RESEARCH ARTICLE OPEN ACCESS

# Securing Cloud-IoT Integration: A Comparative Study of Lightweight Encryption Algorithms

Majoju Kiran Kumar*, Prof. Arathi Chitla**
*Department of Computer Science and Engineering, Telangana University, Nizamabad, India
**Department of Computer Science and Engineering, Telangana University, Nizamabad, India

**ABSTRACT**
The convergence of cloud computing and the Internet of Things (IoT) has created a two-tier architecture in which resource-constrained sensing devices generate data that is aggregated, stored, and analyzed by resource-rich cloud infrastructure. This asymmetry — abundant compute at the cloud edge versus severe power, memory, and silicon-area limits at the device edge — is the central security challenge of Cloud–IoT integration. Conventional cryptographic primitives (AES-256, RSA-2048, full TLS 1.3 handshakes) are well suited to the cloud side but frequently exceed the energy, RAM, and latency budgets of embedded sensors, actuators, and RFID-class devices. This paper surveys the lightweight cryptography (LWC) landscape developed to close that gap, covering lightweight block ciphers (PRESENT, SIMON, SPECK, CLEFIA, GIFT), stream ciphers (Grain, Trivium), lightweight hash functions (PHOTON, SPONGENT), and the sponge-based ASCON family that NIST standardized for constrained devices in Special Publication 800-232. We compare these primitives against hardware footprint, energy consumption, and security-margin criteria, and examine complementary cloud-side mechanisms — Attribute-Based Encryption, searchable encryption, and homomorphic encryption — that extend protection into the cloud without pushing that cost back onto the device. Building on this survey, we propose a layered selection framework that maps device and channel classes to an appropriate cryptographic profile, rather than assuming a single 'optimal' algorithm exists across the entire Cloud–IoT stack. We close by identifying open problems that remain unresolved even after NIST's 2023 standardization decision: post-quantum migration for constrained devices, side-channel resistance under real deployment conditions, and the practical heterogeneity of IoT hardware.
***Keywords:*** lightweight cryptography, Internet of Things, cloud security, ASCON, NIST SP 800-232, AEAD, resource-constrained devices, encryption algorithm selection

## I. INTRODUCTION

Cloud–IoT integration has become the default architecture for modern sensing and automation systems: smart metering, connected health monitors, industrial control, and consumer smart-home devices all follow the same pattern of edge sensing feeding cloud-based storage and analytics. This pattern introduces a security boundary that did not exist in either technology alone. The cloud provider is typically treated as a semi-trusted third party capable of storing and processing data, while the IoT device is treated as an untrusted or physically exposed endpoint that may be captured, cloned, or subjected to side-channel analysis.

Security architects therefore face a dual requirement: cryptographic guarantees strong enough to satisfy cloud-grade compliance and audit expectations, delivered by devices that may have only a few kilobytes of RAM, no hardware acceleration, and a battery that must last years without replacement. Lightweight cryptography is the response to this requirement — a design discipline that intentionally trades some of the performance headroom of conventional ciphers for a smaller implementation footprint, while preserving an acceptable security margin for the threat model at hand.

This paper organizes the lightweight cryptography space specifically around its role in Cloud–IoT integration. Rather than treating lightweight ciphers as a standalone research topic, we examine them as one layer in a larger security architecture that also includes gateway-level protocols and cloud-side access-control cryptography. We further propose a selection framework: since IoT deployments are rarely homogeneous, the practical research question is less

'which single algorithm is optimal' and more 'which algorithm is optimal for which device class, at which point in the architecture.'

## II. CLOUD–IOT ARCHITECTURE AND THE SECURITY BOUNDARY

A typical Cloud–IoT deployment has three tiers. The perception tier consists of sensors, actuators, RFID tags, and similar endpoints with severe resource constraints. The edge/gateway tier aggregates traffic from many perception-tier devices, often translating between constrained protocols (6LoWPAN, Zigbee, BLE) and internet-standard transport. The cloud tier receives aggregated data over the public internet, and is responsible for storage, analytics, and access control to downstream applications and users.

Each tier boundary is also a trust boundary. The perception-to-gateway link is frequently wireless and physically accessible to an attacker, making it the most attractive target for eavesdropping, replay, and cloning attacks. The gateway-to-cloud link crosses the public internet and is exposed to conventional network-level threats. The cloud tier itself introduces a distinct concern: the provider must be assumed capable of inspecting stored data unless cryptographic access control is enforced independently of the provider's own permissions model.

