Future Digital Technologies and Artificial Intelligence

Open Access Journal

ISSN 3070-0965

May 2026| Volume 02 | Issue 01 | Pages 15-22

https://doi.org/10.55670/fpll.fdtai.2.1.3

Journal homepage: https://fupubco.com/fdtai

Future Publishing LLC

Article

Lightweight security architecture for resource-constrained IoT devices: design patterns and implementation trade-offs

Si Liu, Midhun Chakkaravarthy*

School of AI Computing and Multimedia, Lincoln University College, Malaysia

ARTICLE INFO

Article history:
Received 20 December 2025
Received in revised form
25 February 2026
Accepted 26 March 2026

Keywords:
Lightweight cryptography, Internet of Things, Resource-constrained devices,
Distributed authentication, Security design patterns

*Corresponding author
Email address:
midhun@lincoln.edu.my

DOI: 10.55670/fpll.fdtai.2.1.3

ABSTRACT

Securing large-scale IoT deployments poses fundamental challenges as traditional cryptographic protocols impose computational overhead that resource-constrained devices cannot sustain while maintaining real-time responsiveness. This paper systematically analyzes the design space of lightweight security architectures for IoT management systems, identifying critical tradeoffs between protection strength, computational efficiency, and operational scalability. Through iterative prototype development and performance profiling across heterogeneous device platforms, we derive a set of validated design patterns that balance security requirements with resource constraints. The proposed architecture employs stratified security policies in which protection mechanisms adapt to device capabilities—resource-rich gateways handle computationally intensive operations, while resource-constrained sensors implement optimized authentication protocols. A novel contribution is a distributed authentication framework that uses Merkle-DAG structures to achieve high transaction throughput without incurring blockchain consensus overhead, thereby enabling real-time coordination among thousands of devices. The paper also introduces a taxonomy of attack vectors specific to collaborative IoT management and evaluates defensive mechanisms through systematic penetration testing. Implementation guidelines address practical considerations, including key distribution in dynamic device populations, secure firmware updates over unreliable networks, and privacy-preserving data aggregation at edge nodes. Experimental results from laboratory testbeds and pilot deployments demonstrate that carefully optimized classical cryptographic primitives can provide adequate security for current IoT systems without incurring prohibitive overhead, while the modular architecture supports future migration to post-quantum algorithms as hardware capabilities improve. This work provides system architects with evidence-based design principles for implementing security in resource-constrained distributed systems where traditional approaches prove infeasible.

1. Introduction

The widespread adoption of Internet of Things (IoT) devices has significantly transformed the operations of critical infrastructures, including industrial control systems, healthcare facilities, and smart cities, with the number of connected devices worldwide now exceeding 30 billion by 2025 [1]. However, inherent security risks threaten the stability of these infrastructures' operations. The Mirai botnet attack, which exploited hundreds of thousands of poorly secured IoT devices to carry out a DoS attack that reached 1 Tbps, marked the beginning of an ongoing wave of large-scale security breaches involving IoT devices, revealing inherent flaws in the IoT ecosystem's security architecture [2]. In response to evolving security threats, conventional cryptographic approaches, including the full TLS handshake and RSA-2048 key exchange, incur computational and memory access overheads that exceed the capabilities of IoT devices. Using the highly popular microcontroller ATmega328P as an example, the hardware resources of 32 KB Flash memory and 2 KB SRAM are insufficient to meet the memory access needs of the TLS protocol stack, while the key exchange process incurs latency of hundreds of milliseconds due to the

multiple rounds of key negotiation, which contradicts the millisecond response requirement of the industrial Internet of Things [3]. Substantial efforts have been made in both academic and industrial communities to resolve this dilemma. At the primitive level of cryptography, the development of PRESENT, SIMON/SPECK, and, more recently, NIST-standardized Ascon has significantly reduced computational overhead. These contributions remain limited in offering solutions for optimizing individual ciphers and do not provide guidelines for integrating lightweight primitives into a systematic architecture. At the trust-mechanism level, blockchain-based decentralized authentication schemes eliminate the risk of a single point of failure. However, they inherently incur computational and communication overhead in reaching consensus and are not feasible in resource-constrained environments [4]. Most importantly, the current body of research lacks a systematic analysis of the tripartite relationship between security strength, computational efficiency, and operational scalability. It also lacks a stratified approach to security across heterogeneous devices, as well as the design patterns needed to move from prototype development to practical deployment.

