Creative Economy
2026, VOL. 10, NO. 2, 42-50
DOI: 10.47297/wspceWSP2516-251910.20261002

Lightweight Cryptographic Algorithms for Resource-Constrained IoT Environments: Applications, Challenges, and Prospects

Jiayi Yang

Beijing-Dublin International College, Beijing University of Technology, Beijing, 100124, China

ABSTRACT
The rapid growth of resource-constrained IoT devices makes traditional encryption algorithms difficult to deploy, thereby creating an urgent need for lightweight cryptography (LWC). Unlike existing surveys that primarily focus on performance ranking, this paper adopts a decision-oriented perspective to systematically review lightweight cryptographic algorithms suitable for IoT environments. A multi-dimensional evaluation framework is first established, separating algorithm-intrinsic properties (structure, block/key length, security margin) from implementation-related performance (hardware area, throughput, software speed, memory footprint). Based on this framework, a comparative analysis is conducted on representative lightweight block ciphers (PRESENT, SIMON, SPECK, LBLOCK, GIFT, RECTANGLE) and the NIST lightweight cryptography finalist authenticated encryption schemes (ASCON, GIFT-COFB, ISAP, TinyJAMBU, Xoodyak). Subsequently, scenario-specific adaptability is explored for three typical application scenarios: ultra-low-power sensor networks, latency-sensitive control links, and bandwidth-constrained communication. A preliminary decision model for algorithm selection is also proposed to guide practitioners. Finally, several key challenges are critically examined, including the trade-off between security and efficiency, physical attack threats, standardization and interoperability barriers, as well as future directions such as post-quantum cryptography and AI-assisted design. This paper aims to provide IoT security designers with theoretically grounded and practically feasible guidance for selecting lightweight cryptographic algorithms.

KEYWORDS
Lightweight cryptography; IoT; Performance evaluation; Scenario-based adaptation

1 Introduction

With the rapid development of IoT technology, a vast number of smart devices have been widely deployed in domains such as smart homes, industrial monitoring, smart cities, and healthcare. According to IoT Analytics, the number of connected IoT devices is projected to reach 39 billion by 2030. However, IoT devices are generally designed with low cost and low power consumption in mind, which results in extremely limited computing power, storage space, and energy supply (e.g., KB-level memory, MHz-level processors). This makes traditional cryptographic algorithms (such as AES, RSA, SHA-256) difficult or even impossible to deploy. This fundamental contradiction forces many systems to adopt "security compromise" strategies in practice, which in turn leads to numerous security incidents such as DDoS attacks and data breaches [1].

To address the above contradiction, the concept of "lightweight cryptography" has been proposed by both academia and industry. It aims to significantly reduce resource consumption (e.g., gate count, power consumption, memory usage) through optimized design while maintaining basic security. In recent years, international standardization bodies such as ISO/IEC and NIST have initiated lightweight cryptography standardization efforts, promoting the research and application of typical algorithms like PRESENT, SPECK, and ASCON.

However, most existing surveys focus on comparative performance analysis of algorithms while neglecting the diversity of application scenarios, thus failing to provide targeted selection advice [2]. Different scenarios have varying requirements for security, real-time performance, and energy consumption, meaning there is no single "optimal algorithm" that fits all. To this end, this survey constructs a decision-oriented research framework. It abandons the traditional ranking mindset, systematically reviews the classification system and technical paths of lightweight cryptography, establishes a multi-dimensional evaluation framework, and explores the decision logic of algorithm selection in conjunction with typical application scenarios. The goal is to provide IoT security designers with theoretically deep and practically meaningful guidance for algorithm selection.

2 Classification and Technical Approaches of Lightweight Cryptographic Algorithms

2.1 Cryptographic Fundamentals and Lightweight Design Philosophy

Lightweight cryptography mainly involves symmetric encryption, hash functions, and message authentication codes. Public-key cryptography is only used in a limited way for key exchange or signing due to its high computational overhead. Traditional algorithms such as AES and RSA were not designed with resource-constrained environments in

mind. Research shows that AES-256 can take up to 20 milliseconds to encrypt on resource-constrained nodes, with flash and RAM usage reaching as high as 20%, making direct deployment difficult. Lightweight cryptography reduces resource consumption significantly through optimized designs such as smaller state sizes, fewer rounds, or simple non-linear components, all while maintaining basic security. The core idea is to strike a balance between security and efficiency.