This tiered view motivates the structure of the rest of the paper: Section 4 focuses on cryptography appropriate to the perception tier and the perception-to-gateway link, Section 6 covers the standardization outcome most relevant to that tier, and Section 7 addresses cloud-tier mechanisms that operate above and beyond device-level encryption.

## III. THREAT MODEL AND SECURITY REQUIREMENTS

### *A. Adversary Capabilities*

A realistic Cloud–IoT threat model should account for: (a) passive eavesdropping on wireless perception-tier links; (b) active attacks including replay, spoofing, and man-in-the-middle interception at the gateway; (c) physical or semi-invasive access to a captured device, enabling side-channel analysis (power, timing, electromagnetic emission) or key extraction; and (d) an honest-but-curious cloud provider that follows protocol but may inspect stored plaintext or metadata for its own purposes[].

### *B. Security Requirements by Tier*

- Confidentiality of sensor readings and control commands in transit and at rest.
- Integrity and authenticity, typically delivered jointly through Authenticated Encryption with Associated Data (AEAD), to detect tampering and forged commands.
- Availability under constrained energy budgets — a cryptographic scheme that drains a battery prematurely is a denial-of-service vector in its own right.
- Fine-grained access control at the cloud tier, since many users and applications may need different views of the same underlying data stream.
- Forward security and resistance to key-compromise propagation, given the realistic possibility that a single device will eventually be physically compromised.

## IV.LIGHTWEIGHT CRYPTOGRAPHY: MOTIVATION AND DESIGN CONSTRAINTS

Lightweight cryptography (LWC) is not a single algorithm family but a design goal, evaluated against metrics that matter far more at the device edge than in a data center: gate-equivalent (GE) count for hardware implementations, RAM/ROM footprint for microcontroller software, energy consumed per bit encrypted, and latency for real-time control loops. A cipher that is lightweight in silicon area is not automatically lightweight in software, and vice versa — SIMON was optimized primarily for hardware efficiency while its sibling SPECK was optimized for software performance on general-purpose

microcontrollers, illustrating that footprint optimization is implementation-context specific rather than a single scalar property of an algorithm.

The trade-off that makes lightweight ciphers possible is a reduced security margin: smaller internal state, fewer rounds, or simplified key schedules relative to AES-256. This is an explicit, accepted design choice rather than a flaw, provided the deployment's realistic attack budget (attacker compute, physical access window, data lifetime) is matched against the chosen margin. This matching exercise — rather than reflexively choosing 'the strongest available cipher' — is one of the more consequential decisions in Cloud–IoT security engineering.

## V. SURVEY OF LIGHTWEIGHT ENCRYPTION ALGORITHMS

Table 1 summarizes representative lightweight primitives spanning block ciphers, stream ciphers, hash functions, and the AEAD-oriented ASCON family, alongside the asymmetric primitive most commonly paired with them for key exchange. Footprint figures are approximate ranges drawn from published implementation studies; actual figures vary by process node, implementation style, and whether countermeasures (e.g., masking against side-channel attacks) are included.

**TABLE 1. Representative lightweight cryptographic primitives and their fit within a Cloud-IoT architecture.**

| Algorithm | Type | Block/ Key Size | Reported HW Footprint | Primary Strength | Best Fit (Cloud-IoT layer) |
|---|---|---|---|---|---|
| PRESENT | Block cipher (SPN) | 64-bit / 80-128-bit | ~1,000-1,600 GE | Minimal silicon area, ISO/IEC 29192 standardized | RFID tags, passive sensors |
| SIMON | Block cipher (Feistel) | Variable (32-128-bit) | ~1,000-2,500 GE | Very small hardware footprint | ASIC-based sensor nodes |
| SPECK | Block cipher (Feistel/ARX) | Variable (32-128-bit) | Low RAM/ROM in software | Fast on low-power microcontrollers | Software-only MCU endpoints |
| CLEFIA | Block cipher (Gen. Feistel) | 128-bit / 128-256-bit | ~2,400 GE | Strong diffusion, well-studied | Smart cards, embedded controllers |
| GIFT | Block cipher (SPN) | 64-bit / 128-bit | ~1,300 GE | Compact S-box, efficient bit-slicing | Constrained ASIC/FPGA nodes |
| Grain / Trivium | Stream cipher | 80-128-bit key | ~1,000-1,300 GE | Small internal state, streaming-friendly | Continuous sensor telemetry |
| ASCON (NIST SP 800-232) | Sponge-based AEAD & hash | 128-bit key / 320-bit state | ~2,000-2,600 GE | Combined confidentiality + integrity, side-channel resilient design | General-purpose IoT-to-gateway link (NIST standard, 2023/2025) |
| PHOTON / SPONGENT | Lightweight hash | 80-256-bit output | ~1,000-2,000 GE | Integrity checks without full SHA-2/3 cost | Firmware/ data integrity on sensors |
| ECC (Curve25519 / secp160) | Asymmetric (key exchange) | 160-255-bit keys | Higher than symmetric LWC, far below RSA-2048 | Strong security per key bit, small key size | Gateway-to-cloud key exchange |

