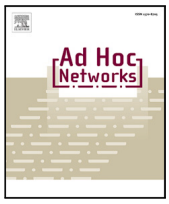
Ad Hoc Networks 181 (2026) 104059



Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc





A blockchain-integrated PUF framework for secure authentication and communication

Koustav Kumar Mondal [a], Debasis Das [b,*], Arpit Khandelwal [c]

[a] Centre for AIoT and Applications, Inter-Disciplinary Research Division, Indian Institute of Technology Jodhpur, Jodhpur, Rajasthan, India
[b] Department of Computer Science and Engineering, Indian Institute of Technology Jodhpur, Jodhpur, Rajasthan, India
[c] Department of Electrical Engineering, Indian Institute of Technology Jodhpur, Jodhpur, Rajasthan, India

ARTICLE INFO

Keywords:
Physical Unclonable Functions (PUF)
Internet of Things
Shamir's Secret Sharing (SSS)
Static monostable
ECC

ABSTRACT

Prior work on Internet-of-Things (IoT) security often splits between hardware roots of trust and decentralized key management: Physically Unclonable Function (PUF) schemes frequently depend on centralized helper infrastructures, whereas blockchain systems typically lack a hardware seed. We present a unified, protocol-enforced framework that integrates PUFs, blockchain-backed Shamir's Secret Sharing (SSS), and elliptic-curve cryptography (ECC) to remove this gap. Concretely: (i) a Static Monostable PUF with error correction derives device keys without storage, achieving $> 95\%$ reconstruction success at $\approx 10\%$ noise; (ii) SSS shares are posted on-chain but re-wrapped every epoch under fresh IND-CPA encryption, eliminating ciphertext staleness and bounding ledger-scraping advantage; and (iii) ECC (Elliptic Curve Cryptography) – including ephemeral Elliptic Curve Diffie–Hellman (ECDH) for forward secrecy and Elliptic Curve Digital Signature Algorithm (ECDSA) for authentication – must rotate each epoch from high-min-entropy PUF material. Our analysis proves hardware-anchored uniqueness, information-theoretic threshold secrecy, and forward secrecy under bounded leakage with non-compounding attacker advantage. We also derive a unified impersonation bound that composes the PUF/ML front-end with the ECC back-end (acceptance probability $\leq \alpha + \alpha_{\mathrm{ML}} + \mathrm{negl}(\kappa)$). A duty-cycle/connectivity model shows constant device-resident state (independent of (t, n)), microjoule–millijoule energy per epoch, and graceful PUF-only operation during outages with automatic re-incorporation of on-chain shares upon reconnection. Containerized experiments across domain-specific deployments – secure supply-chain identification, critical-infrastructure control, and healthcare telemetry – demonstrate sub-second end-to-end handshakes under load, with SSS and ECC costs scaling linearly in t and $\Theta(\log p)$, respectively. These results indicate that enforced rotation plus on-chain ephemerality yields an efficient, scalable, and formally validated tamper-resistant root of trust for next-generation IoT networks.

1. Introduction

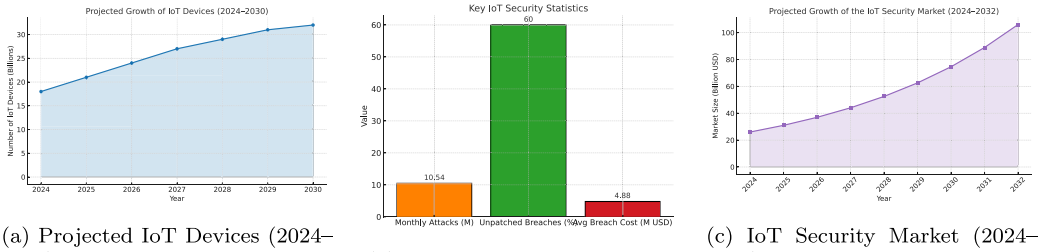
The exponential growth of the Internet of Things (IoT) has ushered in a new era of interconnected smart devices [1], with projections indicating a surge from 18 billion devices in 2024 to over 32 billion by 2030 (see Fig. 1). These devices permeate every facet of our daily lives, enabling intelligent systems for healthcare, transportation, manufacturing, and smart homes [1–3]. However, this rapid proliferation has exposed a significant Achilles' heel: security. The massive and ever-expanding attack surface is not a theoretical vulnerability but one actively being exploited, with IoT-specific attacks reaching over 10.54 million in a single month in late 2022, and 60% of breaches attributed to unpatched firmware. The financial repercussions are severe, with the average cost of a data breach reaching 4.88 million USD in 2024. In response to this escalating crisis, the market is signaling a clear demand for innovative solutions, with the IoT security sector projected to grow from 26 billion USD in 2024 to over 106 billion USD by 2032. The emergence of novel techniques, such as Physically Unclonable Functions (PUFs) [4] and blockchain, offers a promising direction for solving these challenges. Yet, despite their individual potential, integrating them seamlessly into a robust, end-to-end framework remains largely unexplored. This research addresses that critical gap by proposing a unified architecture that leverages the unique strengths of PUFs and blockchain to create a secure, scalable, and lightweight foundation for the next generation of IoT networks.

1.1. Motivation

The explosive proliferation of IoT endpoints – projected to exceed 32 billion devices by 2030 – has dramatically expanded the

* Corresponding author.
E-mail addresses: mondal.4@iitj.ac.in (K.K. Mondal), debasis@iitj.ac.in (D. Das), arpitkhandelwal@iitj.ac.in (A. Khandelwal).

https://doi.org/10.1016/j.adhoc.2025.104059
Received 16 August 2025; Received in revised form 24 September 2025; Accepted 19 October 2025
Available online 24 October 2025
1570-8705/© 2025 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.



(a) Projected IoT Devices (2024–2030) (b) Key IoT Security Statistics (c) IoT Security Market (2024–2032)

Fig. 1. Global IoT landscape: (a) projected device count (2024–2030), (b) key IoT security statistics (incidents and patching rates), and (c) projected IoT security market size (2024–2032). Units: billions of devices; millions of incidents; USD billions.

attack surface of modern networks. Each device, constrained by limited CPU, memory, and power budgets, must perform cryptographic operations under volatile environmental conditions (temperature fluctuations, voltage droops) and in the presence of active adversaries mounting man-in-the-middle breaches, side-channel exfiltration, and device-level impersonation [5,6]. Traditional key-management paradigms (e.g., software-stored secrets or TPM-backed vaults) incur prohibitive energy and latency overheads or require centralized trust anchors that become single points of compromise. Concurrently, advanced approaches – such as PUF-based hardware fingerprints – offer unclonable device identifiers but typically depend on off-chip helper data and centralized provisioning [7]; blockchain-based key sharing decentralizes trust yet lacks a hardware root of randomness [8]. There exists, therefore, a critical unmet need for an end-to-end IoT security architecture that (i) derives unclonable, non-storable keys directly from intrinsic device physics, (ii) distributes global secrets in a tamper-evident, threshold-resilient manner, and (iii) performs lightweight, forward-secret authentication and key exchange within millisecond timeframes. Addressing this trifecta of challenges – hardware-anchored trust, decentralized key resilience, and low-overhead cryptography – forms the central impetus of our work.