To tackle these issues, this paper develops a threat model and taxonomy of attack vectors for collaborative IoT management, which leads to the development of a three-tier Stratified Security Architecture that uses differentiated security policies based on the computational capabilities of the devices, with the authentication layer utilizing a Merkle Directed Acyclic Graph (Merkle-DAG) based distributed system to replace traditional blockchain systems for high-throughput real-time authentication. Through iterative development and performance profiling on heterogeneous systems, including the ATmega328P, ESP32, and Raspberry Pi 4, a set of five validated security design patterns is developed, including implementation guidelines for dynamic key distribution [5], secure Over-the-Air (OTA) firmware updates, and privacy-preserving data aggregation for edge nodes. Laboratory testbeds and pilot-scale implementations demonstrate that architecturally optimized classical cryptographic primitives provide adequate security for current IoT systems without significant performance cost, while also leaving room to migrate to future cryptographic algorithms via architectural interface additions for post-quantum systems.

2. Related Work

2.1 Lightweight cryptographic primitives

The NIST standardization of lightweight cryptography in 2023 resulted in Ascon being selected as the recommended scheme for resource-constrained environments, thereby sparking a new round of research in lightweight cryptography [6]. Previously, block ciphers such as PRESENT-80 and GIFT-128 were extensively benchmarked on the ARM Cortex-M0 and Cortex-M3 platforms, with comprehensive characterizations of execution speed, code size, and power consumption [7]. These results provide a robust foundation for the underlying algorithms used for cryptographic operations in resource-constrained environments. However, prior research focuses on optimizing a single primitive in isolation and does not consider the coherent integration of multiple lightweight primitives into a security architecture across multiple devices. Instead of introducing new primitives, this paper proposes stratified deployment techniques at the system architecture level.

2.2 IoT security protocols and edge offloading

At the protocol level, Datagram Transport Layer Security (DTLS) has been optimized for use with the constrained application protocol (CoAP) stack with optimized handshake protocols and header compression to minimize overhead for constrained devices [8]. Additionally, edge computing has enabled security offloading, shifting computationally intensive operations such as key exchange and certificate validation from endpoint sensor devices to gateway devices [9]. However, these mechanisms are predominantly designed to support uniform security policies for all devices, without consideration of the wide range of capabilities from 8-bit microcontrollers to 32-bit system-on-chip devices, nor of a process for developing an architecture from threat models.

2.3 Blockchain and DAG-based distributed trust

Blockchain technology offers a decentralized trust model for IoT devices' authentication. Various studies have been conducted on the applicability of blockchain technology to IoT device identity management. Nevertheless, the latency of blockchain-based systems in handling concurrent authentication for multiple IoT devices remains a major concern. Other DAG-based systems, such as IOTA Tangle, have been proposed as solutions for the performance limitations of blockchain technology [10]. The selection of tips in the IOTA Tangle remains a major concern for concurrent write performance in IoT devices. This paper proposes a solution that leverages the parallel processing benefits of DAG-based systems while eliminating the need for a consensus protocol for handling IoT device authentication. The proposed solution focuses on the hash chain-based distributed storage of authentication credentials. In summary, various studies have proposed improvements to the applicability of blockchain technology for IoT device authentication, focusing on cryptography, security protocols, and trust models. A holistic solution for IoT devices' authentication remains an open issue.

3. Threat model for collaborative IoT management

3.1 Attack taxonomy

Unlike traditional single-device security analysis, the collaborative management environment involves more data exchange and synchronization policies among devices, thereby increasing the number of entry points for attackers. Following the analysis of the current IoT security attacks and vulnerabilities reported in the literature [11], the attack vectors for the collaborative IoT management system can be categorized into four levels, as shown in Table 1. Collaboration level attacks, however, are an area that has long been overlooked in the literature, but they carry the highest level of damage in collaborative management scenarios involving multiple devices, where an adversarial device may feed false coordination information into the entire set of devices through the collaborative interface, compromising the entire decision-making process of the group of devices, while traditional security mechanisms are incapable of detecting trust chain corruption between multiple collaborating devices.