2.2 Classification of Typical Algorithms

2.2.1 Lightweight Block Ciphers

Block ciphers encrypt plaintext in fixed-length blocks and serve as fundamental building blocks for authenticated encryption and key management. They represent the most mature and widely used primitive in lightweight cryptography. Lightweight block ciphers typically adopt SPN (e.g., PRESENT), Feistel (e.g., SIMON), or ARX (e.g., SPECK) structures to optimize hardware area and power consumption, Some algorithms further incorporate design techniques such as bit-slicing and round unrolling to improve resource adaptability. In addition to the typical algorithms analyzed later, HIGHT is an improved Feistel structure that eliminates complex S-boxes and implements encryption solely through basic bitwise operations, making it suitable for extremely constrained devices like RFID tags and smart cards. Piccolo adopts a lightweight SPN structure with innovative layered S-boxes and a dynamic compact key schedule, offering flexible implementation across multiple architectures while maintaining security. The modular design of lightweight block ciphers allows them to be flexibly combined into various cryptographic schemes to meet the diverse security needs of IoT applications.

2.2.2 Lightweight Stream Ciphers

Stream ciphers generate a pseudo-random keystream from a key and XOR it with the plaintext bit by bit to achieve encryption. Their advantages lie in simple hardware implementation and low latency, making them suitable for applications with high real-time requirements. Typical algorithms include Trivium (with very high hardware efficiency and fast keystream generation) and the Grain series. However, traditional stream ciphers do not provide message integrity protection by themselves, requiring the addition of a MAC (e.g., Poly1305) or combination with a nonce mode, which partly offsets their structural simplicity. Therefore, in IoT communications that require authentication, stream ciphers are often deployed in the form of authenticated encryption (e.g., Trivium+MAC or TinyJambu) rather than as standalone primitives.

2.2.3 Lightweight Authenticated Encryption (AEAD)

Authenticated encryption with associated data (AEAD) provides both confidentiality and integrity protection simultaneously, and has become a mainstream requirement for IoT security. Compared to combined schemes of "block cipher and authentication mode", lightweight AEAD significantly reduces implementation overhead through integrated optimization. In the NIST lightweight cryptography standardization project, all ten finalist algorithms were AEAD schemes, reflecting the dominant position of authenticated encryption in IoT security. Apart from the standardized algorithms detailed later, typical examples include SPARKLE, which is a benchmark for software-optimized AEAD. It performs excellently on 32-bit MCU platforms and is a mainstream choice for software-side encryption in industrial IoT and connected vehicles. Elephant is designed for ultra-low-power IoT scenarios, employing a hybrid design of stream cipher and sponge structure. Its hardware implementation consumes extremely low power and supports lightweight key updates, making it suitable for long-term unattended sensor networks, medical implants, and other extremely resource-constrained scenarios.

2.3 NIST Lightweight Cryptography Standardization Process

To meet the security requirements of resource-constrained environments, NIST launched a lightweight cryptography standardization project in 2015, focusing on soliciting AEAD algorithms. After multiple rounds of evaluation, in February 2023 NIST announced the selection of ASCON as the winning algorithm from 56 candidates, and published the draft standard NIST SP 800-232 in 2024. ASCON implements authenticated encryption, hashing, and extensible output functions under a sponge structure, featuring strong security, high efficiency, and flexibility. Its standardization marks the transition of lightweight cryptography from academic research to industrial application.

3 Performance Evaluation and Scenario-Based Adaptability Analysis

3.1 Constructing a Multi-Dimensional Evaluation Framework

The evaluation of lightweight cryptographic algorithms should be carried out on two levels: algorithm-intrinsic properties and implementation performance. Intrinsic properties are determined by the algorithm design itself and do not vary with the implementation platform. They mainly include structure type (SPN/Feistel/ARX/sponge), block/key size (bits), number of rounds and security margin (attacked rounds/total rounds), design target (hardware-optimized/software-optimized/balanced design), and standardization status (ISO/IEC, NIST standard or candidate). Implementation performance depends on the specific platform and technology, and distinguishes between hardware and software implementations. Hardware implementation focuses on gate equivalent (GE) area, throughput, power consumption, and technology node. Software implementation focuses on encryption speed (cycles/byte), code size (byte), RAM usage (byte), and target platform (ARM/AVR/MSP). Only by combining these two levels can the applicability of an algorithm in a specific scenario be fully assessed.

