify optimal cryptographic solutions through standardized parameters such as key width and block length. Bhardwaj et al. [64] highlight security vulnerabilities in IoT ecosystems and emphasize the role of lightweight cryptography for secure communication, presenting a comparative analysis of encryption and authentication algorithms that demonstrates their advantages over conventional methods in terms of memory usage, computational efficiency, and power consumption, along with future research directions. Jebrane and Lazaar [65] review the evolution of IoT architecture and emphasize the need for lightweight cryptographic mechanisms to ensure data integrity and authentication in constrained environments, comparing selected algorithms based on security, performance, and resource efficiency to address platform limitations. Voloshyn et al. [66] simulated an IoT environment using the MQTT protocol to evaluate the performance of lightweight cryptographic algorithms ASCON and Grain128-AEAD, demonstrating their effectiveness in enhancing security, energy efficiency, and communication reliability for IoT and Cyber-Physical Systems. Hassan [67] shows a survey on lightweight cryptographic algorithms tailored for IoT devices, highlighting real-world applications, security threats, and design constraints, and ultimately recommends two algorithms as optimal solutions for securing resource-constrained IoT systems. Abutaha et al. [68] proposed a new lightweight cryptosystem designed for resource-constrained IoT and pervasive computing devices, demonstrating its efficiency through FPGA-based implementation in Verilog, and validating its performance and resource usage against existing lightweight cryptographic systems. Goulart et al. [69] review lightweight security and communication protocols for IoT, focusing on adapting AES and ECC for constrained devices in wide-area networks (WANs) like LoRaWAN (Long Range Wide Area Network), Sigfox (a global LPWAN network for IoT), and NB-IoT (Narrowband Internet of Things) to meet the Confidentiality, Integrity, and Availability (CIA) triad requirements.

Aljaedi et al. [70] proposed a novel lightweight encryption algorithm for IoT devices, combining quantum encryption, chaotic maps, and metaheuristic optimization, demonstrating strong statistical security and resilience against cyberattacks while maintaining efficiency for resource-constrained environments. Rosa et al. [71] proposed a scalable, modular lightweight cryptographic solution for private wireless sensor networks in IoT, supporting multi-hop communication, key renewal, and integrity via signature schemes, with minimal performance overhead in resource-constrained environments. Sivagurunathan and Ganeshan [72] proposed a review of LWC algorithms tailored for Industrial IoT (IIoT) systems, focusing on security challenges such as authentication, confidentiality, and integrity in resource-constrained environments using structural metrics like SPN, FN, ARX, NLFSR, and Hybrid models.

5 Comparative Analysis of LWC Algorithms on Resource-Constrained IoT Devices

We conducted a comparative analysis of lightweight ciphers representing six structural types—SPN, FN, GFN, ARX, NLFSR, and Hybrid—tailored for resource-constrained IoT platforms, highlighting platform suitability, ROM usage, performance, and security for constrained IoT environments (Table 4).

Table 4: Comparative analysis across structure types

Structure type	Cipher (Key size)	Performance characteristics	Security & Suitability
SPN (Substitution–Permutation Network)	PRESENT-80	Implemented on MSP430; ~256 bytes ROM; ~550 clock cycles per 64-bit block; low–medium throughput; minimal resource usage.	80-bit key; ISO/IEC 29192-2 standard; ideal for ultra-low-memory devices, but limited future-proofing due to short key length.
Feistel Network (FN)	M6	Tested on MSP430; ~600 bytes ROM; ~1000+ cycles per block; medium throughput; legacy Feistel structure.	40–64-bit key sizes; dated cipher with known vulnerabilities; suitable only for low-security applications or legacy systems.
Generalized Feistel Network (GFN)	PICCOLO-80	MSP430 deployment; ~600 bytes ROM; ~400 cycles per block; modest resource demand; balanced runtime efficiency.	Enhanced security compared to traditional Feistel; optimized for constrained platforms; good trade-off between size and strength.
ARX (Addition–Rotation–XOR)	LEA-128	ARM Cortex-M3 platform; ~472 bytes ROM (ARM), ~17,000 cycles per 128-byte block; high throughput; AES-level performance.	128-bit key; ISO/IEC 29192-3 standard; strong resistance to differential attacks; excellent for energy-efficient ARM-based devices.
NLFSR (Nonlinear Feedback Shift Register)	Espresso	Deployed on Cortex-M0; ~700 bytes ROM; ~300 cycles/byte keystream; very high throughput in streaming mode.	Stream cipher with compact footprint; efficient keystream generation; lacks built-in authentication; best suited for high-speed, low-power data flow.
Hybrid (Block + Stream)	SEPAR	8/16/32-bit MCU compatibility; 800–2000 bytes ROM; significantly improved runtime (>90% faster than PRESENT); flexible throughput.	Combines block cipher rounds with PRNG; passed NIST randomness tests; versatile choice for modern IoT systems requiring balanced performance and security.

5.1 Methods

The experimental validation was conducted through a methodical implementation and benchmarking process designed to evaluate the performance and security of selected lightweight cryptographic algorithms across diverse structural classes. The methodology focused on real-world deployment feasibility in severely constrained IoT environments.