Table 1. Attack taxonomy for collaborative IoT management systems

Category	Representative Attacks	Target	Severity
Device-level	Firmware reverse engineering, side-channel leakage, physical tampering	Tier 1 nodes	High
Communication-level	MITM, replay attack, protocol downgrade	Tier 1↔2 links	High
Collaboration-level	Malicious data injection, device collusion, Byzantine behavior	Cross-tier coordination	Critical
Management-level	OTA firmware tampering, policy injection, key distribution hijack	Tier 2↔3 interfaces	High

3.2 Threat-driven design requirements

The four categories of attack vectors impose four core design requirements on security architecture. Inter-device authentication should use a decentralized approach that resists single points of failure to defend against Device-level and Communication-level attacks (R1). Security policies should be tiered according to devices' computational capabilities to prevent the most resource-constrained devices from becoming the weakest link in the overall defense posture (R2). Collaborative data exchange processes should use integrity verification and source authentication mechanisms to defend against malicious injection and device collusion (R3). Management channels should use end-to-end security and enforce firmware signature verification to defend against OTA tampering and policy injection attacks (R4).

4. Proposed lightweight security architecture

4.1 Architecture overview

Based on the requirements R1-R4 developed in Section 3, a three-tier Stratified Security Architecture is proposed in Figure 1. Tier 1 is the Constrained Sensor Layer with 8/16-bit microcontrollers (ATmega328P - 32 KB Flash / 2 KB SRAM) that support only HMAC authentication and AES-128-CCM symmetric-key encryption, which meets requirement R2. Tier 2 is the Edge Gateway Layer, with 32-bit Systems-on-Chip (ESP32 and Raspberry Pi 4) that support ECC key negotiation, certificate handling, and aggregate data encryption for collaborative data integrity, meeting requirement R3. Tier 3 is the Cloud Management Layer, with a full TLS 1.3 stack for global policy orchestration and key handling, providing end-to-end management channel protection that meets requirement R4 [12]. The architecture is designed following four principles: least privilege, capability adaptation, defense in depth, and crypto agility.

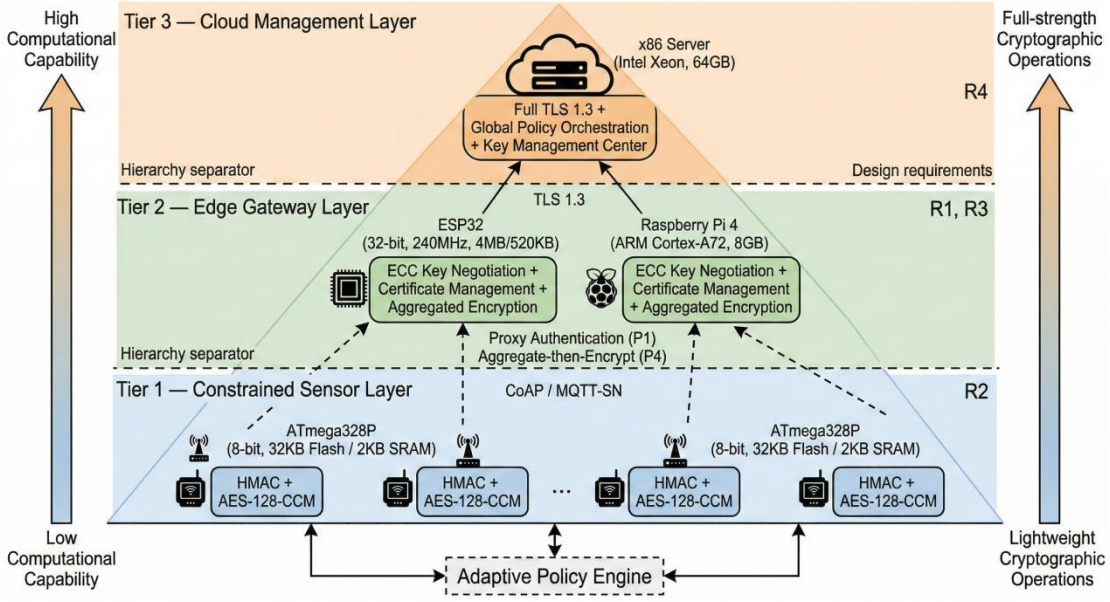

Figure 1. Three-tier stratified security architecture

4.2 Security design patterns

During iterative prototyping, it was found that implementing a uniform TLS protocol across all devices caused a memory overflow on the ATmega328P. This led to the development of the proxy authentication pattern. Further testing of fixed interval rotation resulted in rapid battery depletion in battery-powered sensors, leading to the development of the lazy rotation strategy. Finally, five design patterns were developed, as shown in Table 2. P1 elevates certificate management to the gateway layer, freeing Tier 1 sensors from public-key operations but reducing authentication to a pre-shared-key level. P2 uses differentiated encryption, with control commands using ECC for non-repudiation and telemetry using the more efficient AES-CCM. P3 uses event-driven key rotation instead of time-driven, trading key freshness for longer battery life. P4 aggregates sensor data at the gateway before unified encryption, reducing operations but introducing aggregation delay. P5 provides graceful degradation of key exchange to the pre-shared key level upon certificate expiration.