1.2. Problem statement

Let an IoT network be composed of N devices, denoted by the set $\mathcal{N} = \{D_1, D_2, \ldots, D_N\}$. We aim to assign each device $D_i \in \mathcal{N}$ a unique cryptographic key $\mathcal{K}_i$ under the following formal constraints:

1. **Unique, Non-Storable Keys:**

$$\forall\, D_i \neq D_j \implies \mathcal{K}_i \neq \mathcal{K}_j, \tag{1}$$

and there must exist no persistent on-chip memory that directly stores $\mathcal{K}_i$. Instead, each key should be derived on-the-fly via an *intrinsic function* of the device's hardware parameters. Formally, let

$$\mathcal{K}_i = \mathcal{H}\Big(f_{\mathrm{PUF}_i}(\theta_i)\Big), \tag{2}$$

where $f_{\mathrm{PUF}_i} : \Theta \to \{0,1\}^m$ is device i's Physically Unclonable Function, θ_i are its unique physical parameters, and $\mathcal{H}(\cdot)$ is a cryptographic hash function that ensures uniqueness and uniformity.

2. **Robust Key Reconstruction Under Noise:** If $\widetilde{\mathcal{K}}_i$ denotes the reconstructed key for device D_i in the presence of environmental noise ϵ_i, we require

$$\Pr\big(\mathcal{K}_i \neq \widetilde{\mathcal{K}}_i\big) \to 0, \tag{3}$$

even as ϵ_i varies over practical ranges of temperature, voltage, and manufacturing deviations. Equivalently, let $\Delta\big(f_{\mathrm{PUF}_i}(\theta_i), f_{\mathrm{PUF}_i}(\theta_i + \epsilon_i)\big)$ be the Hamming distance of the raw PUF responses; the system must correct up to $\max(\Delta)$ errors with high probability:

$$\max_{\epsilon_i} \Delta\big(f_{\mathrm{PUF}_i}(\theta_i), f_{\mathrm{PUF}_i}(\theta_i + \epsilon_i)\big) \leq d_{\mathrm{ECC}}, \tag{4}$$

where d_{ECC} denotes the error-correction capability (in bits) of the employed code.

3. **Secure, Threshold-Based Key Distribution:** Let $\mathcal{K}$ be a secret (e.g., system-wide master key) to be divided among n participants, with a threshold $t \leq n$. We use polynomial-based Shamir's Secret Sharing (SSS):

$$s(x) = \mathcal{K} + \sum_{k=1}^{t-1} a_k\, x^k \pmod q, \tag{5}$$

where q is a prime $> \max(\mathcal{K}, a_k)$. Each device or participant i receives a share $\big(x_i, s(x_i)\big)$. The confidentiality requirement demands that knowing any fewer than t shares yields no information about $\mathcal{K}$. Formally,

$$I\big(\mathcal{K} \mid \text{fewer than } t \text{ shares}\big) = 0, \tag{6}$$

reflecting the *perfect secrecy* property of Shamir's Secret Sharing. Additionally, the distribution of these shares should leverage a *decentralized* infrastructure (e.g., a blockchain) to prevent single points of failure.

4. **Elliptic Curve Cryptography (ECC) Based Authentication and Secure Communication:** We require an authentication protocol for each device D_i such that it can prove possession of $\mathcal{K}_i$ without revealing it. Define the Elliptic Curve parameters as:

$$\big\{\mathbb{F}_p,\ E:\ y^2 = x^3 + a\,x + b,\ G,\ n\big\}, \tag{7}$$

where G is a generator of the prime-order subgroup of cardinality n. Each device forms an ECC key-pair (d_i, Q_i) where

$$d_i = \mathcal{K}_i \bmod n, \quad Q_i = d_i\, G. \tag{8}$$

The framework must satisfy:

$$\text{ECDLP Hardness: } \Pr\big(\text{Recover } d_i \mid Q_i\big) \approx 0 \quad \text{in poly}(p) \text{ time.} \tag{9}$$

thereby ensuring security against Elliptic Curve Discrete Logarithm Problem (ECDLP) based cryptanalysis. Furthermore, devices need to establish shared secrets securely via Elliptic Curve Diffie–Hellman (ECDH):

$$K_{ij} = d_i\, Q_j = d_j\, Q_i \in E(\mathbb{F}_p), \tag{10}$$

with the property that any eavesdropper lacking d_i or d_j faces the intractable Computational Diffie–Hellman Problem (CDHP).

Therefore our overall problem statement is design a unified cryptographic framework that (i) derives device-specific unclonable keys via PUF, (ii) guarantees near-zero key reconstruction errors, (iii) distributes global keys or session secrets securely under threshold constraints, and (iv) provides efficient elliptic-curve-based authentication and secure communication within the tight computational $\big(O(\log p)\big)$ and storage limits typically found in IoT devices.

Table 1
IoT authentication protocols (non-PUF-centric): crypto design vs. operational properties.

Reference	Core technique	PFS	Eph. rotation	Overhead
Yadav et al. [9]	ECC + Symmetric keys	Optional (◐)	None (✗)	Moderate
Gao et al. [10]	ECC + Threshold	Full (✓)	Optional (◐)	Moderate
Wang et al. [11]	Ring signatures on blockchain	Optional (◐)	Optional (◐)	High
Zheng et al. [12]	Chaotic maps + blockchain	Optional (◐)	Optional (◐)	Mod–High
He et al. [2]	ECC challenge–response	Full (✓)	Optional (◐)	Low
Wazid et al. [13]	ECC + Hash	Full (✓)	None (✗)	Moderate
Proposed work	**PUF + (Blockchain) SSS + ECC with enforced rotation**	**Full (✓)**	**Full (✓)**	**Low**

1.3. Contributions

The primary contribution of this work is the design, validation, and analysis of a end-to-end security framework that bridges the gap between hardware-anchored trust, decentralized key management, and efficient cryptography for IoT networks. We introduce a unified framework that synergistically combines PUFs, blockchain-based SSS, and ephemeral ECC to provide a scalable and robust root of trust. Our specific contributions are as follows:

1. **Protocol-enforced ephemerality with per-epoch re-wrapping:** Keys are mandatorily rotated each epoch from fresh PUF/extractor output, and on-chain Shamir shares are re-encrypted under new IND-CPA ephemeral keys—eliminating ciphertext staleness and clearly distinguishing our design from prior PUF+blockchain schemes with optional or static rotation.
2. **Composable guarantees with bounded-leakage security:** We provide a single proof that composes hardware uniqueness, threshold secrecy, and forward secrecy, yielding a system-level acceptance bound $\Pr[\text{accept}] \leq \alpha + \alpha_{\text{ML}} + \text{negl}(\kappa)$ and a leakage-aware extractor whose advantage does not compound across epochs; during outages, PUF-only rotation preserves forward secrecy.
3. **Practical scalability validated in domains:** An $O(t)$ Shamir reconstruction (constant device state) and $\Theta(\log p)$ ECC costs deliver sub-second handshakes with <3% jitter in supply chain, critical infrastructure, and healthcare settings, mapping governance to (t, n) and rotation to risk; a heterogeneous testbed confirms feasibility, and PUF reliability >95% under ≈10% noise supports the assumptions.

2. Related work

This section organizes the literature along three tightly coupled axes that directly motivate our integrated design: (i) PUF-based identification in ultra-low-power or battery-less settings, where the physical interface constraints dominate protocol shape; (ii) SRAM-based device fingerprinting integrated with lightweight cryptographic protocols, where stable silicon-intrinsic features are combined with error correction and compact crypto; and (iii) blockchain-assisted accountability for physical object authentication in distributed systems, where decentralized trust, threshold recovery, and revocation govern end-to-end assurances. Throughout, we indicate where prior work treats ephemeral rotation as optional policy rather than a protocol invariant, a gap our framework closes by mandating rotation and per-epoch re-wrapping of on-chain shares.