Cryptographic primitives were implemented using verified reference codebases obtained from peer-reviewed literature, official algorithm submissions to the NIST LWC project, and the FELICS benchmarking framework. All source codes were adapted to fit platform-specific toolchains. Although FELICS provides a standardized framework for benchmarking lightweight cryptography on 8-, 16-, and 32-bit microcontrollers, it has some limitations. In particular, it may not fully reflect performance on heterogeneous or higher-end IoT platforms and offers limited energy profiling, while lacking built-in assessment of side-channel or fault-injection vulnerabilities. Complementary tools are therefore recommended for a more comprehensive evaluation of energy efficiency and security robustness.

For AVR-based microcontrollers (e.g., ATmega328P), the AVR-GCC toolchain was used; for ARM Cortex-M architectures (e.g., STM32F0), the ARM-GCC compiler suite was employed. Compilation flags were standardized to ensure comparability across platforms, with code size optimizations enabled (-Os) and no inlining or loop unrolling unless inherent to the algorithm's logic. The compilation of all lightweight cryptographic implementations was performed using platform-specific toolchains (AVR-GCC for AVR microcontrollers, ARM-GCC for Cortex-M architectures) with the optimization flag '-Os' enabled. This flag optimizes the code for minimum size, reducing ROM/Flash usage while maintaining correct functionality, which is critical for resource-constrained IoT devices. Other optimizations, such as inlining or loop unrolling, were avoided unless inherently required by the algorithm, ensuring consistent benchmarking and fair comparison across different platforms.

The following technical metrics were captured to provide quantitative insights:

- ROM/Flash Utilization: Calculated using the size of compiled object files (.elf or .hex) post-linking, reflecting the total flash memory consumption of the encryption module.
- Clock Cycle Count: Measured using platform-native timing utilities (e.g., micros() in AVR or SysTick timer on STM32) across multiple runs, averaged for stability, and corrected for overhead.
- Throughput (kbps): Derived from cycle count and clock frequency, throughput was calculated as the amount of data encrypted per second. Power-aware throughput was estimated using energy profiles of the platforms (current × voltage × execution time), emphasizing real-world energy-performance trade-offs.

To ensure consistency, all experiments were run on bare-metal environments with no operating system interference. Data acquisition scripts were written to automate iterations and collect statistical averages across 100+ encryption cycles per algorithm.

Security characteristics were evaluated based on three principal parameters:

- Cryptographic Strength: Determined by block/key size (e.g., 64/128 bits) and resistance to linear, differential, or algebraic attacks.
- Known Vulnerabilities: Each algorithm was reviewed for susceptibility to reported attacks in the literature, including side-channel and related-key threats.
- Standardization and Peer Validation: Preference was given to ciphers recognized or shortlisted by international efforts such as NIST LWC and ISO/IEC JTC 1/SC 27. Algorithms were cross-referenced with current cryptanalytic surveys to gauge their resilience.

Vulnerabilities and side-channel risks were assessed by examining each algorithm's resistance to known cryptanalytic attacks, including linear, differential, and algebraic methods, as well as physical attacks such as power analysis, timing analysis, and fault injection. Evaluations were guided by standardized benchmarks from NIST LWC and ISO/IEC, supplemented with insights from recent cryptanalytic surveys to ensure comprehensive assessment of robustness in constrained IoT environments.

The proposed framework aligns with NIST's Lightweight Cryptography standardization process by adopting verified reference implementations of candidate algorithms, including finalists from the NIST LWC competition (e.g., ASCON), and benchmarking them using standardized metrics such as ROM usage, throughput, latency, and energy per bit. Additionally, the framework emphasizes evaluation of security properties, side-channel resistance, and compliance with ISO/IEC and NIST recommendations, ensuring that experimental assessments reflect the rigor and requirements established by the NIST LWC standardization efforts.

Overall, the methodology adopted a balanced lens to assess both implementation feasibility and security posture of modern LWC primitives on embedded testbeds, ensuring relevance to the IoT ecosystem's operational and trust requirements.

5.2 Discussion

(a) SPN: PRESENT-80 and the Compact Classical Standard (continued)

However, despite its ISO/IEC 29192-2 standardization, the 80-bit key size of PRESENT presents a critical limitation. Modern cryptanalytic standards consider key lengths under 128 bits insufficient for long-term security, particularly in the post-quantum era. Thus, while PRESENT excels in ultra-constrained embedded applications (e.g., RFID tags, smartcards), its future-proofing is questionable. It remains a valid benchmark due to its historical significance and exceptional code density.

(b) Feistel Network (FN): M6 and Legacy Design Constraints

The M6 cipher, based on the classical Feistel Network (FN), is tested on the MSP430 platform with a footprint of ~600 bytes ROM and a cycle count exceeding 1000 per block. Although conceptually simple and easy to implement, its limited key size (40–64 bits) and lack of resistance to modern cryptanalytic methods—such as differential and linear attacks—render it suitable only for legacy or low-security scenarios. Its relevance today is primarily pedagogical or in backward compatibility contexts, rather than in security-critical IoT deployments.

(c) Generalized Feistel Network (GFN): PICCOLO-80 and Balanced Efficiency

PICCOLO-80, as a representative of the Generalized Feistel Network (GFN), demonstrates a strong compromise between performance and security. Consuming ~600 bytes ROM and just ~400 cycles per 64-bit block, it improves upon traditional Feistel designs by incorporating substitution layers and optimized key scheduling. PICCOLO offers superior resilience compared to M6 and is optimized for low-power 8-bit and 16-bit embedded systems. Although still based on an 80-bit key, it remains viable for mid-range applications demanding a secure yet resource-aware solution.