3.2 Comparative Analysis of Similar Algorithms

3.2.1 Analysis of Lightweight Block Ciphers

Table 1 Inherent Characteristics of Representative Lightweight Block Ciphers

Name	Example	Structure	Block size	Key size	Rounds	Design objectives	Standardized state
PRESENT	PRESENT-80	SPN	64	80	31	Hardware optimization	ISO/IEC 29192-2
SIMON	SIMON64/128	Feistel	64	128	44	Hardware optimization	N/A
SPECK	SPECK64/128	ARX	64	128	27	software optimization	N/A
LBLOCK	LBLOCK-80	Feistel	64	80	32	balanced design	N/A
GIFT	GIFT-64-128	SPN	64	128	28	Hardware optimization	NIST candidate
RECTANLE	RECTANGLE-80	SPN	64	80	25	Slice design	N/A

Table 1 summarizes the intrinsic characteristics of seven representative algorithms. SPN structures (PRESENT, GIFT, RECTANGLE) and Feistel structures (SIMON, LBLOCK) account for a large proportion, while the ARX structure (SPECK) represents the direction of software optimization. PRESENT has been adopted in ISO/IEC 29192-2; GIFT is a NIST candidate; SIMON/SPECK were designed by NSA, targeting hardware and software respectively; LBLOCK is a balanced design; RECTANGLE uses bit-slice optimization.

Table 2 Hardware Performance of Lightweight Block Ciphers

Name	Technology used	Architecture	Area (GE)	Throughput (Kb/s @100kHz)	Power consumption (μW)	Reference
PRESENT-80	0.18μm	4-bit Serial	1075	11.7	1.4	[3]
SIMON64/128	0.13μm	4-bit Serial [Derived]	1000	16.7	-	[4]
SPECK64/128			1127	13.8	-	
LBLOCK-80	0.18μm	64-bit parallel	1320	200	-	[5]
GIFT-64-128	90nm	1-bit Serial	930	85	35.9	[6]
		4-bit Serial	1113	573	39.0	
RECTANGLE-80	0.13μm	Round-based	1599.5	246	0.74 [derived]	[7]

*Note: The data of 85 Kb/s@100kHz and 573 Kb/s@100kHz in this table are the results of unit conversion from the original literature. The conversion is based on 1 MHz = 1000 kHz, and the frequency scaling factor is 10 MHz / 100 kHz = 100.**The original literature value of 74.31 μW@10 MHz is linearly scaled by frequency to 0.7431 μW@100 kHz in this table. This value is cross-validated with the literature's energy efficiency of 3.0 pJ/bit and throughput of 246 Kbits/sec, yielding 0.738 μW. The discrepancy is attributed to static leakage, which is negligible for engineering purposes. In summary, the power consumption of this architecture at 100 kHz is approximately taken as 0.74 μW.*

Hardware implementation performance (Table 2) presents hardware data under different technology nodes and implementation architectures. It should be noted that since the algorithms adopt different technology nodes (0.18μm, 0.13μm, 90nm) and implementation architectures (serial/parallel), the area and throughput figures are not directly comparable across algorithms. However, they reflect the relative characteristics of each algorithm. Hardware-optimized algorithms such as PRESENT and GIFT generally use serial architectures to gain area advantages (930-1075 GE). The logic behind this design is to trade more clock cycles for fewer logic gates, making them very suitable for extremely constrained scenarios where speed is not a priority. In contrast, LBLOCK uses a 64-bit parallel architecture. Although its area is larger, its throughput jumps to 200 Kb/s@100kHz, reflecting the trade-off between area and speed in balanced

designs. RECTANGLE demonstrates another possibility: a round-unrolled architecture achieves "slightly larger area but faster speed and lower power consumption", giving it a unique advantage in battery-powered devices that require a certain level of processing capability.

Table 3 Software Performance of Lightweight Block Ciphers (C Implementation, Scenario 0)

Name	Platform	Encryption Speed (cycles/byte)	Code size (byte)	RAM Usage (byte)	Data source
PRESENT-80	AVR	3102	578	32	FELICS
	MSP	2952	547	28	
	ARM	62	698	32	
SIMON64/128	AVR	2701	523	24	
	MSP	2501	502	20	
	ARM	30	647	24	
SPECK64/128	AVR	2601	512	22	
	MSP	2401	493	18	
	ARM	28	618	22	
LBLOCK-80	AVR	2751	552	24	
	MSP	2601	531	20	
	ARM	35	748	24	
GIFT-64-80	AVR	2902	453	24	
	MSP	2751	432	20	
	ARM	40	549	24	
RECTANGLE-80	AVR	2801	501	18	
	MSP	2701	481	16	
	ARM	30	589	18	

Software implementation performance (Table 3) shows that software performance is strongly correlated with algorithm structure. On ARM platforms, the ARX-structured SPECK is the fastest (28 cycles/byte), followed by the Feistel-structured SIMON and LBLOCK (30, 35). On 8-bit AVR platforms, all algorithms slow down to 2400–3100 cycles/byte, but code size (432–698 bytes) and RAM usage (16–32 bytes) remain within the capabilities of extremely constrained nodes. The hundred-fold gap between ARM and AVR suggests that, on 8-bit platforms, pursuing algorithmic structure optimization yields limited returns; upgrading to a 32-bit MCU is more effective than switching algorithms. ARX and Feistel structures, which rely on single-cycle instructions, remain the preferred choice for software implementation on microcontrollers.