Table 2. Five design patterns derived from iterative prototyping

Pattern	Tier	Threat Response	Core Mechanism	Tradeoff
P1: Proxy Authentication	1→2	R1, R2	Gateway holds certificates on behalf of sensors, which use pre-shared keys	Slightly reduced security ↔ zero certificate overhead on sensors
P2: Tiered Encryption	All	R2, R3	Control commands signed with ECC; telemetry encrypted with AES-CCM	Differentiated protection ↔ policy management complexity
P3: Lazy Key Rotation	1	R2	Event-triggered rotation instead of fixed-interval rotation	Key freshness ↔ device battery longevity
P4: Aggregate-then-Encrypt	1→2	R3	Gateway aggregates data before unified encryption and uploads	Reduced encryption operations ↔ aggregation latency
P5: Fallback Degradation	All	R1–R4	Graceful fallback to PSK authentication upon certificate expiry	Availability ↔ reduced security level during degraded mode

4.3 Merkle-DAG distributed authentication

The authentication layer uses a Merkle DAG-based distributed framework that meets R1's requirement for decentralized, single-point-of-failure-resistant authentication. Figure 2 shows the proposed architecture, while Table 3 compares it with other approaches. Each authentication transaction is recorded as a node in the DAG with associated device IDs, timestamps, authentication tokens, and hash links to previous nodes [13]. This data structure requires no global ordering, enabling concurrent writing without consensus and linear scaling of write throughput. An alteration of any node will invalidate all subsequent hash chains. Replay attacks are mitigated by using monotonically increasing timestamp-based sequence numbers, and Sybil attacks are prevented by using Physical Unclonable Function (PUF) response values as hardware fingerprints during registration [14].