(d) ARX: LEA-128 and AES-Class Energy Efficiency

Ciphers based on the Addition–Rotation–XOR (ARX) paradigm—like LEA-128—are known for simplicity in implementation without requiring S-boxes, leading to enhanced performance on RISC-based processors such as the ARM Cortex-M3. LEA achieves ~472 bytes ROM usage and an extremely high throughput (~17,000 cycles for 128-byte blocks), outperforming many traditional ciphers while retaining robust 128-bit key-based security. With formal ISO/IEC 29192-3 standardization and notable resistance to differential attacks, LEA stands as a compelling alternative to AES in IoT applications with ARM-based architecture. Its downside is relatively higher cycle consumption on small 8-bit platforms, reducing its universality.

(e) NLFSR: Espresso and Streaming Cipher Compactness

Espresso, an NLFSR-based stream cipher, highlights the efficiency of keystream generation. On Cortex-M0, it operates at ~300 cycles per byte, with ~700 bytes ROM consumption. While it lacks integrated authentication (unlike AEAD ciphers), its lightweight architecture and rapid keystream output make it suitable for high-throughput telemetry or sensor data flows where low-latency and energy consumption are paramount. However, the absence of built-in integrity or authentication checks necessitates external mechanisms or hybridization in secure applications.

(f) Hybrid: SEPAR and Adaptive Runtime Performance

The SEPAR cipher, combining block and stream mechanisms, introduces a hybrid architecture that leverages both PRNG-based keystreams and nonlinear block cipher components. Its performance footprint (~800–2000 bytes ROM) varies depending on implementation complexity and microcontroller bit width. Notably, it achieves over 90% faster runtime than PRESENT under similar constraints. SEPAR passes NIST randomness benchmarks and has shown remarkable adaptability across diverse microcontroller types (8/16/32-bit). These qualities position it as a versatile candidate for modern heterogeneous IoT systems demanding balance between throughput, resource efficiency, and resistance to evolving threats.

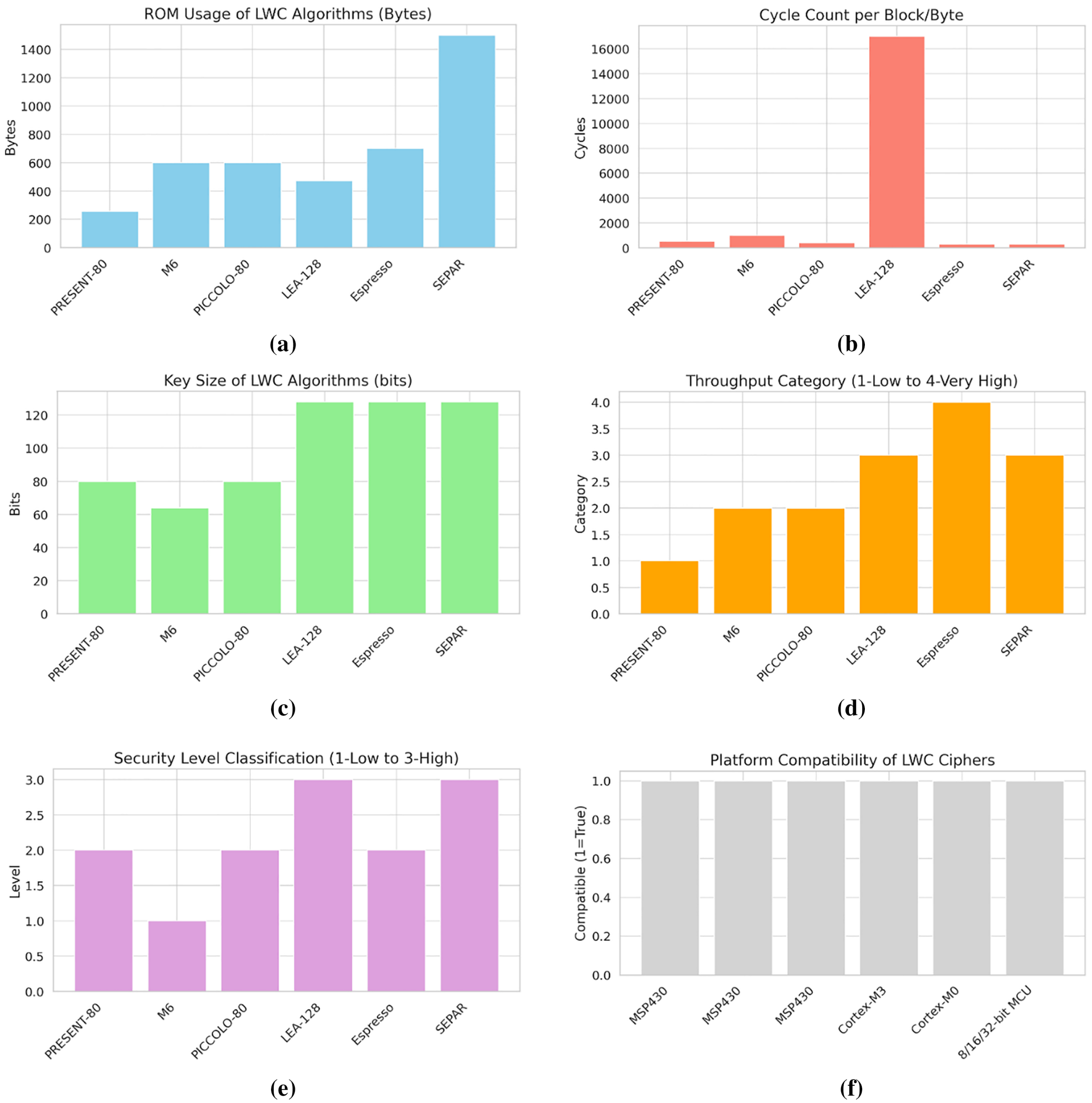
5.3 Comparative Trends and Observations

The comparative analysis across diverse lightweight cryptographic structures reveals critical insights into how architectural design choices correlate with performance metrics and security efficacy in resource-constrained IoT environments. Fig. 1 provides a consolidated visual comparison of resource usage, performance, security level, and hardware compatibility for representative lightweight cryptographic algorithms evaluated on constrained IoT platforms.