### *A. Block Ciphers*

PRESENT, standardized in ISO/IEC 29192, was among the first ciphers explicitly designed for extremely small hardware footprints and remains a common baseline in RFID-class deployments [1]. SIMON and SPECK, published by NSA cryptographers, offer a family of block/key size combinations and split the optimization target between hardware (SIMON) and software (SPECK)[2]; their initial lack of published design rationale drew early cryptanalytic scrutiny, which is a relevant consideration when selecting a cipher for long-lived deployments. It has been observed that GIFT improved on PRESENT's S-box efficiency and is frequently used as a building block inside newer AEAD constructions. As discussed in [3] [4], CLEFIA targets a middle ground of moderate hardware cost with stronger diffusion properties, making it a reasonable fit for smart-card-class devices.

### *B. Stream Ciphers*

According to the researchers Grain and Trivium in [5] and [6], it is suggested to prioritize minimal internal state and are well suited to continuous sensor telemetry, where a stream of readings must be

encrypted with low per-symbol overhead rather than in fixed-size blocks.

### *C. Lightweight Hashing and Integrity*

PHOTON and SPONGENT provide integrity-checking capability at a fraction of the cost of SHA-2/SHA-3, which matters when a device needs to verify firmware or data integrity but cannot justify a full general-purpose hash *function*[10].

#### *D. ASCON and AEAD-Oriented Design*

ASCON combines confidentiality and integrity in a single sponge-based construction, avoiding the overhead of composing a separate cipher and MAC. It was originally designed in 2014 by researchers from Graz University of Technology, Infineon Technologies, and Radboud University, and was selected as the primary lightweight cipher in the CAESAR authenticated-encryption competition in 2019 — evidence that it had already withstood several years of public cryptanalytic review before further standardization[7][8].

### *E. Asymmetric Primitives for Key Establishment*

Symmetric lightweight ciphers still require a key-establishment mechanism. Elliptic-curve options such as Curve25519 or secp160 deliver equivalent security to RSA at a fraction of the key size, and are the conventional choice for gateway-to-cloud or device-to-gateway key exchange where some asymmetric-key budget is affordable, typically at the gateway rather than on the most constrained end devices.

## VI. NIST LIGHTWEIGHT CRYPTOGRAPHY STANDARDIZATION

NIST ran a multi-year, multi-round public lightweight cryptography standardization process, narrowing an initial pool of submissions to a set of finalists that included ASCON, Xoodyak, and SPARKLE, among others. In February 2023, NIST announced the selection of the ASCON family as the basis for its lightweight cryptography standard. The finalized standard, published as NIST Special Publication 800-232 (Ascon-Based Lightweight Cryptography Standards for Constrained Devices) in August 2025, defines four concrete algorithms: Ascon-AEAD128 for combined authenticated encryption, and Ascon-Hash256, Ascon-XOF128, and Ascon-CXOF128 for hashing and extendable-output needs, the latter allowing a customization label that reduces the risk of hash collisions across distinct operations performed by the same device [5][6][7].

This outcome is significant for Cloud–IoT research in two respects. First, it gives the field a single, vetted default for the AEAD role that previously required combining a separate cipher and MAC construction. Second, some competing finalists (notably Xoodyak and SPARKLE) reportedly showed better raw throughput in certain benchmark studies; NIST's selection weighted security margin, energy footprint, and memory footprint jointly rather than optimizing for throughput alone, which is a useful reminder that 'optimal' in lightweight cryptography is a multi-criteria judgment, not a single leaderboard position.