2.1. PUF-based identification and tag architectures for low power or battery-less systems

Identification-class devices – RFID tags, energy-harvesting labels, and passive/ battery-less nodes – prioritize μJ energy envelopes and bit-level simplicity in challenge–response flows. Rullo et al. design a PUF-oriented tag stack that emphasizes a stable PUF front-end and a streamlined protocol, explicitly minimizing the hardware attack surface in smart-ID tags [4]. In clinical workflows, Gope et al. (“QR-PUF”) combine XOR-Arbiter PUFs with QR codes to achieve secure, low-power inpatient identification and introduce obfuscation to reduce ML modeling success [14]. At the extreme-constrained frontier, Sangar et al. show that battery-less PACT tags extend range via hybrid backscatter/active radios, tightening the latency/energy budgets that PUF protocols must satisfy [15].

These designs demonstrate (i) repeatable hardware uniqueness at tag-class power, (ii) robustness to moderate environmental variation via PUF stabilization, and (iii) protocol minimalism suitable for sub-mW budgets. However, most ID/tag stacks rely on static or infrequently rotated cryptographic contexts; i.e., forward secrecy and epoch-wise key churn are optional and typically not hardware-enforced. Our framework keeps the low-state virtues (no stored long-term secrets) but enforces per-epoch rotation and binds any threshold recovery to a hardware seed, preserving the operational simplicity demanded by tags while closing persistence windows.

2.2. SRAM-based device fingerprinting and its integration with lightweight cryptographic protocols

SRAM start-up states provide high-entropy, device-unique bitstrings without special macros, making them attractive across embedded IoT. Kohli et al. leverage SRAM power-up fingerprints and an autoencoder-based verifier to achieve accurate device recognition with modest compute [17]. Jain et al. integrate SRAM-PUF with ECC decoding to derive *one-time* AES keys in IIoT content-protection pipelines, pushing practice toward forward-secrecy-like operation on constrained MCUs [18]. Complementary surveys and systems (e.g., [24–26]) detail fuzzy extractors, helper-data secure sketches, and coding strategies that stabilize PUF responses under temperature/voltage excursions.

Two recurring limitations emerge. First, many SRAM-PUF systems still depend on centralized helper infrastructures [27], creating a single point of failure and complicating distributed attestation. Second, ephemeral key rotation is often a policy (recommended) rather than a protocol invariant; when rotation is skipped or delayed, an exposed long-term credential can endanger future sessions. Our architecture addresses both by (i) anchoring ECC credentials in error-corrected PUF material each epoch, and (ii) coupling an on-chain, threshold-protected auxiliary secret that is re-wrapped under fresh IND-CPA encryption every epoch. This preserves the lightweight footprint of SRAM-derived seeds while delivering hardware-anchored forward secrecy in operation, not just in principle.

2.3. Blockchain assisted accountability models applied to physical object authentication in distributed systems

Decentralized ledgers provide tamper-evident storage, auditability, and cross-domain accountability; when combined with threshold cryptography, they enable resilient recovery without central trust. Mahmood et al. place PUF-derived shares on-chain using Shamir’s SSS for decentralized keying (PUF+BC) [19]; Barbareschi et al. implement SRAM-PUF with blockchain-backed authentication on MCUs to remove

Table 2
The landscape of integrated security architectures: a categorized comparison of recent PUF, SRAM, and Blockchain-based approaches.

Reference	Core technique/Focus	PFS	Eph. rotation	Overhead
PUF-based identification and tag architectures for low power or battery-less systems				
Rullo et al. [4]	Stable PUF front-end + streamlined tag authentication (ID tags)	N/A (ID/auth focus)	None (✗)	Low
Gope et al. [14]	XOR-Arbiter PUF + QR for secure inpatient RFID; ML obfuscation	N/A/Optional	None (✗)	Low
SRAM-based device fingerprinting and integration with lightweight cryptographic protocols				
Sahu et al. [16]	PUF + Symmetric MAC (vehicular/IoV)	Full (✓)	Optional (◐)	Low
Kohli et al. [17]	SRAM power-up fingerprint + autoencoder verification (embedded IoT)	N/A/Optional	None (✗)	Low
Jain et al. [18]	SRAM-PUF + ECC decoding to derive one-time AES keys (Industrial IoT)	Full (✓)	Optional (◐)	Low
Blockchain assisted accountability models applied to physical object authentication				
Mahmood et al. [19]	PUF + Blockchain SSS for decentralized keying	Full (✓)	None (✗)	Low
Barbareschi et al. [20]	SRAM-PUF + Blockchain authentication on MCUs (no stored keys)	Optional (◐)	Optional (◐)	Low
Jiang et al. [21]	Two-layer collaborative blockchains + ECC/CL-PKC (CPS power systems)	Optional (◐)	Optional (◐)	Mod
Xu et al. [22]	Blockchain-assisted CPS data sharing with deniability/traceability	Optional (◐)	None (✗)	Moderate
Wang et al. [23]	Blockchain-assisted revocable anonymous authentication (IIoT)	Optional (◐)	Optional (◐)	Low–Mod
Proposed (this work)	**PUF + (Blockchain) SSS + ECC with *enforced* rotation**	**Full (✓)**	**Full (✓)**	**Low**

stored keys and central authorities [20]. In cyber–physical systems, Jiang et al. propose two-layer collaborative blockchains with ECC/CL-PKC for cross-domain verification and accountability [21]; Xu et al. add deniability and multi-recipient encryption for CPS data sharing [22]; Wang et al. design revocable, anonymous authentication for IIoT [23].

Beyond PUF-anchored designs, non-PUF blockchain authentication includes ring/group signatures [11] and chaotic-map constructions [12], while ECC-focused IoT protocols (without blockchain) emphasize low-cost forward secrecy [2,13] or threshold availability [10] and mixed ECC+symmetry stacks [9]. These lines contribute anonymity, revocation, and cross-domain governance but often (i) lack a hardware root of randomness, and/or (ii) treat key rotation as an optional mode. Our framework unifies these directions by binding threshold recovery to hardware (PUF+fuzzy extractor) and enforcing epoch-wise re-derivation of ECC credentials alongside per-epoch re-encryption of on-chain shares, thereby minimizing ciphertext staleness and bounding ledger-scraping advantage. For completeness, we retain two concise, tables (Table 1 & Table 2) that synthesize the operational properties of (A) non-PUF-centric IoT authentication protocols and (B) PUF-anchored and blockchain-assisted schemes across IoT/CPS/IIoT. The proposed work is included side-by-side to clarify differences in forward secrecy, enforced rotation, and effective overhead.

2.4. Research gap

The preceding survey reveals a persistent architectural schism in IoT security: systems that anchor trust in hardware primitives like PUFs often remain dependent on centralized helper data infrastructures [27], while decentralized accountability models built on blockchains typically lack a hardware-based root of randomness and trust [28]. Even as individual technologies mature – from stabilized PUF tags [4] to SRAM-derived one-time keys [18] and privacy-preserving blockchain governance [22,23] – a critical gap remains. State-of-the-art frameworks rarely achieve a cohesive synthesis that simultaneously (i) binds decentralized, threshold-based secret recovery directly to a hardware seed, and (ii) elevates ephemeral key rotation from an optional policy to a non-negotiable, protocol-enforced invariant.

This work closes that gap. We introduce a unified architecture where each security primitive addresses the limitations of the others. Device identity is anchored in an error-corrected PUF, eliminating stored keys. Global secrets are managed via blockchain-backed SSS, removing central points of failure. Crucially, we mandate a dual-ephemerality model: ECC credentials must be regenerated from fresh PUF material each epoch, and on-chain SSS shares must be re-wrapped under new IND-CPA encryption. This protocol-enforced rotation provides hardware-anchored forward secrecy in operation – not just in principle – and yields a formally validated, tamper-resistant foundation for next-generation IoT networks.