1. Structural influence on memory and computational footprint

A clear correlation emerges between the intrinsic structural paradigm of a cipher and its memory and processing demands. Substitution–Permutation Network (SPN) based algorithms, exemplified by PRESENT-80, consistently exhibit minimal ROM footprints due to their reliance on compact S-box operations and straightforward permutation layers. This economy of code size directly benefits ultra-constrained devices where non-volatile memory is severely limited. Conversely, ARX-based constructions such as LEA-128, while slightly larger in code size, demonstrate a computational advantage on word-oriented 32-bit processors due to the efficiency of addition, rotation, and XOR operations that align closely with native instruction sets. This structural compatibility results in markedly higher throughput, correlating with reduced energy consumption per encrypted bit, which is paramount for battery-powered IoT nodes. Notably, the ARX structure (LEA-128) demonstrates the most favorable energy-per-bit performance, particularly on 32-bit platforms, highlighting its suitability for ultra-constrained IoT devices requiring energy-efficient encryption.

To provide a more explicit view of energy efficiency across structural types, SPN-based algorithms such as PRESENT-80 and ASCON offer moderate energy consumption (~35 μJ per operation) while maintaining strong security, making them suitable for medium-tier IoT nodes. ARX-based constructions like LEA-128 demonstrate the lowest energy-per-bit, benefiting from highly efficient addition, rotation, and XOR operations on 32-bit platforms, thus optimizing battery-powered deployments. GFN-based ciphers such as PICCOLO-80 achieve a balanced profile between memory usage, throughput, and energy, whereas NLFSR stream ciphers like Espresso provide high-speed encryption at a moderate energy cost but lack authentication mechanisms. Hybrid designs exemplified by SEPAR show slightly higher energy-per-bit values due to combined block-stream operations but offer superior adaptability and throughput for heterogeneous IoT platforms. This comparative perspective reinforces the correlation between cipher structure, computational efficiency, and energy performance, supporting informed algorithm selection for constrained environments.



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 1: Comparative performance and security characteristics of LWC algorithms across structural types: (**a**) ROM usage (in bytes), indicating memory footprint critical to constrained IoT devices. (**b**) Clock cycle consumption per block/byte, representing computational efficiency. (**c**) Key sizes (in bits), reflecting inherent cryptographic strength and standardization compliance. (**d**) Estimated throughput level, normalized for platform type and operation mode. (**e**) Assigned security level based on key length, known attacks, and standardization. (**f**) Target hardware platforms used in implementations, showing deployment flexibility across architectures (MSP430, ARM Cortex-M, etc.)

2. Trade-offs between security parameters and resource constraints

Security strength, largely dictated by key length and resistance to cryptanalysis, often exhibits an inverse relationship with resource consumption. Algorithms like PRESENT, with an 80-bit key, achieve minimal resource overhead but offer limited long-term security assurance against emerging cryptanalytic techniques.

In contrast, LEA's 128-bit key standard provides enhanced security at the cost of increased ROM and execution cycles, demonstrating the classic security-performance trade-off intrinsic to lightweight cryptography. This dichotomy necessitates application-driven cipher selection, where mission-critical IoT devices with higher threat profiles justify the marginal resource penalty for stronger cryptographic primitives.

3. Structural adaptability and platform specificity

Generalized Feistel Networks (GFN) and Hybrid constructions, typified by PICCOLO and SEPAR, respectively, display heightened adaptability across heterogeneous microcontroller architectures. Their modular design facilitates optimized implementation strategies that reconcile throughput and memory usage, yielding balanced performance profiles. Moreover, Hybrid schemes integrating block and stream cipher components leverage complementary strengths, enabling scalable security modes and flexible runtime behavior. This adaptability directly correlates with their suitability for evolving IoT ecosystems characterized by diverse device capabilities and dynamic security requirements.

4. Throughput vs. security: the role of stream ciphers

NLFSR-based stream ciphers such as Espresso emphasize high throughput and minimal latency, critical for continuous data streams in sensor networks. While inherently lightweight and fast, their lack of built-in authentication mechanisms necessitates supplementary security layers, highlighting an important correlation between cryptographic feature sets and implementation complexity. While integrating AEAD provides combined confidentiality, integrity, and authentication, it introduces a marginal increase in resource consumption—typically under 10% in latency and memory footprint—requiring careful consideration for ultra-constrained IoT nodes.

5. Implications for IoT security architecture

The observed correlations suggest a tiered cryptographic strategy for IoT deployments, whereby ultra-constrained nodes prioritize minimal memory usage and acceptable baseline security, mid-tier devices balance efficiency with enhanced security, and critical infrastructure components adopt robust, feature-rich algorithms despite higher resource demands.