Table 4 Security Performance of Lightweight Block Ciphers

Name	Anti-differential analysis	Anti-linear analysis	Resisting other mainstream attacks	Security redundancy	Complexity (Time/Data)	Reference
PRESENT-80	31-round theoretical proof	31-round theoretical proof	Resistant to structured, algebraic, and key scheduling attacks	Full rounds secure	N/A (theoretically secure)	[8]
SIMON64/128	26/44-round key recovery attack	20/44-round distinguisher	Resistant to impossible differential/integral attacks; Vulnerable to side-channel/DFA	18 rounds	$2^{121}/2^{62}$	[9]
SPECK64/128	16/27-round key recovery attack	14/27-round distinguisher		11 rounds	$2^{80}/2^{62}$	
LBLOCK-80	22/32-round differential attack	18/32-round linear attack18/32-round linear attack	Resistant to integral; Vulnerable to DPA/DFA	10 rounds	$2^{64}/2^{58}$	[5,10]
GIFT-64-128	25/28-round related-key rectangle attack	Designer's security bound<=26 rounds secure	Resistant to integral/impossible differential/algebraic attacks; Designed for side-channel resistance	3 rounds	$2^{120.92}/2^{63.78}$	[11]
RECTANGLE-80	18/25-round key recovery attack	18/25-round distinguisher	Resistant to impossible differential/integral/statistical saturation/related-key/slide attacks	7 rounds	Unknown/2^{63}	[7]

A comparison of the security characteristics of lightweight block ciphers with their design objectives reveals a clear logical thread. PRESENT provides a guarantee for hardware optimization with full-round security proof and is resistant to structural and algebraic attacks. SIMON and SPECK trade software efficiency for a precisely controlled security boundary but are vulnerable to side-channel and fault injection attacks. LBLOCK and RECTANGLE strike a balance with quantifiable security redundancy (10 and 7 rounds respectively). As an improved version of PRESENT, GIFT incorporates side-channel

resistance into its design, attempting a more refined trade-off between security, performance and side-channel protection. These differences serve as an important basis for the cross-category algorithm comparison in the following sections: when physical security becomes a constraint, GIFT's side-channel design may be more valuable than PRESENT's area advantage; when compliance requirements are dominant, PRESENT's standardized status plays a decisive role.

3.2.2 Analysis of Lightweight Authenticated Encryption Modes

Table 5 Intrinsic Properties & Security of Lightweight authenticated ciphers

Algorithm	Ascon-128	GIFT-COFB	ISAP	TinyJAMBU	Xoodyak
Standard/Competition	NIST LWC Winner	NIST LWC Finalist	NIST LWC Finalist	NIST LWC Finalist	NIST LWC Finalist
Underlying Primitive	Ascon permutation (320-bit)	GIFT-128 (block cipher)	Ascon / Keccak permutation	Tweakable block cipher	Xoodoo permutation
Key / Nonce / Tag Size (bits)	128 / 128 / 128	128 / 96 / 128	128 / 128 / 128	128 / 96 / 128	128/ 96 / 128
State size (bits)	320	128	320	128	384
One-pass	Yes	Yes	Yes	Yes	Yes
Misuse Resistance	Limited (nonce-respecting)	No	No	No	No
Security Notion	IND-CCA2, INT-CTXT	IND-CCA2	IND-CCA2	IND-CCA2	IND-CCA2
Reference	[12-13]	[12,14]	[12,15]	[12,16]	[12,17]

The five NIST lightweight cryptography finalist AEAD algorithms show clear divergences in design philosophy. ASCON, as the ultimate winner, emphasizes balance. Its configuration of 128-bit keys and tags fully aligns with NIST standard requirements. Although GIFT-COFB, ISAP, TinyJAMBU, and Xoodyak are all finalists, they make different trade-offs in functional completeness. Xoodyak has the largest state size (384 bits), but it uses one-pass processing and supports permutation reuse, providing AEAD, hashing, and XOF in one primitive. TinyJAMBU and GIFT-COFB are based on block ciphers with the smallest state sizes, but they lack misuse resistance. ASCON has a slight advantage in this regard, as its nonce-respecting mechanism provides limited protection in misuse scenarios. ISAP takes a relatively hybrid approach, allowing either ASCON or Keccak permutation. It can be seen that these different design choices directly affect the implementation characteristics: block-cipher-based schemes have small states and compact hardware, while permutation-based schemes tend to trade larger state space for structural simplicity.