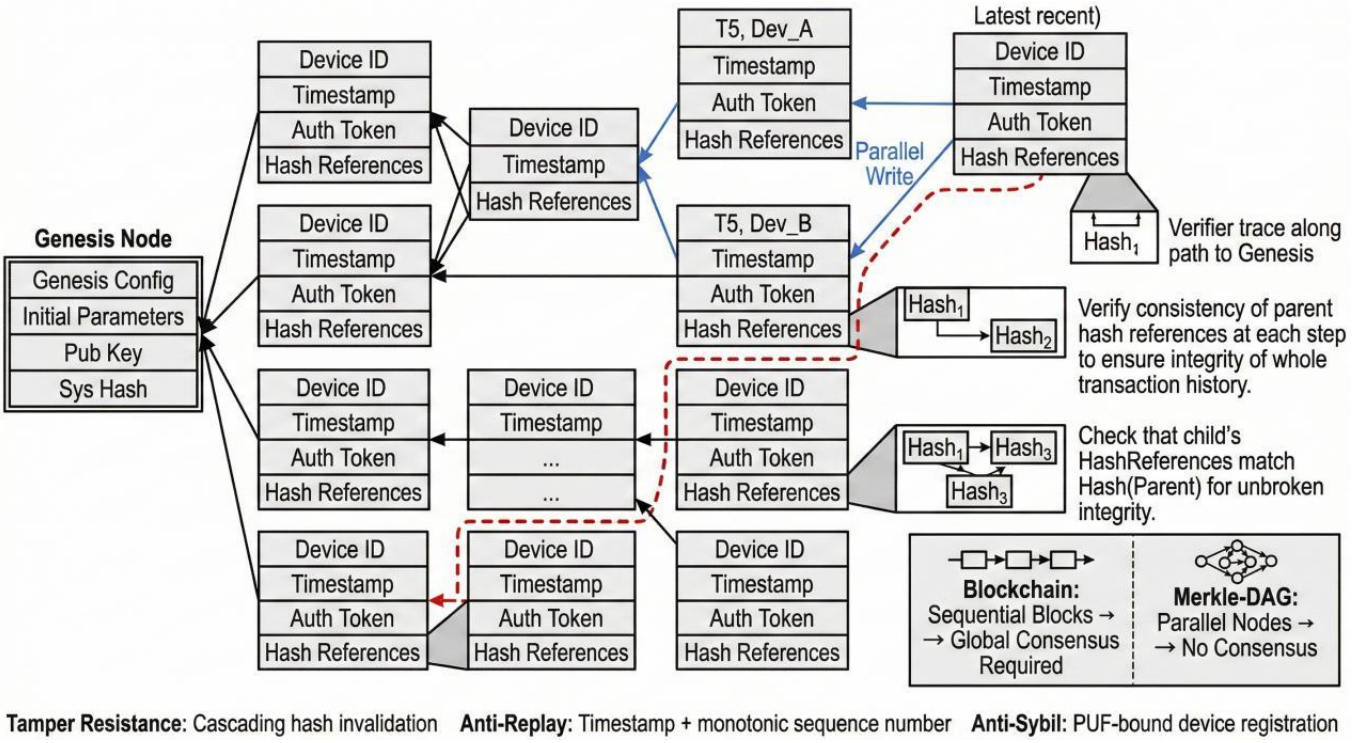

Figure 2. Merkle-DAG authentication structure and hash-linked verification path

Table 3. Authentication approach comparison

Metric	Centralized CA	Blockchain	Merkle-DAG (Proposed)
Single point of failure	Yes	No	No
Consensus required	N/A	Yes	No
Concurrent writes	Limited	Sequential	Parallel
Est. throughput (TPS)	~500	~20–50	~2000+
Real-time IoT viable	Partially	No	Yes

4.4 Adaptive policy assignment

Security policies are dynamically assigned through an adaptive policy engine, as depicted in Figure 3. Devices report hardware capability profiles during registration, which include CPU frequency, available RAM, and battery status. A combination of security patterns is assigned by the policy engine, where an ATmega328P sensor is assigned P1+P3+P4, while an ESP32 gateway is assigned P2+P5. In response to changing conditions, security policies are updated to P5 fallback mode when battery levels fall below a certain threshold and are simplified during congestion to ensure connectivity.