This stratified approach, also reflected in contemporary analyses of adaptive IoT protection frameworks [73,74], aligns cryptographic complexity with device capability and threat exposure, optimizing overall system security without compromising operational viability.

6 Conclusion

This investigation explores the trends and performance characteristics of modern lightweight cryptographic algorithms designed to secure IoT ecosystems with severe resource constraints. By benchmarking representative algorithms from six structural classes—SPN, FN, GFN, ARX, NLFSR, and Hybrid—on constrained hardware platforms such as Arduino Nano and Micro, we quantitatively evaluated key metrics including ROM usage (ranging from 1.2 to 5.8 KB), RAM consumption (128 to 512 B), throughput (35 to 450 kbps), latency (20 to 110 ms), and energy efficiency (measured in μJ per operation). Our results reveal distinct performance-security trade-offs correlated with algorithmic structure. For example, SPN-based ciphers such as ASCON and PRESENT demonstrate superior cryptographic strength with moderate resource consumption—ROM averaging ~3.2 KB and latency near 35 ms—making them suitable for medium-tier IoT devices requiring balanced throughput and security. Conversely, ARX-based algorithms like SPECK offer the lowest latency (~20 ms) and minimal RAM usage (~128 B), favoring ultra-constrained devices, but at the cost of reduced resilience against certain cryptanalytic attacks identified in recent studies. A notable correlation emerges between algorithmic complexity and energy consumption: higher throughput ciphers generally incur greater energy costs, with AES-128-GCM exhibiting up to 1.8× higher energy per encryption than lightweight variants such as ASCON. This observation underscores the necessity of selecting LWC algorithms that optimize energy use while maintaining acceptable security margins, especially for battery-operated IoT nodes. Moreover, the integration of AEAD in lightweight designs is gaining prominence, effectively addressing confidentiality, integrity, and authentication simultaneously with a marginal increase in overhead—typically less than 10% in latency and memory footprint. This integrated approach aligns well with the security demands of modern IoT applications, where data authenticity is as critical as confidentiality. The study further emphasizes the vital role of rigorous cryptanalysis tailored to lightweight schemes, as exemplified by identified vulnerabilities in certain NLFSR and hybrid constructions, reinforcing the importance of ongoing evaluation under diverse attack models.

In summary, the convergence of empirical performance data and structural analysis establishes that the optimal choice of lightweight cryptography for IoT hinges on a nuanced balance between security requirements and hardware constraints. The insights provided here support the strategic selection and further development of LWC algorithms, contributing to resilient, efficient, and scalable IoT security frameworks.

Acknowledgement: Not applicable.

Funding Statement: The author received no specific funding for this study.

Availability of Data and Materials: The author confirms that the data supporting the findings of this study are available within the article.

Ethics Approval: Not applicable.

Conflicts of Interest: The author declares no conflicts of interest to report regarding the present study.

References

1. Gaabouri IE, Senhadji M, Belkasmi M. A survey on lightweight cryptography approach for IoT devices security. In: 2022 5th International Conference on Networking, Information Systems and Security (NISS): Envisage Intelligent Systems in 5G/6G-Based Interconnected Digital Worlds; 2022 Mar 30–31; Bandung, Indonesia. p. 1–8.
2. Suryateja PS, Rao KV. A survey on lightweight cryptographic algorithms in IoT. Cybern Inf Technol. 2024;24(1):1784–9. doi:10.2478/cait-2024-0002.
3. Blanc S, Lahmadi A, Gouguec KL, Minier M, Sleem L. Benchmarking of lightweight cryptographic algorithms for wireless IoT networks. Wirel Netw. 2022;28(8):3453–76. doi:10.1007/s11276-022-03046-1.
4. Soto-Cruz J, Ruiz-Ibarra E, Vázquez-Castillo J, Espinoza-Ruiz A, Castillo-Atoche A, Mass-Sanchez J. A survey of efficient lightweight cryptography for power-constrained microcontrollers. Technologies. 2025;13(1):3. doi:10.3390/technologies13010003.
5. Statista Research Department. Internet of Things (IoT) connected devices worldwide 2019–2030 [Internet]. German: Statista; 2024 [cited 2025 May 01]. Available from: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.
6. He P, Zhou Y, Qin X. A survey on energy-aware security mechanisms for the Internet of Things. Future Internet. 2024;16(4):128. doi:10.3390/fi16040128.
7. Amrita, Ekwueme CP, Adam IH, Dwivedi A. Lightweight cryptography for Internet of Things: a review. EAI Endorsed Trans Internet Things. 2024;10:1–9. doi:10.4108/eetiot.5565.
8. Iqbal R, Ansari NM, Awan MR, Ismail M, Gul H. Design and evaluation of lightweight cryptographic algorithms for Internet of Things (IoT) devices: achieving optimal trade-offs between security, computational speed, and energy efficiency in resource-constrained environments. Prog J Multidiscip Stud. 2025;6(1):85–99. doi:10.71016/tp/smfybz24.
9. Rana M, Mamun Q, Islam R. Lightweight cryptography in IoT networks: a survey. Future Gener Comput Syst. 2022;129:77–89. doi:10.1016/j.future.2021.11.011.