## VII. COMPLEMENTARY CLOUD-SIDE SECURITY TECHNIQUES

Device-level lightweight encryption addresses the perception tier and the wireless link, but it does not by itself solve access control once data reaches the cloud. Three complementary techniques are common in the literature:

### *A. Attribute-Based Encryption (ABE)*

According to the author in [13], ABE lets the cloud enforce fine-grained, policy-based access — for example, allowing a maintenance application to read equipment-health attributes while withholding raw location data — without requiring the IoT device to manage a separate key per authorized user. The cryptographic access-control decision is embedded in the ciphertext and the user's attribute-based private key, rather than in the cloud provider's own access-control lists.

### *B. Searchable and Homomorphic Encryption*

As discussed in [14] and [15], Searchable encryption allows the cloud to locate relevant encrypted records (e.g., for a keyword or range query) without decrypting them, while homomorphic encryption allows arithmetic operations to be performed directly on ciphertext. Both address the 'honest-but-curious cloud' element of the threat model in Section 3, at the cost of significant additional computational overhead on the cloud side — overhead that is generally acceptable there, since the cloud tier is not resource-constrained in the way the perception tier is.

### *C. Lightweight Transport-Layer Adaptations*

Constrained devices rarely run a full TLS 1.3 stack. DTLS variants using pre-shared keys, or 6LoWPAN-adapted transport security profiles, reduce handshake overhead while preserving an authenticated, encrypted channel between the device or gateway and the cloud endpoint[6][11].

## VIII. A LAYERED FRAMEWORK FOR OPTIMAL ALGORITHM SELECTION

Because Cloud–IoT deployments span device classes with very different constraints, a single 'optimal' encryption algorithm rarely exists across an entire deployment. Table 2 proposes a mapping from device or link class to a recommended cryptographic approach, intended as a starting point for architecture decisions rather than a fixed prescription.

**Table 2. Device/link class to cryptographic approach mapping.**

| Device / Link Class | Dominant Constraint | Recommended Approach |
|---|---|---|
| Passive RFID / NFC tags | No onboard power source; sub-1,500 GE budget | PRESENT or GIFT for confidentiality; skip public-key operations entirely |
| Battery-powered sensor motes | Energy-per-bit and RAM (<10 KB) | SPECK or Grain in software; ASCON AEAD for combined encryption + authentication |
| Smart edge devices / gateways | Moderate compute, needs interoperability | ASCON AEAD + ECC (Curve25519) key exchange, session keys pushed to endpoints |
| Gateway-to-cloud channel | Higher compute available, but must scale to many devices | Lightweight TLS/DTLS 1.3 profile with PSK or ECC, terminating at the gateway |
| Cloud-side storage & analytics | No hardware constraint; access control is the priority | AES-256 at rest, Attribute-Based Encryption for fine-grained sharing, optional homomorphic/searchable encryption |

The framework's underlying principle is that cryptographic cost should be pushed as far toward the cloud tier as the threat model allows. Passive tags and battery-limited motes should carry only the minimum symmetric AEAD operation required to protect their own data; asymmetric key-exchange and access-control cryptography should be concentrated at the gateway and cloud tiers, where compute is not a scarce resource. This also limits the blast radius of a single compromised device, since long-term asymmetric key material is not typically stored on the most exposed endpoints.

## IX. EVALUATION CONSIDERATIONS AND DISCUSSION

Evaluating lightweight algorithms for a specific Cloud–IoT deployment should combine at least four axes: (1) hardware/software footprint appropriate to the target device class, (2) energy consumption per operation, ideally measured on representative hardware rather than assumed from cycle counts alone, (3) security margin relative to the realistic attacker budget and data lifetime, and (4) side-channel resistance under the physical exposure conditions the device will actually face. A common gap in early lightweight-cryptography literature was over-indexing on footprint and throughput while treating side-channel resistance as a separate, optional evaluation — a gap that later NIST evaluation rounds explicitly tried to close by asking finalist teams to address masking and countermeasure cost as part of the submission, rather than as an afterthought.

A second recurring finding across the literature is that no single algorithm dominates on every axis simultaneously. This is precisely why NIST's public benchmarking discussions noted that some other finalists outperformed ASCON on raw throughput, while ASCON was selected for its more balanced

profile across security, energy, and memory footprint jointly. This paper's recommendation is therefore to treat 'optimal' as a per-tier, per-constraint judgment (Section VIII) rather than searching for a universal winner.