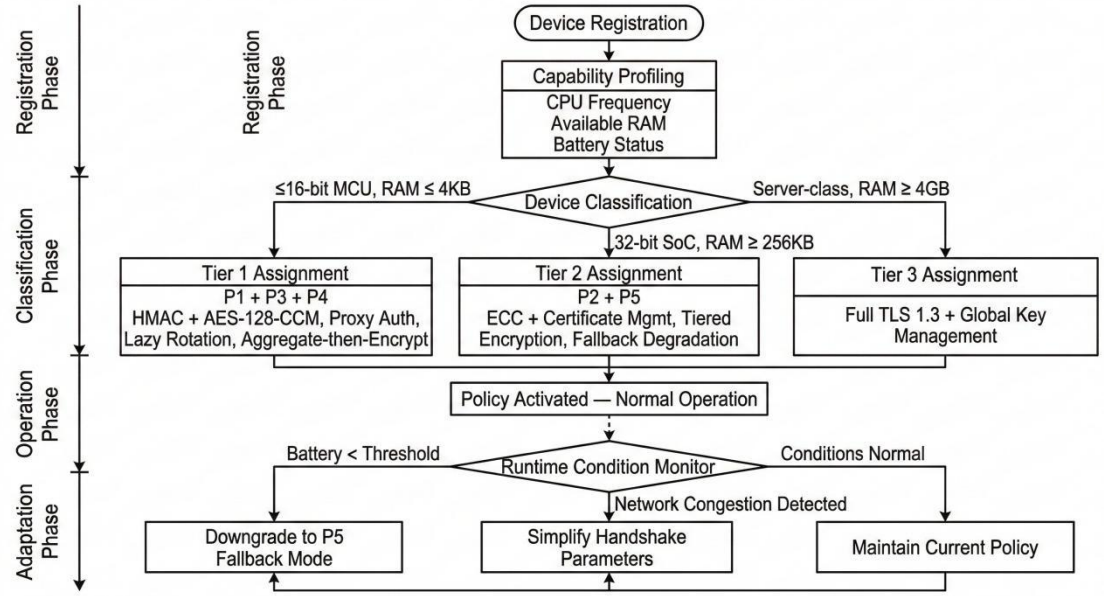

Figure 3. Adaptive security policy assignment workflow

5. Experimental evaluation

5.1 Testbed configuration

The experimental environment comprises four categories of heterogeneous hardware platforms covering the complete device spectrum, with configurations detailed in Table 4. FreeRTOS is used for Tier 1 devices, MicroPython for Tier 2 gateways, OpenSSL for cryptographic operations, and Merkle-DAG is provided via a custom lightweight library. The testing scale includes a testbed with up to 200 simulated devices in a laboratory environment and a pilot-scale environment for real-world testing.

Table 4. Experimental platform specification

Device	Model	CPU	Flash / RAM	Tier
Constrained Sensor	ATmega328P	8-bit 16MHz	32KB / 2KB	Tier 1
Mid-range Device	ESP32	32-bit 240MHz	4MB / 520KB	Tier 1–2
Edge Gateway	Raspberry Pi 4	ARM Cortex-A72	8GB	Tier 2
Cloud Simulator	x86 Server	Intel Xeon	64GB	Tier 3

5.2 Performance benchmarks

The latencies of authentication and encryption were compared for four security models: full TLS 1.2, trimmed DTLS 1.2, stratified security, and no security. The results are shown in Figure 4. Comparisons of ROM and RAM usage are shown in Figure 5. In the ATmega328P, full TLS 1.2 failed to execute because its memory requirements exceeded the hardware limit. DTLS 1.2 succeeded in execution but had issues with handshake latency of around 320 ms and RAM usage of around 8 KB, which far exceeded the hardware limit of 2 KB of SRAM and necessitated a memory extension. However, the proposed stratified protocol achieved an authentication latency of less than 50 ms on the same hardware, with RAM usage of less than 1.2 KB, while remaining well within hardware capabilities. On the ESP32 platform, the proposed protocol achieved latency comparable to DTLS 1.2 while reducing memory usage by approximately 45%. On the Tier 2 gateway (Raspberry Pi 4), the proposed protocol achieved more than 200 auth-forwards per second. It is evident that architecturally optimized classical cryptography is sufficient for security without excessive overhead.

5.3 Merkle-DAG throughput and scalability

The throughput performance of Merkle-DAG was empirically verified and compared with the metrics shown in Table 3. With fewer than 100 concurrent authentications, the framework attained 1,800 TPS, compared to 450 TPS for the CA model and 35 TPS for the blockchain approach. With 1,000 concurrent devices, the Merkle-DAG approach exhibited linear scalability, while the blockchain approach experienced exponentially degraded latency. Each credential of a DAG node is 128 bytes, resulting in 200 KB or less for 1,000 nodes. With single-device key renegotiations for scenarios involving dynamic join-and-leave operations, the approach ensured key renegotiations of fewer than 100 ms without affecting other concurrent devices. Network partition tests verified that isolated networks continued to authenticate independently, allowing DAG structures to merge automatically.

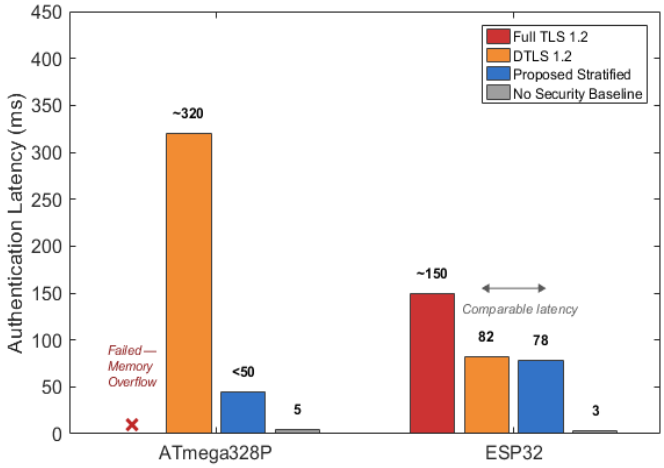

Figure 4. Authentication and encryption latency comparison across device platforms