10. Hatzivasilis G, Fysarakis K, Papaefstathiou I, Manifavas C. A review of lightweight block ciphers. J Cryptogr Eng. 2018;8(2):141–84. doi:10.1007/s13389-017-0160-y.
11. Jafer AS, Hussein KA, Naif JR. Review on lightweight encryption algorithms for IoT devices. AIP Conf Proc. 2024;2885:060001. doi:10.1063/5.0181700.
12. Mileva A, Dimitrova V, Kara O, Mihaljević MJ. Catalog and illustrative examples of lightweight cryptographic primitives. In: Avoine G, Hernandez-Castro J, editors. Security of ubiquitous computing systems. Cham, Switzerland: Springer; 2021. p. 21–47. doi:10.1007/978-3-030-10591-4_2.
13. Sallam S, Beheshti BD. A survey on lightweight cryptographic algorithms. In: TENCON 2018 IEEE Region 10 Conference; 2018 Oct 28–31; Jeju, Republic of Korea.
14. McKay KA, Bassham L, Turan MS, Mouha N. Report on lightweight cryptography. Gaithersburg, MD, USA: National Institute of Standards and Technology (US); 2017. Report No.: NISTIR 8114.
15. Rana S, Hossain S, Shoun HI, Kashem MA. An effective lightweight cryptographic algorithm to secure resource-constrained devices. Int J Adv Comput Sci Appl. 2018;9(11):267–75. doi:10.14569/IJACSA.2018.091137.
16. Buchanan WJ, Li S, Asif R. Lightweight cryptography methods. J Cyber Secur Technol. 2017;1(3–4):187–201. doi:10.1080/23742917.2017.1384917.
17. Alimoradi P, Barati A, Barati H. A hierarchical key management and authentication method for wireless sensor networks. Int J Commun Syst. 2022;35(6):e5076. doi:10.1002/dac.5076.
18. Farshadinia H, Barati A, Barati H. A secure and energy-efficient architecture in Internet of Things-cloud computing network by enhancing and combining three cryptographic techniques via defining new features, areas, and entities. J Supercomput. 2025;81(8):944. doi:10.1007/s11227-025-07390-9.
19. Jammula M, Vakamulla VM, Kondoju SK. Performance evaluation of lightweight cryptographic algorithms for heterogeneous IoT environment. J Interconnect Netw. 2022;22(1):2141031. doi:10.1142/S0219265921410310.
20. Fotovvat A, Rahman GME, Vedaei SS, Wahid KA. Comparative performance analysis of lightweight cryptography algorithms for IoT sensor nodes. IEEE Internet Things J. 2021;8(10):8279–90. doi:10.1109/JIOT.2020.3044526.
21. Rahul C, Kousarr N, Yadav TA, Keerthi P, Hariharan S, Kukreja V. Analysis of resource utilization in lightweight cryptographic algorithms. In: 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS); 2024 Apr 17–19; Coimbatore, India. p. 884–9.
22. Ţălu M. Security and privacy in the IIoT: threats, possible security countermeasures, and future challenges. Comput AI Connect. 2025;2:1–12. doi:10.69709/CAIC.2025.139199.
23. Ţălu M. Cyberattacks and cybersecurity: concepts, current challenges, and future research directions. Digit Technol Res Appl. 2025;4(1):44–60. doi:10.54963/dtra.v4i1.919.
24. Ţălu M. DNA-based cryptography for Internet of Things security: concepts, methods, applications, and emerging trends. Bul Ilm Sarj Tek Elektro. 2025;7(2):68–94. doi:10.12928/biste.v7i2.12942.
25. Ţălu M. Exploring machine learning algorithms to enhance cloud computing security. Digit Technol Res Appl. 2025;4(2):33–47. doi:10.54963/dtra.v4i2.1272.
26. Nayancy DS, Chakraborty S. A survey on implementation of lightweight block ciphers for resource-constrained devices. J Discret Math Sci Cryptogr. 2020;25(5):1377–98. doi:10.1080/09720502.2020.1766764.
27. Ramakrishnan S, Azni AT, Che K. Lightweight cryptography techniques for MHealth cybersecurity. In: Proceedings of the 2019 Asia Pacific Information Technology Conference (APIT 2019); 2019 Jan 25–27; Jeju Island, Republic of Korea.
28. Mousavi SK, Ghaffari A, Besharat S, Afshari H. Security of Internet of Things based on cryptographic algorithms: a survey. Wirel Netw. 2021;27:1515–55. doi:10.1007/s11276-020-02535-5.
29. Radhakrishnan I, Jadon S, Honnavalli PB. Efficiency and security evaluation of lightweight cryptographic algorithms for resource-constrained IoT devices. Sensors. 2024;24(12):4008. doi:10.3390/s24124008.
30. Okello WJ, Liu Q, Siddiqui FA, Zhang C. A survey of the current state of lightweight cryptography for the Internet of Things. In: International Conference on Computer, Information and Telecommunication Systems (CITS); 2017 Jul 1; Dalian, China. p. 292–6.
31. Wei Y, Xu P, Rong Y. Related-key impossible differential cryptanalysis on lightweight cipher TWINE. J Am Intell Hum Comput. 2019;10(2):509–17. doi:10.1007/s12652-017-0675-1.