3. Proposed framework overview

Our proposed Blockchain-Integrated PUF Framework integrates three core components: Key Generation using Static Monostable Physically Unclonable Functions (PUFs), Key Distribution using Blockchain-Based Shamir's Secret Sharing (SSS), and Authentication and Communication using Elliptic Curve Cryptography (ECC). Each component is mathematically modeled and simulated to address the challenges of secure device authentication and communication in resource-constrained IoT networks. As shown in Fig. 2, the cryptographic framework operates through the following key stages:

1. **Key Generation:** Each device generates a unique and stable cryptographic key from its intrinsic physical characteristics using a PUF. This process leverages error correction to ensure reliability against environmental noise, producing a dependable hardware-anchored key without needing to store it permanently.
2. **Key Distribution:** This PUF-derived key, or another global secret, is distributed securely using Shamir's Secret Sharing (SSS). The secret is split into multiple shares which are managed via a decentralized blockchain, eliminating single points of failure. The scheme ensures that only an authorized threshold of participants can reconstruct the original secret.
3. **Authentication and Communication:** The device-specific key is used as a seed to derive Elliptic Curve Cryptography (ECC) credentials. These credentials enable strong, lightweight device authentication using the Elliptic Curve Digital Signature Algorithm (ECDSA) and establish secure communication channels with forward secrecy via the Elliptic Curve Diffie–Hellman (ECDH) key exchange protocol.

In the following sections, we provide detailed mathematical models and algorithms for each of these framework components.

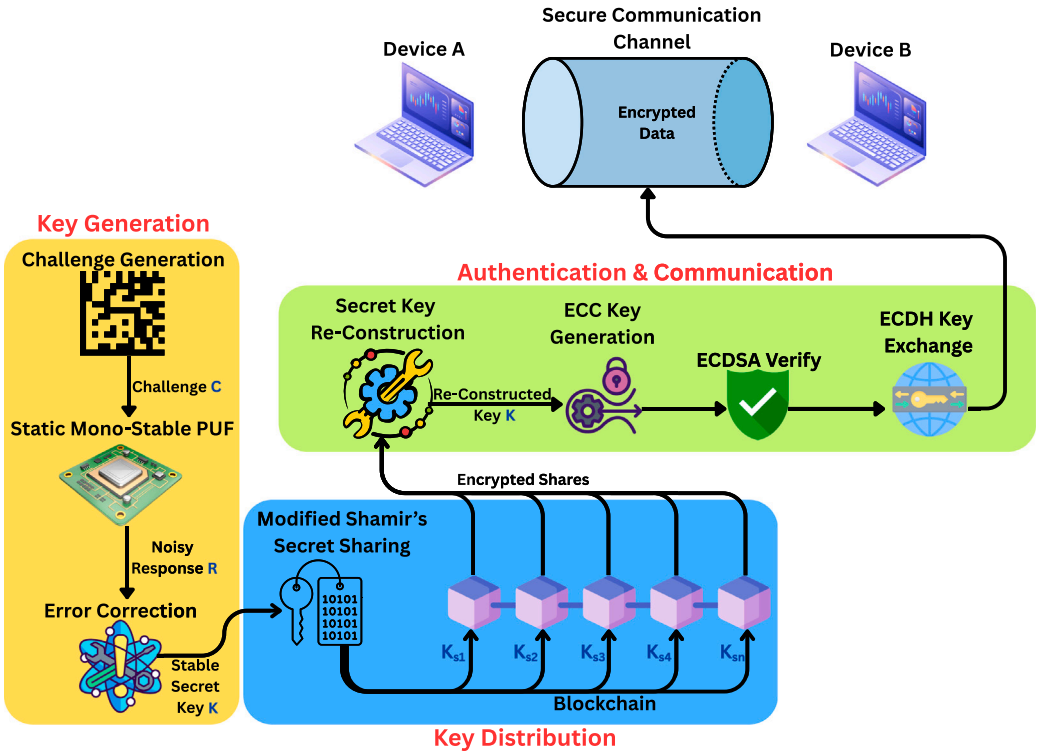


Fig. 2. End-to-end architecture of the proposed framework: PUF–based seed generation → SSS with blockchain anchoring → ECC for authentication and key exchange. Inputs/outputs are shown per stage.

4. Key generation

4.1. Mathematical modeling of static monostable PUFs

Physically Unclonable Functions (PUFs) exploit the inherent physical variations in integrated circuits (ICs) resulting from the manufacturing process. These variations are unique to each device and can be used to generate device-specific cryptographic keys without storing them in memory. Let:

- $\theta \in \Theta$ denote the physical parameters unique to each device.
- $C \in \mathcal{C}$ represent the challenge input vector.
- $R \in \mathcal{R}$ be the PUF response vector.

The PUF is modeled as a function:

$$R = f_{\text{PUF}}(C, \theta) \tag{11}$$

Eq. (11) describes how the PUF maps a given challenge and device-specific parameters to a response. To ensure consistency, the PUF must be robust against environmental variations such as temperature (T) and voltage (V). We require:

$$\frac{\partial R}{\partial T} \approx 0 \quad \text{and} \quad \frac{\partial R}{\partial V} \approx 0 \tag{12}$$

Eq. (12) ensures that the response R remains stable under variations in T and V.

4.2. Error correction and key derivation

During the key generation process, the response R obtained from the Physically Unclonable Function (PUF) may suffer from transient noise and minor environmental variations. These perturbations can lead to inconsistencies in R, potentially reducing the reliability of the derived cryptographic key. To address this challenge, we integrate an error-correcting code (ECC) to stabilize the PUF response, ultimately obtaining a stable response $\tilde{R}$ from which a cryptographic key K is derived.

The process is conceptually divided into two main phases: an *Enrollment Phase* and a *Reconstruction Phase*. During enrollment, we generate helper data W that is stored and utilized later during reconstruction to correct any errors induced by noise. After obtaining the error-corrected response $\tilde{R}$, we apply a cryptographic hash function to produce a uniformly distributed key K.

4.2.1. Enrollment phase

The initial, noise-free PUF response is modeled based on Eq. (11). In Eq. (11), C is a challenge vector and θ encapsulates the device-specific physical parameters intrinsic to the PUF structure. The function $f_{\text{PUF}}(\cdot,\cdot)$, determined by the microfabrication variations, maps the input challenge C to a binary response vector R. Under ideal conditions (no noise), R is stable and repeatable.

Codeword generation. To enable error correction, we employ an error correcting code defined by a generator matrix G. Multiplying the binary PUF response R by G in a finite field (usually GF(2) for binary codes) produces a codeword c:

$$c = RG \tag{13}$$

In practice, R is a binary vector of length n, and G is an $n \times m$ generator matrix that maps an n-bit message (the PUF response) into an m-bit codeword c. The additional redundancy bits in c are critical for detecting and correcting errors during the reconstruction phase.

Helper data computation. We define the helper data W as the bitwise XOR of R and c:

$$W = R \oplus c \tag{14}$$

The helper data W encodes the relationship between the original PUF response R and the codeword c. Importantly, W does not reveal R directly; instead, it stores the information necessary to guide the reconstruction process. Since W is typically stored in non-volatile memory or on a secure server, it is crucial that W does not leak information about R that would compromise the security of the PUF-based key generation.

4.2.2. Reconstruction phase

During the reconstruction phase, environmental factors (e.g., temperature variations, supply voltage fluctuations) may introduce a noise vector ϵ, resulting in a noisy response:

$$R' = f_{\text{PUF}}(C, \theta) + \epsilon \tag{15}$$

Here, R' is the response obtained under realistic conditions. The noise vector ϵ represents small, random perturbations flipping some bits of the ideally stable response. While ϵ is typically small, even a few flipped bits can disrupt key generation if not corrected.