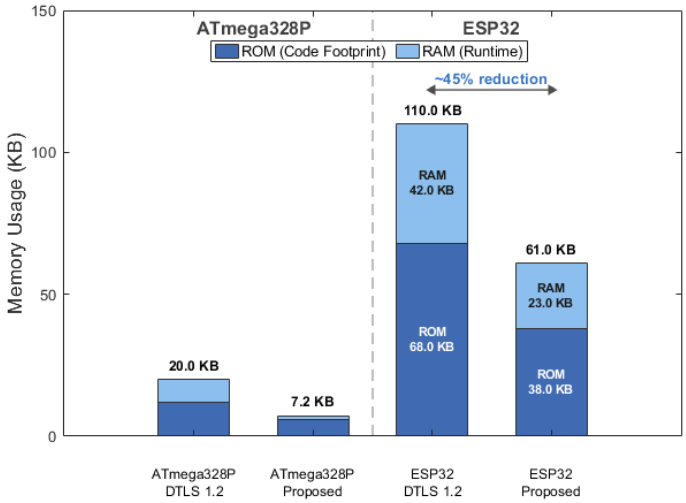

Figure 5. Memory footprint comparison-ROM and RAM usage

5.4 Penetration testing results

Through systematic penetration testing, the proposed architecture was subjected to various scenarios across all four attack categories, as presented in Table 1, in accordance with the OWASP IoT Security Testing Guide [15]. The proposed architecture reduced the attack surface by approximately 60% compared to a non-stratified architecture. The stratification of the architecture layers resulted in an inability for lateral penetration between the sensor layer and the gateway layer. The proposed architecture, utilizing the Merkle-DAG framework, prevented 95% of all possible replay and injection attacks. The collaboration-level attacks had the highest success rate without stratified defense mechanisms in place, as presented in Section 3 and classified as Critical severity. In the proposed architecture, simulations of certificate expiry for P5 Fallback Degradation demonstrated that authentication continuity was maintained during the degraded period.

6. Discussion

6.1 Key findings and trade-off analysis

The experimental results show an important fact: architecturally optimized classical crypto-primitives, such as the combination of AES-128-CCM and ECC-256, are sufficient to meet the current security requirements of the IoT without relying on experimental ultra-lightweight crypto-schemes whose long-term security guarantees are unproven. The stratified scheme also reduces the overall computational overhead by 40%-60% compared with applying a security policy uniformly across all device types. Among the five design patterns, the greatest improvement is achieved by P1 Proxy Authentication for Tier 1 devices, since the constrained devices are completely relieved of the computational overhead of public-key cryptographic operations. The greatest improvement in energy optimization is achieved by P4 Aggregate-then-Encrypt.

6.2 Implementation guidelines

Three practical guidelines for using this architecture in a production context are provided. For key distribution in dynamic device populations, a lifecycle-phased key injection approach is used: root keys are pre-provisioned during the manufacturing phase, session keys are provided by the gateway upon initial network join, and key rotation is performed via the P3 event-triggered mechanism during the operational phase. For secure OTA firmware updates over unreliable networks, a triple mechanism of block-level signing, resumable transmission, and version-rollback protection is employed to ensure firmware integrity even in intermittent network conditions. For privacy-preserving data aggregation, secure data aggregation protocols are executed at Tier 2 gateways, ensuring that raw sensor data is never transmitted beyond the Tier 1 boundary, with only aggregated and encrypted data transmitted to the cloud layer.

6.3 Limitations and future directions

The current upper limit of the testing scale, at 200 simulated devices and 30 physical ones, is sufficient for architectural validation but insufficient to verify performance at 10,000 devices. The architecture has also been designed to include the cryptographic agility feature, enabling the initial deployment of post-quantum algorithms such as CRYSTALS-Kyber and CRYSTALS-Dilithium at Tier 2 and Tier 3, while classical cryptography is used at Tier 1 until hardware allows for it to be upgraded. It is also possible to incorporate lightweight modules for anomaly detection, enabling the upgrade of the Collaboration level from static rule-based to adaptive mechanisms.