Table 6 Comparison of Software and Hardware Performance of Lightweight Authentication Encryption Algorithms[18]

Algorithm	Hardware Area (kGE)	Software Efficiency (ARM Cortex-M3)	Energy Efficiency Feature	Applicable Platform
Ascon-128	11 kGE	Moderate	Comprehensive balance, scalable to 5.5 Gbps throughput	General-purpose software and hardware
GIFT-COFB	8.1 kGE	—	Small hardware area, based on lightweight GIFT block cipher	Hardware priority
ISAP	15.4 kGE	Moderate	Emphasizes security rather than peak performance	High-security scenarios
TinyJAMBU	3.6 kGE	Fast	Minimum area, highest energy efficiency	lightweight software/hardware
Xoodyak	11.9 kGE	Moderate	Integrated multiple functions, reusable permutations	General-purpose software and hardware

The differences in implementation performance are equally clear. For sensor nodes with extreme hardware area constraints, TinyJAMBU (3.6 kGE) and GIFT-COFB (8.1 kGE) are the top choices. The former has a smaller area, while the latter, based on a block cipher, may be easier to reuse with existing hardware modules. For general-purpose devices that need to balance software efficiency and hardware implementation, ASCON and Xoodyak offer more balanced options. ASCON's standardization status adds to its appeal, while Xoodyak's functional integration suits scenarios requiring multiple cryptographic primitives. For applications where security takes precedence over everything else, ISAP, despite its largest area, provides stronger protection due to its security-oriented design.

This divergence echoes the earlier observation on block ciphers: hardware-optimized types (GIFT-COFB, TinyJAMBU), balanced types (ASCON, Xoodyak), and security-first types (ISAP) each have their own focus. However, real-world IoT scenarios impose more complex constraints—resource levels, security requirements, and communication patterns jointly determine algorithm selection. The following section expands on this by considering three typical scenarios.

4 Cross-Category Algorithm and Scenario Adaptation

4.1 Ultra-Low-Power Sensor Networks: Hardware-Optimized Block Ciphers

Ultra-low-power sensor networks typically consist of battery-powered sensor nodes. They are deployed at large scale and are sensitive to maintenance costs. Typical scenarios include environmental monitoring, structural health monitoring, and wearable devices. The core constraints in such scenarios are hardware resources and energy consumption. Therefore, from the perspective of algorithm adaptation, such scenarios should prioritize hardware-optimized block ciphers, with PRESENT and GIFT as typical representatives. The reasons are as follows:

Area efficiency first: Table 2 shows that PRESENT requires only 1075 GE with its 4-bit serial architecture, and GIFT can be compressed to as low as 930 GE, far lower than LBLOCK (1320 GE) and RECTANGLE (1599.5 GE). In ASIC or low-cost FPGA implementations, smaller area directly translates to lower chip cost and smaller physical size.

Power advantage: Although the serial architecture of lightweight block ciphers yields low throughput (e.g., GIFT at 10 MHz achieves only 8.5 Mbps), dynamic power consumption can be significantly reduced by lowering the clock frequency. Taking RECTANGLE as an example, its power consumption at 100 kHz is only 0.74 μW, enabling several years of battery life.

Acceptable security margin: For environmental monitoring data (e. g., temperature, humidity) transmitted in sensor networks, the timeliness requirement for confidentiality is usually not high, but the requirement for long-term reliable operation of the devices is extremely stringent. PRESENT's full-round security proof and GIFT's side-channel-resistant design provide sufficient lifecycle protection.

It should be noted that if sensor nodes need to execute complex network protocol stacks (e. g., 6LoWPAN), software implementation performance must also be considered. In such hybrid scenarios, Feistel structures (e. g., SIMON) or balanced designs (e.g., LBLOCK) may be better choices.

4.2 Real-Time Control Links: Low-Latency Authenticated Encryption

Control links in industrial IoT, connected vehicles, smart grids, and similar scenarios have extremely high requirements for real-time performance and determinism. Typical applications include closed-loop control commands in industrial control systems, cooperative driving information between vehicles, and protection control signals in power grids. The core constraint in such scenarios is low latency and low jitter: encryption operations must complete without affecting the control cycle. Therefore, the core logic of algorithm adaptation is to select low-latency AEAD modes rather than simply pursuing minimal area. Specific recommendations are as follows:

Stream cipher-based AEAD: For example, Trivium+MAC combination or TinyJAMBU. Their bit-by-bit encryption characteristic eliminates the block padding delay inherent in block ciphers. TinyJAMBU has a hardware area of only 3.6 kGE and also exhibits very high throughput on software platforms, making it suitable for control nodes implemented with microcontrollers.

ARX-based AEAD: As the NIST LWC winning algorithm, ASCON uses an ARX-based permutation function that achieves low cycles/byte on ARM Cortex-M processors. At the same time, its sponge structure supports online processing, eliminating the need to buffer complete data packets before performing encryption and authentication.