Codeword recovery. To correct the noisy response R', we exploit the stored helper data W. By XORing R' with W, we retrieve a perturbed codeword c':

$$c' = R' \oplus W \tag{16}$$

Because $W = R \oplus c$, we have:

$$R' \oplus W = R' \oplus (R \oplus c) = (R' \oplus R) \oplus c. \tag{17}$$

If $R' = R$ (no noise), then $c' = c$, perfectly retrieving the original codeword. In the presence of noise, c' will differ from c by a small number of bits. With c' in hand, we decode it using the ECC's decoding algorithm. This step corrects any bit errors introduced by noise, recovering the original codeword $\tilde{c}$. The decoding process typically involves syndrome computation and the application of a correction algorithm (e.g., for Hamming codes, BCH codes, or Reed–Solomon codes, depending on the chosen ECC scheme). The result is a noise-free codeword $\tilde{c}$. From this corrected codeword $\tilde{c}$, we reverse the process (e.g., by removing redundancy bits) to extract the stable response $\tilde{R}$. This response now matches the originally enrolled PUF response R (or its stable equivalent) despite any environmental interference.

Hash-based key derivation. Finally, to derive the cryptographic key K, we apply a cryptographic hash function $H : \mathcal{R} \to \mathbb{F}_q$ to the stable response $\tilde{R}$:

$$K = H(\tilde{R}) \tag{18}$$

The hash function H ensures that K is uniformly distributed and free from structural biases that might be present in $\tilde{R}$. Since H is a one-way function, the resulting key K cannot be feasibly inverted to recover $\tilde{R}$ or R, ensuring strong security properties. The randomness and uniformity of K are crucial for cryptographic primitives, such as generating ECC keys, employing encryption or authentication schemes, and resisting statistical or modeling attacks. The combination of PUF-based variability, ECC-driven error correction, and hash-based key derivation ensures that the final cryptographic key K is both stable and secure, resistant to environmental perturbations and cryptanalytic attacks. This integrated approach is well-suited for resource-constrained IoT devices, enabling secure authentication and communication protocols without the need for permanent key storage.

5. Key distribution

5.1. Modified Shamir's secret sharing scheme

To securely distribute and store the cryptographic key K, we utilize Shamir's Secret Sharing (SSS), which is based on polynomial interpolation over finite fields.

1. **Choose Finite Field:**
   - Select a prime q such that $q > K$.
   - Define the finite field $\mathbb{F}_q$.
2. **Construct Secret Polynomial:**

$$s(x) = K + \sum_{i=1}^{t-1} a_i x^i \mod q \tag{19}$$

where $a_i \in \mathbb{F}_q$ are random coefficients, and t is the threshold.

3. **Generate Shares:**
   - For $i = 1$ to n:

$$y_i = s(x_i) \mod q \tag{20}$$

where $x_i \in \mathbb{F}_q \setminus \{0\}$ are distinct non-zero elements.
   - Each share is (x_i, y_i).

Algorithm 1 Unified PUF-anchored single-pass authentication and ECC key agreement with enforced rotation.

Inputs: PUF function, error correction, encrypted Shamir shares, elliptic-curve domain parameters.
Outputs: ephemeral ECC key pair, ECDSA signature, ECDH session key.
Preconditions: constant-time ECC; bounded leakage for extractor.
Complexity: $\Theta(\log p)$ per scalar multiplication; $O(t)$ SSS reconstruction with prefix/suffix products.

Require: PUF function $f_{\text{PUF}_i}(\theta_i)$, error-correction $\mathcal{E}$, Shamir shares $(x_i, \text{EncShare}_i)$ on blockchain, ECC domain (p, E, G, n).
Ensure: Device D_i obtains ephemeral ECC credentials (d_i, Q_i), enabling ECDSA authentication and ECDH key agreement, continuously rotated to thwart persistent attackers.

(1) PUF-Based Seed and Share Decryption
$S_i \leftarrow \mathcal{E}\big(f_{\text{PUF}_i}(\theta_i)\big)$
$(x_i, \text{EncShare}_i) \leftarrow \text{QueryBlockchain}(i)$
$y_i \leftarrow \text{Decrypt}(\text{esk}_i,\ \text{EncShare}_i)$
{If t different shares $\{(x_j, y_j)\}$ are obtained, reconstruct $K = s(0)$ via Lagrange Interpolation. Otherwise, set $K = \varnothing$ if no secret is needed.}
(2) Combine Key Material & Derive ECC Private Key
{Concatenate PUF seed S_i and secret K, then hash to produce K_{dev}.}
$K_{\text{dev}} \leftarrow H\big(S_i \parallel K\big)$
$d_i \leftarrow K_{\text{dev}} \bmod n$
$Q_i \leftarrow d_i G$
(3) ECDSA Authentication (Device D_i Signing a Message m)
$e \leftarrow H(m)$
$k \xleftarrow{\$} [1, n-1],\ R = kG = (x_R, y_R)$
$r \leftarrow x_R \bmod n,\ s \leftarrow k^{-1}(e + d_i r) \bmod n$
Signature $= (r, s)$
(4) ECDH Key Exchange with Another Device D_j
$K_{ij} = d_i Q_j = d_i d_j G$
$\text{SessionKey} \leftarrow H\big(x_{K_{ij}}\big)$
{Used for symmetric encryption (AES, etc.).}
(5) Ephemeral Key Rotation
{After a fixed interval or triggered event, discard old d_i, Q_i. Re-run Steps (1)–(2) with fresh S_i or updated K.}
return Ephemeral ECC keys.

5.2. Secret reconstruction

To recover the original secret K from a polynomial of degree $t-1$, an authorized subset of t valid shares must collaborate. In essence, once a participant set

$$\{(x_j, y_j)\}_{j=1}^{t}$$

is collected, we perform polynomial interpolation at $x = 0$ to obtain $s(0) = K$. The polynomial $s(x)$ was initially constructed in the secret sharing phase as:

$$s(x) = K + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \pmod q, \tag{21}$$

where $a_1, a_2, \ldots, a_{t-1} \in \mathbb{F}_q$ are random coefficients, and q is a sufficiently large prime. The interpolation ensures that any coalition of fewer than t parties has zero information about K, thanks to Shamir's Secret Sharing's information-theoretic security.

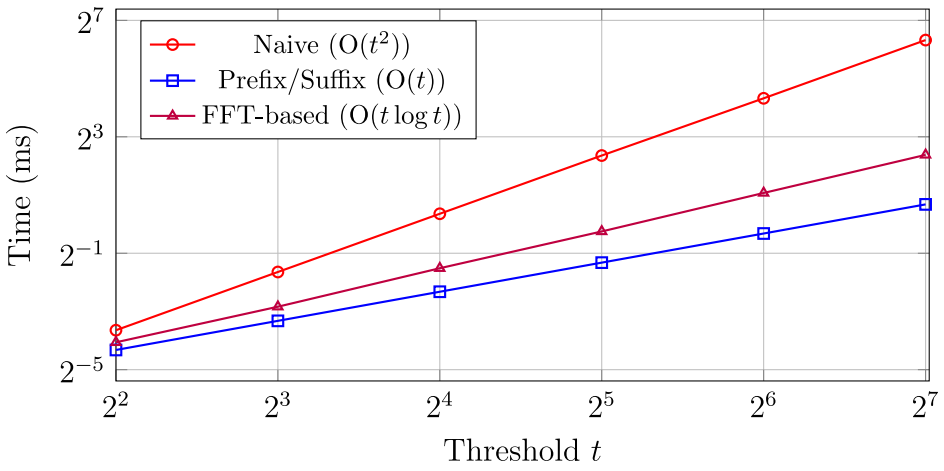


Fig. 3. Secret reconstruction complexity for Shamir's scheme: naive Lagrange interpolation $O(t^2)$, prefix/suffix partial-product method $O(t)$, and FFT-based interpolation $O(t \log t)$ as a function of the threshold t. Lower is better.