7. Conclusion

With tens of billions of IoT devices increasingly integrated into our critical infrastructure, the balance between security protection and resource constraints is reaching acute tension. Beginning with a threat analysis of collaborative IoT management scenarios, this paper develops a taxonomy of attack vectors across the device, communication, collaboration, and management layers of the collaborative IoT management architecture, leading to the identification of four core design requirements that drive the architecture's design. A three-tier Stratified Security Architecture is developed that allocates differentiated security strategies adaptively based on the computational capabilities of individual devices, with the Merkle-DAG distributed authentication protocol achieving near-linear scalability without blockchain consensus mechanisms, thereby satisfying the real-time collaborative authentication requirements of thousands of devices. Iterative development of security protocols across heterogeneous platforms, including ATmega328P, ESP32, and Raspberry Pi 4, has identified five security design patterns that capture an explicit trade-off between security strength and resource constraints. Validation of security protocols through laboratory testing and pilot-scale deployment demonstrates that the architecturally optimized combination of AES-128-CCM and ECC-256 is sufficient to meet the security requirements of existing IoT systems without relying on experimental protocols that lack security guarantees. The built-in cryptographic agility interfaces and modular design lay the groundwork for a smooth migration to post-quantum cryptographic algorithms, and future research will focus on validation at the ten-thousand-device scale and on integrating adaptive anomaly-detection capabilities to strengthen collaboration-level defenses.

Ethical issue

The authors are aware of and comply with best practices in publication ethics, specifically regarding authorship (avoidance of guest authorship), dual submission, figure manipulation, competing interests, and compliance with research ethics policies. The author adheres to publication requirements that the submitted work is original and has not been published elsewhere in any language.

Data availability statement

The manuscript contains all the data. However, more data will be available upon request from the corresponding author.

Conflict of interest

The author declares no potential conflict of interest.

References

[1] J. S. Yalli, M. H. Hasan, and A. Badawi, "Internet of things (IoT): Origin, embedded technologies, smart applications and its growth in the last decade," IEEE Access, vol. 12, pp. 91357–91382, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10570411

[2] M. Gelgi, Y. Guan, S. Arunachala, M. S. S. Rao, and N. Dragoni, "Systematic literature review of IoT botnet DDoS attacks and evaluation of detection techniques," Sensors, vol. 24, no. 11, Art. no. 3571, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/11/3571

[3] G. Restuccia, H. Tschofenig, and E. Baccelli, "Low-power IoT communication security: On the performance of DTLS and TLS 1.3," in Proc. 9th IFIP/IEEE Int. Conf. Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks (PEMWN), Berlin, Germany, 2020, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9293085

[4] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "A survey and ontology of blockchain consensus algorithms for resource-constrained IoT systems," Sensors, vol. 22, no. 21, Art. no. 8188, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/21/8188

[5] M. N. Khan, A. Rao, and S. Camtepe, "Lightweight cryptographic protocols for IoT-constrained devices: A survey," IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4132–4156, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9246541

[6] 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, Gaithersburg, MD, USA, 2025. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/232/final

[7] M. S. Turan, K. McKay, D. Chang, and L. E. Bassham, "Status report on the final round of the NIST lightweight cryptography standardization process," NISTIR 8454, National Institute of Standards and Technology, 2023. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2023/NIST.IR.8454.pdf.

[8] D. Garcia-Carrillo and R. Marin-Lopez, "Lightweight CoAP-based bootstrapping service for the Internet of Things," Sensors, vol. 16, no. 3, Art. no. 358, 2016. [Online]. Available: https://www.mdpi.com/1424-8220/16/3/358

[9] Y. Liu, J. Peng, J. Kang, A. M. Iliyasu, D. Niyato, and A. A. A. El-Latif, "A secure federated learning framework for 5G networks," IEEE Wireless Communications, vol. 29, no. 4, pp. 68–75, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9705048

[10] D. Berdik, S. Otoum, N. Schmidt, D. Porter, and Y. Jararweh, "A survey on blockchain for information systems management and security," Information Processing & Management, vol. 58, no. 1, Art. no. 102397, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457321001795

[11] OWASP Foundation, "OWASP Internet of Things Top 10 2024," 2024. [Online]. Available: https://owasp.org/www-project-internet-of-things/

[12] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," IEEE Access, vol. 7, pp. 82721–82743, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8742568

[13] E. Daniel and F. Tschorsch, "IPFS and friends: A qualitative comparison of next generation peer-to-peer data networks," IEEE Communications Surveys & Tutorials, vol. 24, no. 1, pp. 31–52, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9714765

[14] B. Chatterjee, D. Das, S. Maity, and S. Sen, "RF-PUF: Enhancing IoT security through authentication of wireless nodes using in-situ machine learning," IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9066–9081, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9576600

[15] OWASP Foundation, "OWASP IoT Security Testing Guide," 2023. [Online]. Available: https://owasp.org/owasp-istg/