Parallel implementation: For control gateways implemented in FPGAs or ASICs, algorithms such as Xoodyak that support high-throughput parallel processing can be adopted. Its hardware area is approximately 11.9 kGE, but throughput can be scaled up to Gbps levels, meeting the high-speed requirements of aggregation nodes.

In latency-sensitive scenarios, algorithms with high round counts or complex key schedules should be avoided, because even if their hardware area is small, the resulting computational delay may violate control cycle constraints.

4.3 Bandwidth-Constrained Communication Scenarios: Compression-Friendly Cryptographic Primitives

Satellite IoT, LPWAN (e. g., LoRa, NB-IoT), underwater acoustic communication, and similar scenarios face severe communication bandwidth constraints. The single transmission payload of typical IoT devices may be only tens to hundreds of bytes, and any protocol overhead directly impacts the proportion of effective data. The core requirement in such scenarios is to minimize the additional data length introduced by encryption. Algorithm adaptation should focus on the following two types of designs:

High-compression authentication tags: AEAD algorithms output both ciphertext and an authentication tag. The tag length directly affects communication overhead. All NIST LWC candidate algorithms support 128-bit tags, but for ultra-narrowband scenarios, algorithms that support configurable tag lengths (such as ISAP) can be considered, allowing a trade-off between security and bandwidth.

Non-expanding encryption modes: Some stream cipher-based AEAD algorithms (e. g., TinyJAMBU) do not introduce any padding or expansion during encryption; the ciphertext length is exactly the same as the plaintext length, with only a fixed-length tag added. Compared to block cipher CBC mode (which requires padding to a multiple of the block size), this

can save several bytes of transmission overhead.

4.4 Preliminary Decision Model for Algorithm Selection

Synthesizing the above analysis, a single algorithm metric cannot cover the diversity of IoT application scenarios. This section attempts to construct a practice-oriented decision model for algorithm selection, providing recommendation paths based on three core dimensions:

Resource level: L1 (8-bit MCU, <2KB RAM, battery-powered); L2 (32-bit MCU, 2-32KB RAM, rechargeable); L3 (gateway, stable power supply)

Security level: A (low sensitivity); B (medium sensitivity, remote attacks); C (high sensitivity, physically accessible)

Communication mode: I (low-frequency small packets); II (high-frequency real-time); III (irregular variable-length)

Based on these dimensions, the following algorithm recommendation path can be provided:

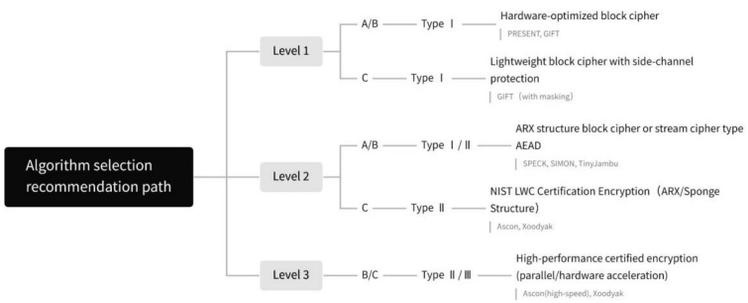

Figure 1 Recommendation path for lightweight cryptographic algorithms based on resources, security and communication mode

In actual selection, the following process can be followed: first, determine the device's resource level (L1/L2/L3), security requirement level (A/B/C), and communication mode characteristics (I/II/III) based on scenario requirements. Then cross-match them with Figure 1 to identify candidate algorithm types. Subsequently, conduct prototype verification of 2–3 candidate algorithms on the target platform, measuring throughput, memory footprint, energy consumption, and other metrics. Finally, perform a security audit, check for side-channel leakage, and evaluate the feasibility of the key management scheme. However, it should be noted that this decision model aims to provide a structured selection framework for system designers, not a rigid rule. In practice, adjustments must be made based on specific scenario constraints.

5 Key Challenges and Critical Reflections

5.1 The Eternal Trade-off Between Security and Efficiency

The core design philosophy of lightweight cryptography is to reduce resource consumption while maintaining basic security. However, defining "basic security" itself is a matter of debate. Whether "lightweight necessarily comes at the cost of security" has not yet reached a consensus between academia and industry. SIMON and SPECK are among the most controversial cases in lightweight cryptography. These two algorithm families, made public by the NSA, did not provide detailed design rationale in their documentation, which has led to long-standing doubts about potential backdoors [4]. Although subsequent cryptanalysis gradually revealed the compactness of their security boundaries, this case also reflects that algorithm selection involves not only technical metrics such as throughput and area, but also considerations of design transparency, security proof strength, and endorsement by standardization bodies. For high-security applications such as government and finance, it is more prudent to choose algorithms that have undergone years of public cryptanalysis and have been adopted by ISO or NIST. For performance-priority scenarios with limited attacker capabilities, the software/hardware efficiency advantages of SIMON and SPECK still offer practical value.