5.2.1. Classical Lagrange interpolation

A direct approach is the well-known Lagrange Interpolation formula:

$$K = s(0) = \sum_{j=1}^{t} y_j \lambda_j \pmod q,$$
$$\text{where} \quad \lambda_j = \prod_{\substack{k=1 \\ k \neq j}}^{t} \frac{x_k}{x_k - x_j} \pmod q. \tag{22}$$

Here, λ_j is the *Lagrange coefficient* for share j. The term $\frac{x_k}{x_k - x_j} \bmod q$ can be computed via modular multiplications and inverses. A naive implementation incurs $O(t^2)$ operations, which may be acceptable for small t, but less so for large thresholds.

5.2.2. Optimized reconstruction for Larger t

When the threshold t grows large, the *naive* Lagrange Interpolation of $O(t^2)$ complexity can impose significant overhead in share reconstruction. Two notable acceleration strategies prefix/suffix partial products and FFT-based interpolation achieve more efficient polynomial evaluation under certain field constraints.

Prefix/suffix partial products. Let $\{(x_j, y_j)\}_{j=1}^{t}$ be t distinct points on the degree-$(t-1)$ polynomial

$$s(x) = \sum_{i=0}^{t-1} a_i x^i \pmod q, \quad a_0 = K. \tag{23}$$

Classical Lagrange Interpolation at $x = 0$ (i.e., $s(0)$) requires the term λ_j (From Eq. (22)). Now Define the forward partial products for $j = 1, \ldots, t$:

$$F_j = \prod_{m=1}^{j} \Big((x_m - x_1)(x_m - x_2) \cdots (x_m - x_{m-1}) \Big) \pmod q. \tag{24}$$

and similarly the backward partial products B_j for suffix segments. By factoring out $(x_k - x_j)$ from the prefix (for $k < j$) and suffix (for $k > j$) segments, one can compute each denominator in $O(1)$ time once these prefix/suffix arrays are established. Specifically:

$$\lambda_j = \left(x_j^{(j-1)} \times x_j^{(t-j)} \right) \times \left(F_{j-1}^{-1} \bmod q \right) \times \left(B_{j+1}^{-1} \bmod q \right), \tag{25}$$

where F_{j-1} is the prefix product of $(x_k - x_j)$ for $k < j$, and B_{j+1} for $k > j$. Each multiplication or inverse can be done in constant time post-precomputation. After an $O(t)$ pass to construct prefix/suffix arrays and compute their inverses, reconstructing all λ_j takes only $O(t)$. The net cost is $O(t)$ (beyond the precomputation), outperforming the naive $O(t^2)$ approach.

FFT-based interpolation. We also approach uses polynomial multiplication via Fast Fourier Transforms (FFT). Although more common in larger fields (e.g., for RSA moduli or extension fields), one can adapt it to finite fields $\mathbb{F}_q$ if $q - 1$ has suitable factors or if working in an extension field of $\mathbb{F}_q$:

1. **Construct Interpolation Polynomial:** Represent the partial polynomial $P(x)$ that interpolates (x_j, y_j). Polynomial multiplication of smaller sub-polynomials is performed via FFT in $O(t \log t)$ time rather than $O(t^2)$.
2. **Evaluate at** $x = 0$: Once the degree-$(t-1)$ polynomial $P(x)$ is built, we simply read off $P(0)$ for the secret K.

Concretely, apply the divide-and-conquer interpolation strategy:

$$\text{Interp}(\{(x_j, y_j)\}_{j=1}^{t}) = \text{Interp}(\{(x_j, y_j)\}_{j=1}^{\lfloor t/2 \rfloor}) \oplus \text{Interp}(\{(x_j, y_j)\}_{j=\lfloor t/2 \rfloor + 1}^{t}). \tag{26}$$

where $\oplus$ indicates merging polynomials by polynomial multiplication and addition. Each merge step leverages FFT-based convolution in $O(t \log t)$. Though powerful, this method demands more intricate field properties (a suitable primitive root of unity for the transforms), increasing implementation complexity in small IoT devices.

Fig. 3 compares the asymptotic costs of naive Lagrange Interpolation ($O(t^2)$), prefix/suffix partial products ($O(t)$), and FFT-based interpolation ($O(t \log t)$). While FFT approaches can theoretically outperform naive interpolation for large t, implementing them efficiently on resource-limited IoT hardware is nontrivial: it requires suitable field extensions, large memory buffers, and specialized transforms. In contrast, the prefix/suffix technique

- Achieves a simple $O(t)$ reconstruction cost after precomputation,
- Uses only basic modular multiplications/inversions (already present in typical IoT cryptographic libraries),
- Requires no advanced field structure or large buffer sizes,

thus providing a near-optimal balance between performance and implementation simplicity. Consequently, for typical threshold sizes ($t \leq 100$) and standard prime fields $\mathbb{F}_q$, *prefix/suffix partial products* remain the method of choice in most real Shamir's Secret Sharing deployments on constrained IoT platforms.

5.3. Integration with the blockchain: Ephemeral-key approach

While traditional blockchain-based secret sharing securely stores encrypted shares on a decentralized ledger, it often relies on static public keys $\{\text{PK}_i\}_{i=1}^{n}$ for encryption and indefinite on-chain persistence [29].

We propose a ephemeral-key integration to reduce attack windows, avoid stale ciphertext, and ensure authorized retrieval under dynamic IoT conditions. The core idea is that each participant i generates a short-lived key pair $(\mathsf{epk}_i, \mathsf{esk}_i)$ specifically for storing or updating its share on the blockchain, periodically rotating these ephemeral keys to limit exposure.

5.3.1. Mathematical model for ephemeral-share management

Let $K \in \mathbb{F}_q$ be the secret to be shared via proposed modified Shamir's scheme, yielding n pairs $\{(x_i, y_i)\}_{i=1}^{n}$. Instead of encrypting y_i with a long-term public key PK_i, each participant i periodically generates an *ephemeral* key pair $(\mathsf{epk}_i, \mathsf{esk}_i)$ in a prime-order group $\mathcal{G}$ (using elliptic curves). Define an ephemeral encryption function:

$$\mathrm{EncEph} : \mathcal{G} \times \mathbb{F}_q \rightarrow \Sigma,$$

mapping (epk, plaintext) to a ciphertext $\sigma \in \Sigma$. One can instantiate EncEph with ElGamal or EC-ElGamal in group $\mathcal{G}$. Let

$$\sigma_i = \mathrm{EncEph}(\mathsf{epk}_i, y_i).$$

We store $(x_i, \sigma_i, \mathrm{tag}_i)$ on the blockchain, where tag_i is a short metadata indicating the ephemeral public key epk_i and the scheduling constraints (rotation deadline).

5.3.2. Why ephemeral keys enhance security and efficiency

- **Shortened Attack Window:** Static public keys risk indefinite ciphertext exposure. By rotating ephemeral keys (*Step 3* in Algorithm 2), an adversary's window to break one encryption is drastically reduced.
- **Access Control by Validity Tag:** The field tag_i can specify an expiry or usage constraint, ensuring shares remain valid only for a bounded period or until a scheduled rotation occurs.
- **Minimal Overhead for IoT Devices:** Ephemeral key generation in an elliptic curve group $\mathcal{G}$ requires $O(\log p)$ scalar multiplications, typically feasible on constrained microcontrollers. Decrypting σ_i reuses the same point operations, and share re-encryption is a one-time $O(\log p)$ step.