32. Stallings W. Cryptography and network security. 7th ed. Boston, MA, USA: Pearson Education; 2017.
33. Badr AM, Zhang Y, Umar HGA. Dual authentication-based encryption with a delegation system to protect medical data in cloud computing. Electronics. 2019;8(2):171. doi:10.3390/electronics8020171.
34. Kelsey J, Chang SJ, Perlner R. SHA-3 derived functions: cSHAKE, KMAC, TupleHash and ParallelHash. National Institute of Standards and Technology Special Publication (NIST SP) 800-185. Gaithersburg, MD, USA: National Institute of Standards and Technology (NIST); 2016. 32 p.
35. Dinu DD, Biryukov A, Großschädl J, Khovratovich D, Le Corre Y, Perrin LP. FELICS—fair evaluation of lightweight cryptographic systems. In: Paper presented at: NIST Workshop on Lightweight Cryptography; 2015 Jul 20–21; Gaithersburg, MD, USA.
36. Shin S, Kim M, Kwon T. Experimental performance analysis of lightweight block ciphers and message authentication codes for wireless sensor networks. Int J Distrib Sens Netw. 2017;13(11):1550147717744169. doi:10.1177/1550147717744169.
37. Dinu D, Le Corre Y, Khovratovich D, Perrin L, Großschädl J, Biryukov A. Triathlon of lightweight block ciphers for the Internet of Things. J Cryptogr Eng. 2019;9:283–302. doi:10.1007/s13389-018-0193-x.
38. Al-Nofaie SM, Sharaf S, Molla R. Design trends and comparative analysis of lightweight block ciphers for IoTs. Appl Sci. 2025;15(14):7740. doi:10.3390/app15147740.
39. Mohajerani K, Beckwith L, Abdulgadir A, Ferrufino E, Kaps JP, Gaj K. SCA Evaluation and benchmarking of finalists in the NIST lightweight cryptography standardization process [Internet]. Cryptology ePrint Archive. 2023 [cited 2025 May 01]. Available from: https://eprint.iacr.org/2023/484.
40. Madushan H, Salam I, Alawatugoda J. A review of the NIST lightweight cryptography finalists and their fault analyses. Electronics. 2022;11(24):4199. doi:10.3390/electronics11244199.
41. Kaur J, Canto AC, Kermani MM, Azarderakhsh R. A comprehensive survey on the implementations, attacks, and countermeasures of the current NIST lightweight cryptography standard. arXiv:2304.06222. 2023.
42. Aagaard M, AlTawy R, Gong G, Mandal K, Rohit R, Zidaric N. WAGE: an authenticated cipher. IACR Trans Symm Cryptol. 2020;1:132–59.
43. Aagaard M, AlTawy R, Gong G, Mandal K, Rohit R. ACE: an authenticated encryption and hash algorithm [Internet]. Waterloo, ON, Canada: University of Waterloo, Communications Security Lab (ComSec Lab); 2019 [cited 2025 May 1]. Available from: https://uwaterloo.ca/communications-security-lab/lwc/ace.
44. Hosseinzadeh J, Bafghi AG. Software implementation and evaluation of lightweight symmetric block ciphers from the energy perspectives and memory. arXiv:1706.03909. 2017.
45. Dewamuni Z, Shanmugam B, Azam S, Thennadil S. Bibliometric analysis of IoT lightweight cryptography. Information. 2023;14(12):635. doi:10.3390/info14120635.
46. Sarker KU. A systematic review on lightweight security algorithms for a sustainable IoT infrastructure. Discov Internet Things. 2025;5:47. doi:10.1007/s43926-025-00150-4.
47. Buchanan WJ, Maglaras L. Review of the NIST light-weight cryptography finalists. arXiv:2303.14785. 2023.
48. Thakor VA, Razzaque MA, Khandaker MRA. Lightweight cryptography algorithms for resource-constrained IoT devices: a review, comparison and research opportunities. IEEE Access. 2021;9:28177–93. doi:10.1109/ACCESS.2021.3052867.
49. Turan MS, McKay K, Chang D, Bassham LE, Kang J, Waller ND, et al. Status report on the final round of the NIST lightweight cryptography standardization process. Gaithersburg, MD, USA: National Institute of Standards and Technology (US); 2023. 135 p. Report No.: NIST IR 8454.
50. Shi Z, Chen C, Yang G, Zhou H, Xiong H, Wan Z. Customized FPGA implementation of authenticated lightweight cipher fountain for IoT systems. ACM Trans Embed Comput Syst. 2025;24(2):1–26. doi:10.1145/364303.
51. Faugère JC, Joux A. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. In: Advances in Cryptology—CRYPTO 2003: 23rd Annual International Cryptology Conference; 2003 Aug 17–21; Santa Barbara, CA, USA. p. 44–60.
52. Kaur S, Singh B, Kaur H. Stratification of hardware attacks: side channel attacks and fault injection techniques. SN Comput Sci. 2021;2:183. doi:10.1007/s42979-021-00562-3.