5.2 Physical Attack Threats and Lightweight Protection

The physical accessibility of IoT devices exposes them to serious threats from side-channel attacks and fault injection attacks. For terminals without physical protection, attackers can use lasers, voltage glitches, and other means to induce errors, leading to differential fault analysis. Traditional side-channel countermeasures significantly increase hardware area and power consumption, creating an inherent conflict with lightweight goals. In recent years, algorithm-level side-channel resistance has gradually emerged: GIFT was designed with masking friendliness in mind, enabling first-order masking at relatively low overhead; permutation-based AEADs (Ascon, Xoodyak) naturally reduce leakage by eliminating

key scheduling steps. However, most lightweight cryptography standardization processes have not mandated physical attack protection. Therefore, for high-security devices deployed in physically accessible environments, relying solely on lightweight algorithms is insufficient. Hardware isolation, sensor shielding, or high-order masking must be combined. This is essentially another trade-off between cost and security.

5.3 Real-world Difficulties of Standardization and Interoperability

The fragmented ecosystem of IoT makes the standardization of lightweight cryptography highly challenging. Current mainstream IoT protocol stacks (such as CoAP, MQTT, LoRaWAN, etc.) often do not reserve interfaces for lightweight algorithms in their security layer design. If developers integrate lightweight algorithms themselves, they may break protocol interoperability. The NIST lightweight cryptography standardization project took eight years to select ASCON, but after the standard is released, it will still take time for chip manufacturers, protocol stack developers, and cloud platform vendors to gradually adapt—a cycle that typically takes 3–5 years. During this period, developers may still adopt non-standardized, self-developed solutions, further exacerbating fragmentation.

Meanwhile, differences in byte order, padding modes, and associated data handling across implementations can also cause interoperability failures. Research indicates that in industrial IoT scenarios, the lack of a unified lightweight cryptography implementation specification often forces system integrators to abandon interoperability in favor of proprietary security solutions, which ironically increases overall security risk. Breaking this deadlock requires tripartite collaboration: IoT protocol standards organizations should incorporate NIST LWC standards into mandatory or recommended algorithm suites; chip manufacturers should integrate lightweight cryptography hardware acceleration units into low-cost MCUs to lower integration barriers; and compliance testing and certification mechanisms for algorithm implementations should be established to ensure interoperability across different implementations.

5.4 Future-Oriented Challenges

Among the most concerning directions for the future development of lightweight cryptography are the impact of quantum computing and post-quantum cryptography (PQC), and AI-assisted cryptographic design. Once quantum computers become practical, they will pose a serious threat to the current public-key cryptography system. Although symmetric lightweight cryptography is less affected by Grover's algorithm, key lengths would need to be increased appropriately, which would partially undermine the lightweight advantage. Post-quantum cryptography such as Kyber requires 10–20 KB of RAM and 30–50 KB of Flash on an ARM Cortex-M4, far exceeding the capabilities of L1 sensor nodes. Recent research has reduced Kyber's memory footprint to below 8 KB through state compression [19], but there is still a long way to go before widespread adoption. Therefore, directly deploying PQC on extremely resource-constrained IoT terminals remains a major challenge. In the area of AI-assisted cryptographic design, deep learning has already been applied to differential cryptanalysis and side-channel analysis. For example, Gohr used deep learning to improve differential attacks on Speck32/64. However, AI-generated components are difficult to provide with security proofs, and standardization bodies typically require designs to be transparent and withstand years of human cryptanalysis. Looking ahead, AI is more likely to serve as an auxiliary tool to accelerate human analysis and optimize implementation parameters, rather than completely replacing human designers. For IoT security, applications of AI-assisted automatic code generation and configuration optimization may have more near-term practical value than algorithm design itself.

6 Conclusion

This paper has provided a systematic review and analysis of lightweight cryptographic algorithms for resource-constrained IoT environments, covering application requirements, technical approaches, performance evaluation, scenario adaptation, and key challenges. Unlike traditional surveys that focus on performance ranking, this paper adopts a decision-oriented perspective. By constructing a multi-dimensional evaluation framework (covering intrinsic properties and implementation performance) and analyzing scenario-specific adaptability for three typical application scenarios (ultra-low-power sensor networks, real-time control links, bandwidth-constrained communication), the inherent mapping between algorithm structures and scenario requirements is revealed. The paper also discusses the serious challenges that lightweight cryptography still faces when moving toward practical application: the security-efficiency trade-off, physical attack threats, standardization and interoperability difficulties, and future quantum/AI impacts. The preliminary decision model proposed in this paper aims to provide structured guidance for IoT security designers.