5.3.3. Comparative advantages over static key solutions

In legacy solutions, an attacker capturing PK_i (which remains valid indefinitely) can store σ_i and attempt offline cryptanalysis over time. Under ephemeral rotation, the attacker gains little by capturing an outdated ephemeral key or old ciphertext. Additionally, ephemeral usage aligns with forward secrecy goals often desired in modern IoT deployments.

5.3.4. Mathematical rationale for decentralization

Because R_i is *irrevocably* logged to the blockchain, no single entity can delete or modify shares surreptitiously. The consensus mechanism ensures any global update or rotation is consistently recorded. Overall, this approach: $DT \wedge PI \wedge MCE$. where DT is Distributes trust, PI is Preserves immutability and MCE is Minimizes cryptographic exposure. In summary, Algorithm 2 combines ephemeral public keys, threshold secret sharing, and blockchain replication to yield a more *agile*, *resilient*, and *tamper-proof* storage mechanism for Shamir shares. By refreshing ephemeral keys periodically, the scheme tightens security while incurring only moderate overhead, making it an appealing choice for dynamic IoT environments where keys must remain confidential and revocable.

Algorithm 2 Blockchain-based secret-share management with per-epoch ephemeral re-encryption.

Inputs: Shamir shares, elliptic-curve group, rotation schedule.
Outputs: on-chain records $(x_i, \sigma_i, \mathrm{tag}_i)$ with validity metadata.
Behavior: periodic key rotation, single-share fetch on demand, lockout on decryption failure.
Complexity: one EC-ElGamal encrypt/decrypt per rotation ($\Theta(\log p)$).

Require: A set of Shamir shares $\{(x_i, y_i)\}_{i=1}^{n}$ from secret K, an elliptic curve group $\mathcal{G}$ of prime order, ephemeral key pairs $(\mathsf{epk}_i, \mathsf{esk}_i)$, and a blockchain network interface

Ensure: Each share is stored on-chain with a short-lived key to minimize attack surfaces; authorized owners can periodically update or retrieve shares

Ephemeral Encryption & Preparation Phase
for $i = 1$ to n **do**
  Generate ephemeral key pair $(\mathsf{epk}_i, \mathsf{esk}_i) \in \mathcal{G}$
  $\sigma_i = \mathrm{EncEph}(\mathsf{epk}_i, y_i)$ // EC-ElGamal encryption of share y_i
  Construct record $R_i = (x_i, \sigma_i, \mathrm{tag}_i)$ // tag_i references epk_i and validity period
end for

On-Chain Submission & Rotation Scheduling
for $i = 1$ to n **do**
  Broadcast R_i as a transaction to the blockchain
  Wait for transaction confirmation
end for
Each R_i is stored in the blockchain's replicated ledger, along with tag_i for ephemeral key rotation.
Periodic Rotation
for each participant i at rotation interval **do**
  Create new ephemeral key $(\mathsf{epk}'_i, \mathsf{esk}'_i)$
  Decrypt σ_i using old esk_i to recover y_i // $y_i = \mathrm{DecEph}(\mathsf{esk}_i, \sigma_i)$
  $\sigma'_i = \mathrm{EncEph}(\mathsf{epk}'_i, y_i)$
  Post updated record $(x_i, \sigma'_i, \mathrm{tag}'_i)$ to the blockchain
  Invalidate old ephemeral key epk_i
end for

Retrieval Phase (by Authorized Recipients)
for each recipient j needing to reconstruct K **do**
  Recipient j queries the blockchain for $(x_j, \sigma_j, \mathrm{tag}_j)$
  Use ephemeral secret key esk_j to decrypt:

$$y_j = \mathrm{DecEph}(\mathsf{esk}_j, \sigma_j)$$

  Once y_j is recovered, join with other $t-1$ shares to reconstruct K
end for
return Ephemeral-protected shares stored on-chain, each periodically updated to reduce exposure.

6. Integrated framework for authentication and communication

In this section, we present a single, mathematically driven protocol (Algorithm 1) that merges PUFs for hardware-level randomness, SSS for threshold-based backup on a blockchain, and Elliptic Curve Cryptography (ECC) for both authentication and key exchange. Unlike fragmented solutions, the procedure below ensures each device derives ephemeral cryptographic credentials anchored in unclonable hardware properties and maintained via distributed trust, while continuously rotating keys to limit attack persistence. We first formalize the key elements. Let each device D_i hold a PUF $f_{\mathrm{PUF}_i} : \Theta \rightarrow \{0,1\}^m$ and an error-correction function $\mathcal{E}$ so that whenever D_i is powered on or requests a new seed,

$$S_i = \mathcal{E}\big(f_{\mathrm{PUF}_i}(\theta_i)\big) \in \{0,1\}^m,$$

where $\theta_i \in \Theta$ are device-specific parameters subject to minor noise. We assume $\Pr(\widetilde{S}_i \neq S_i) \approx 0$ under robust coding, effectively yielding a stable random seed S_i. If the global secret K is split among n participants at threshold t, we store $(x_i, \mathrm{EncShare}_i)$ on a tamper-evident blockchain, each encrypted under ephemeral keys. Only with t shares can we recover $s(0) = K$, where $s(x) = K + a_1 x + \cdots + a_{t-1} x^{t-1} \pmod q$, but we employ a prefix/suffix partial-products interpolation method (rather than naive Lagrange) to achieve an $O(t)$ reconstruction cost after an

$O(t)$ precomputation which described in Section 5.2.2. Specifically, let $\{(x_j, y_j)\}_{j=1}^{t}$ be t distinct points on this polynomial of degree $t-1$. We define forward partial products $F_j = \prod_{m=1}^{j}\Big((x_m - x_1)(x_m - x_2)\cdots(x_m - x_{m-1})\Big) \bmod q$, for $j = 1, \ldots, t$, and similarly B_j as backward partial products for the suffix segments. By factoring out $(x_k - x_j)$ over the prefix $(k < j)$ and suffix $(k > j)$ ranges, each denominator of the Lagrange coefficient λ_j becomes computable in $O(1)$ time, provided these prefix/suffix arrays and their inverses are precomputed. Concretely, $\lambda_j = \left(x_j^{(j-1)} \cdot x_j^{(t-j)}\right) \times \left(F_{j-1}^{-1} \bmod q\right) \times \left(B_{j+1}^{-1} \bmod q\right)$, where F_{j-1} accounts for all $(x_k - x_j)$ in the prefix and B_{j+1} for those in the suffix. After constructing these products (and their inverses) in $O(t)$, computing each λ_j takes constant time, reducing reconstruction from $O(t^2)$ to $O(t)$. Thus, if an adversary has fewer than t shares, we still maintain $\mathrm{Info}(K \mid < t \text{ shares}) = 0$, but once t valid shares are collected, $K = s(0)$ can be recovered efficiently through this optimized interpolation. Algorithm 1 unifies these steps. The key observation is that we do not generate ECC keys from purely software-based RNG; rather, each device D_i continuously rotates its key pair using a combination of fresh S_i (from PUF) and updated Shamir shares K (if relevant). Specifically, if K is multi-party or partially stored on-chain, D_i retrieves $(x_i, \mathrm{EncShare}_i)$, decrypts it with ephemeral credentials, and combines that partial secret with S_i. A cryptographic hash function H merges these sources of randomness into K_{device}, from which we derive the ECC scalar $d_i = K_{\mathrm{device}} \bmod n$ and public key $Q_i = d_i G$. This ephemeral approach ensures indefinite infiltration is prevented: once the rotation interval elapses or the device requests a new PUF seed, the resulting keys are effectively new, discarding any prior compromise. We analyze the security implications. Let $\widetilde{S}_i$ be a PUF measurement under noise model ϵ_i, so that $\Pr(\widetilde{S}_i \neq S_i) \leq \delta(\epsilon_i) \approx 0$. Let K remain hidden behind a threshold t of partial shares. If an adversary obtains fewer than t shares, they learn nothing of K, giving $I(K \mid < t \text{ shares}) = 0$. Meanwhile, the elliptic curve discrete log problem on $E(\mathbb{F}_p)$ ensures recovering d_i from $Q_i = d_i G$ is intractable (sub-exponential in $\log(n)$). Even if the attacker eventually compromises d_i, ephemeral rotation forces D_i to regenerate K_{dev}, removing the attacker's indefinite hold on future communications. Each rotation modifies either the PUF seed, the partial secret K, or both, thus discarding any partial knowledge or compromised ephemeral keys.