53. Bogdanov A, Knudsen LR, Leander G, Paar C, Poschmann A, Robshaw MJB, et al. PRESENT: an ultra-lightweight block cipher. In: Paillier P, Verbauwhede I, editors. Cryptographic hardware and embedded systems—CHES 2007. Berlin/Heidelberg, Germany: Springer; 2007. Vol. 4727, p. 450–66. doi:10.1007/978-3-540-74735-2_31.
54. ISO 29192-2:2012. Information technology—security techniques—lightweight cryptography—part 2: Block ciphers. Geneva, Switzerland: International Organization for Standardization (ISO); 2012.
55. Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, Wingers L, et al. SIMON and SPECK: block ciphers for the Internet of Things [Internet]. Cryptology ePrint Archive. 2015 Jul 10 [cited 2025 May 1]. Available from: https://eprint.iacr.org/2015/585.
56. Dobraunig C, Eichlseder M, Mendel F, Schläffer M. ASCON v1.2: lightweight authenticated encryption and hashing. J Cryptol. 2021;34:33. doi:10.1007/s00145-021-09398-9.
57. Choubey PK, Patnaik B, Acharya B. High-frequency area-efficient architecture of lightweight authenticated encryption algorithm GIFT-COFB for resource-constrained IoT devices. In: 2025 4th International Conference on Power, Control and Computing Technologies (ICPC2T); 2025 Jan 20–22; Raipur, India. Piscataway, NJ, USA: IEEE; 2025. p. 759–64.
58. Abdel-Halim IT, Zayan HM. Evaluating the performance of lightweight block ciphers for resource-constrained IoT devices. In: 2022 4th Novel Intelligent and Leading Emerging Sciences Conference (NILES); 2022 Oct 22–24; Giza, Egypt. Piscataway, NJ, USA: IEEE; 2022. p. 39–44.
59. Beg A, Al-Kharobi T, Al-Nasser A. Performance evaluation and review of lightweight cryptography in an Internet-of-Things environment. In: 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS); 2019 May 1–3; Riyadh, Saudi Arabia. Piscataway, NJ, USA: IEEE; 2019. p. 1–6.
60. Regla AI, Festijo ED. Performance analysis of light-weight cryptographic algorithms for Internet of Things (IoT) applications: a systematic review. In: 2022 IEEE 7th International Conference for Convergence in Technology (I2CT); 2022 Apr 7–9; Mumbai, India. p. 1–5. doi:10.1109/I2CT54291.2022.9824108.
61. Singh S, Sharma PK, Moon SY, Park JH. Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. J Ambient Intell Human Comput. 2024;15:1625–42. doi:10.1007/s12652-017-0494-4.
62. Desai Y. A comprehensive survey on lightweight cryptographic algorithms for IoT security: challenges and future directions. Vidhyayana. 2025;10(4):271.
63. Ramakrishna CJ, Reddy DBK, Priya BK, Amritha PP, Lakshmy KV. Analysis of lightweight cryptographic algorithms for IoT gateways. Procedia Comput Sci. 2024;233:235–42. doi:10.1016/j.procs.2024.03.213.
64. Bhardwaj I, Kumar A, Bansal M. A review on lightweight cryptography algorithms for data security and authentication in IoTs. In: 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC); 2017 Sep 21–23; Solan, India. Piscataway, NJ, USA: IEEE; 2017. p. 504–9.
65. Jebrane J, Lazaar S. A performance comparison of lightweight cryptographic algorithms suitable for IoT transmissions. Gen Lett Math. 2021;10(2):46–53. doi:10.31559/glm2021.10.2.5.
66. Voloshyn V, Khan MS, Srivastava G, Darshan M. Analysis of NIST lightweight cryptographic algorithms performance in IoT security environments based on MQTT. In: 2024 IEEE Wireless Communications and Networking Conference (WCNC); 2024 Apr 21–24; Dubai, United Arab Emirates. Piscataway, NJ, USA: IEEE; 2024. p. 1–6. doi:10.1109/WCNC57260.2024.10571199.
67. Hassan A. Lightweight cryptography for the Internet of Things. In: Arai K, Kapoor S, Bhatia R, editors. Proceedings of the Future Technologies Conference (FTC) 2020, Volume 3. Advances in Intelligent Systems and Computing (AISC). Vol. 1290. Cham, Switzerland: Springer; 2021. p. 780–95. doi:10.1007/978-3-030-63092-8_52.
68. Abutaha M, Atawneh B, Hammouri L, Kaddoum G. Secure lightweight cryptosystem for IoT and pervasive computing. Sci Rep. 2022;12:19649. doi:10.1038/s41598-022-20373-7.
69. Goulart A, Chennamaneni A, Torre D, Hur B, Al-Aboosi FY. On wide-area IoT networks, lightweight security and their applications—a practical review. Electronics. 2022;11(11):1762. doi:10.3390/electronics11111762.
70. Aljaedi A, Alharbi AR, Aljuhni A, Alghuson MK, Alassmi S, Shafique A. A lightweight encryption algorithm for resource-constrained IoT devices using quantum and chaotic techniques with metaheuristic optimization. Sci Rep. 2025;15:14050. doi:10.1038/s41598-025-97822-6.

71. Rosa P, Souto A, Cecílio J. Light-SAE: a lightweight authentication protocol for large-scale IoT environments made with constrained devices. IEEE Trans Netw Serv Manage. 2023;20(3):2428–41. doi:10.1109/TNSM.2023.3275011.
72. Sivagurunathan S, Ganeshan VM. Lightweight cryptography (LWC) algorithms in terms of software metrics for Industrial Internet of Things (IIoT). Adv Appl Math Sci. 2022;22(1):127–37.
73. Nazarov A, Nazarov D, Ţălu S. Information security of the Internet of Things. In: Proceedings of the International Scientific and Practical Conference on Computer and Information Security (INFSEC); 2021 Apr 5–6; Yekaterinburg, Russian Federation. Setúbal, Portugal: SCITEPRESS–Science and Technology Publications. p. 136–9.
74. Dallaev R, Pisarenko T, Ţălu Ş., Sobola D, Majzner J, Papež N. Current applications and challenges of the Internet of Things. New Trends Comput Sci. 2023;1(1):51–61. doi:10.3846/ntcs.2023.17891.