Looking ahead, several research directions deserve significant attention. At the algorithm level, greater emphasis should be placed on design transparency and quantitative proof of security margins, incorporating side-channel resistance into initial designs. For physical security protection, lower-overhead masking or permutation architectures

need to be explored. In standardization and interoperability, efforts are needed to promote the adoption of NIST LWC standards in protocol stacks and chip manufacturers, and to establish interoperability testing mechanisms. For post-quantum cryptography, research could explore extremely compact post-quantum variants, hybrid key exchange protocols, or leveraging physical layer security to reduce reliance on public keys. In the area of AI assistance, the focus could be on engineering-oriented directions such as automated implementation optimization and intelligent detection of side-channel leakage. Lightweight cryptography is one of the foundational technologies for IoT security. There is no universally optimal algorithm, only the most suitable choice based on specific scenario constraints. As new technologies such as quantum computing and artificial intelligence permeate the field, the research boundaries of lightweight cryptography will continue to expand. Its core proposition—providing sufficient security within limited resources—will remain unchanged.

References

[1] KUMARI P, JAIN A K. A comprehensive study of DDoS attacks over IoT network and their countermeasures[J]. Comput. Secur., 2023, 127(C):23.

[2] SILVA C, TENóRIO N, BERNARDINO J. Lightweight Encryption Algorithms for IoT[J]. Computers, 2025, 14(12):505.

[3] POSCHMANN A. Lightweight cryptography-cryptographic engineering for a pervasive world[J]. Cryptology ePrint Archive, 2009.

[4] BEAULIEU R, SHORS D, SMITH J, et al. The SIMON and SPECK families of lightweight block ciphers. IACR Cryptol. ePrint Arch[J]. Preprint at http: //eprint. iacr. org/2013/404, 2013.

[5] WU W, ZHANG L. LBlock: A Lightweight Block Cipher[C]. Applied Cryptography and Network Security, 2011:327–344.

[6] BANIK S, PANDEY S K, PEYRIN T, et al. GIFT: A Small Present[C]. Cryptographic Hardware and Embedded Systems – CHES 2017, 2017:321–345.

[7] ZHANG W, BAO Z, LIN D, et al. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms[J]. Science China Information Sciences, 2015, 58(12):1–15.

[8] BOGDANOV A, KNUDSEN L R, LEANDER G, et al. PRESENT: An ultra-lightweight block cipher[C]. International workshop on cryptographic hardware and embedded systems, 2007:450–466.

[9] ABED F, LIST E, LUCKS S, et al. Differential Cryptanalysis of Round-Reduced Simon and Speck[C]. Fast Software Encryption, 2015:525–545.

[10] LIU S, GONG Z, WANG L. Improved Related-Key Differential Attacks on Reduced-Round LBlock[C]. Information and Communications Security, 2012:58–69.

[11] CAO M, ZHANG W. Related-Key Differential Cryptanalysis of the Reduced-Round Block Cipher GIFT[J]. IEEE Access, 2019, 7:175769–175778.

[12] TURAN M S, TURAN M S, MCKAY K, et al. Status report on the final round of the NIST lightweight cryptography standardization process[J], 2023.

[13] DOBRAUNIG C, EICHLSEDER M, MENDEL F, et al. Ascon v1. 2: Lightweight authenticated encryption and hashing[J]. Journal of Cryptology, 2021, 34(3):33.

[14] CAFORIO A, COLLINS D, BANIK S, et al. A small GIFT-COFB: lightweight bit-serial architectures[C]. International Conference on Cryptology in Africa, 2022:53–77.

[15] DOBRAUNIG C, EICHLSEDER M, MANGARD S, et al. Isap v2. 0[J]. IACR Transactions on Symmetric Cryptology, 2020:390–416.

[16] WU H, HUANG T. TinyJAMBU: A family of lightweight authenticated encryption algorithms (version 2)[J]. Submission to the NIST Lightweight Cryptography Standardization Process, 2021.

[17] DAEMEN J, HOFFERT S, PEETERS M, et al. Xoodyak, a lightweight cryptographic scheme[J]. IACR Transactions on Symmetric Cryptology, 2020: 60–87.

[18] BUCHANAN W J, MAGLARAS L. Review of the nist light-weight cryptography finalists[C]. 2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), 2023:469–474.

[19] HEINZ D, KANNWISCHER M J, LAND G, et al. First-order masked Kyber on ARM Cortex-M4[J]. Cryptology ePrint Archive, 2022.