## X. OPEN CHALLENGES AND FUTURE DIRECTIONS

- Post-quantum migration for constrained devices: NIST's post-quantum standards (ML-KEM/FIPS 203, ML-DSA/FIPS 204, SLH-DSA/FIPS 205) were finalized for general use, but shrinking these lattice- and hash-based constructions to fit sub-kilobyte RAM budgets remains an active and largely unresolved research area[12].
- Side-channel resistance in real deployment conditions: laboratory gate-count and cycle-count comparisons do not always predict how a cipher behaves under power-analysis or fault-injection attack in the field.
- Heterogeneity of the IoT device population: a fleet that spans RFID tags, battery sensors, and smart cameras cannot be served by one cipher choice, which argues for the layered, adaptive selection approach proposed in Section 8 rather than a one-size-fits-all standard.
- Long-term key and firmware update management for devices that may remain in the field for a decade or more, well beyond the expected security lifetime of any single cryptographic parameter set.
- Standard-conformance testing and interoperability: as ASCON adoption grows following NIST SP 800-232, cross-vendor interoperability testing will determine how quickly it displaces ad hoc, pre-standard lightweight cipher choices already embedded in shipping devices.

## XII. CONCLUSION

Securing Cloud–IoT integration requires reconciling two very different resource realities: cloud-grade security expectations and device-level constraints on power, memory, and silicon area. Lightweight cryptography — spanning block ciphers such as PRESENT, SIMON, SPECK, and GIFT, stream ciphers such as Grain and Trivium, and the sponge-based ASCON family now standardized by NIST in SP 800-232 — provides the device-level building blocks needed to close that gap. These primitives must be paired with cloud-side mechanisms such as Attribute-Based Encryption and, where applicable, searchable or homomorphic encryption, to extend protection through to the cloud tier's own access-control needs. Rather than seeking a single universally optimal algorithm, the practical path forward is a layered selection framework that matches cryptographic cost to each tier's actual constraints, while keeping an eye on the unresolved challenges of post-quantum migration, side-channel resistance, and long-term key management that will shape the next generation of Cloud–IoT security research.

## REFERENCES

[1] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," in Proc. Cryptographic Hardware and Embedded Systems (CHES), 2007.

[2] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK Families of Lightweight Block Ciphers," IACR Cryptology ePrint Archive, 2013.

[3] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit Blockcipher CLEFIA," in Proc. Fast Software Encryption (FSE), 2007.

[4] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "GIFT: A Small Present — Towards Reaching the Limit of Lightweight Encryption," in Proc. CHES, 2017.

[5] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, "Ascon v1.2: Lightweight Authenticated Encryption and Hashing," Journal of Cryptology, 2021 (CAESAR competition submission, 2014).

[6] National Institute of Standards and Technology, "NIST Selects 'Lightweight Cryptography'

Algorithms to Protect Small Devices," NIST News, Feb. 2023. Available: https://csrc.nist.gov/news/2023/lightweight-cryptography-nist-selects-ascon

[7] National Institute of Standards and Technology, "Ascon-Based Lightweight Cryptography Standards for Constrained Devices: Authenticated Encryption, Hash, and Extendable Output Functions," NIST Special Publication 800-232, Aug. 2025. Available: https://csrc.nist.gov/pubs/sp/800/232/final

[8] M. Hell, T. Johansson, and W. Meier, "Grain: A Stream Cipher for Constrained Environments," International Journal of Wireless and Mobile Computing, 2007.

[9] C. De Cannière and B. Preneel, "Trivium: A Stream Cipher Construction Inspired by Block Cipher Design Principles," in New Stream Cipher Designs, Springer, 2008.

[10] J. Guo, T. Peyrin, and A. Poschmann, "The PHOTON Family of Lightweight Hash Functions," in Proc. CRYPTO, 2011.

[11] W. J. Buchanan, S. Li, and R. Asif, "Lightweight Cryptography Methods," Journal of Cyber Security Technology, vol. 1, no. 3–4, pp. 187–201, 2017.

[12] National Institute of Standards and Technology, "Module-Lattice-Based Key-Encapsulation Mechanism Standard," FIPS 203; "Module-Lattice-Based Digital Signature Standard," FIPS 204; "Stateless Hash-Based Digital Signature Standard," FIPS 205, 2024.

[13] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in Proc. IEEE Symposium on Security and Privacy, 2007.

[14] D. X. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," in Proc. IEEE Symposium on Security and Privacy, 2000.

[15] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in Proc. ACM Symposium on Theory of Computing (STOC), 2009.