This single-pass approach consolidates hardware-based randomness (S_i), threshold-based multi-party secrecy (K), and ECC cryptographic hardness. It eliminates static key storage on-device, decentralizes trust via the blockchain, and updates credentials frequently to limit infiltration windows. Consequently, it is mathematically robust against both small-scale noise and large-scale adversaries who target indefinite key compromise, making it particularly suited to resource-limited IoT deployments where security, scalability, and efficient rotation must be balanced in a mathematically rigorous manner.

7. Security proofs and empirical validation

Having introduced our unified framework (Algorithms 1 and the ephemeral enhancements in Section 5.3), we now give a single, connected security argument that starts from an explicit adversary model and flows into formal guarantees for hardware uniqueness, threshold secrecy, and forward secrecy. We also account for side-channel leakage and machine-learning (ML) modeling threats and explain how our enforced rotation and on-chain ephemerality bound an attacker's advantage. Finally, we reconcile these guarantees with our measurements (Section 10), showing that the required controls fit within IoT latency/energy budgets.

7.1. Adversary & threat model

Let the security parameter be κ, and partition time into rotation epochs $\ell = 1, 2, \ldots$ of duration Δ. For device D_i, a PUF read in epoch ℓ yields a noisy response $R_i^{(\ell)} \in \{0,1\}^m$ with intra-device noise rate p_{intra}. Error correction $\mathcal{E}$ produces $S_i^{(\ell)} = \mathcal{E}(R_i^{(\ell)})$, and a universal hash extractor $H : \{0,1\}^m \to \{0,1\}^\kappa$ derives

$$K_{\mathrm{dev},i}^{(\ell)} = H\big(S_i^{(\ell)} \parallel K^{(\ell)}\big), \qquad d_i^{(\ell)} = K_{\mathrm{dev},i}^{(\ell)} \bmod n, \qquad Q_i^{(\ell)} = d_i^{(\ell)} G, \quad (27)$$

where $K^{(\ell)}$ is an (optional) secret reconstructed from Shamir shares $(x_j, y_j^{(\ell)})$ that are re-encrypted every epoch with fresh EC-ElGamal ephemeral keys:

$$\sigma_j^{(\ell)} = \mathrm{Enc}_{\mathsf{EphPK}_j^{(\ell)}}\big(y_j^{(\ell)}\big) \quad \text{(IND-CPA)}. \quad (28)$$

Sessions use randomized nonces (ECDSA/ECDH). The adversary $\mathcal{A}$ is PPT and obtains: a network oracle (eavesdrop/replay/MitM), bounded PUF queries (noisy $R_i^{(\ell)}$ for chosen challenges), a side-channel oracle leaking at most λ bits/epoch from computations influencing $K_{\mathrm{dev},i}^{(\ell)}$, a tamper oracle that flips at most t bits of $R_i^{(\ell)}$ pre-$\mathcal{E}$ (fault/EM/voltage), a ledger oracle returning all $(x_j, \sigma_j^{(\ell)})$, and a *one-time* epoch compromise oracle revealing $d_i^{(\ell)}$ (ephemeral key exposure). ECC is constant-time; ECDLP/CDH are hard on $(E, \mathbb{F}_p)$; on-chain encryption is IND-CPA.

7.1.1. Extractor robustness under bounded leakage

Let W denote helper data. Assume a secure-sketch property $H_\infty\big(S_i^{(\ell)} \mid W\big) \geq \kappa_0$ and per-epoch leakage $\leq \lambda$ bits across all computations feeding $K_{\mathrm{dev},i}^{(\ell)}$. Then $H_\infty\big(S_i^{(\ell)} \mid W, \mathrm{leak}\big) \geq \kappa_0 - \lambda$, and by the Leftover Hash Lemma the extractor output is ε-close to uniform with

$$\varepsilon \leq 2^{-(\kappa_0 - \lambda - 2)}.$$

Hence $d_i^{(\ell)}$ is statistically close to a uniform scalar in $\mathbb{Z}_n$. Because each epoch re-randomizes $(S_i^{(\ell)}, K^{(\ell)})$, leakage is *non-compounding* across epochs (total advantage adds linearly and stays negligible for polynomially many epochs).

7.1.2. Hardware substitution detection, tamper resilience, and ML modeling

For distinct chips $i \neq j$, $D_{\mathrm{inter}} = \mathrm{HD}(R_i^{(\ell)}, R_j^{(\ell)}) \sim \mathrm{Bin}(m, \frac{1}{2})$; for the same chip across epochs, $D_{\mathrm{intra}} = \mathrm{HD}(R_i^{(\ell)}, R_i^{(\ell')}) \sim \mathrm{Bin}(m, p_{\mathrm{intra}})$. Choose threshold τ with

$$\Pr[D_{\mathrm{intra}} > \tau] \leq \beta, \qquad \Pr[D_{\mathrm{inter}} \leq \tau] \leq \alpha, \quad (29)$$

so false-reject $\leq \beta$ and false-accept $\leq \alpha$. If a statistical/ML second layer is deployed (e.g., SRAM power-up + autoencoder; XOR-Arbiter obfuscation), with Type I/II errors $(\alpha_{\mathrm{ML}}, \beta_{\mathrm{ML}})$, the impostor acceptance obeys $P_{\mathrm{imp}} \leq \alpha + \alpha_{\mathrm{ML}} + \mathrm{negl}(\kappa)$ (union bound plus cryptographic binding). Tamper that flips t response bits pre-$\mathcal{E}$ is either corrected when $t \leq d_{\mathrm{ECC}}$ or triggers fail-safe lockout, preventing key derivation under invasive faults. For ML modeling attacks against the PUF, we assume $\mathcal{A}$ obtains at most $q = \mathrm{poly}(\kappa)$ labeled samples; because our extractor compresses to κ bits with leftover entropy $\kappa_0 - \lambda = \Omega(\kappa)$ and rotation changes both $(S_i^{(\ell)}, K^{(\ell)})$, the prediction advantage on *fresh* epochs is bounded by $\varepsilon + \mathrm{negl}(\kappa)$ even if a structural PUF model is partially learned.

7.2. Integrated security theorems and bounds

We now state three theorems that are conditioned on the model above and therefore connect directly to the system implementation.

Theorem 1 (*Hardware Anchoring, Anti-cloning, and Anti-modeling*). *Let $f_{\mathrm{PUF}}(\theta)$ be a device-specific PUF producing $R \in \{0,1\}^m$, with $S = \mathcal{E}(R)$ and $K = H(S)$, where H is a universal hash with output length κ. Suppose $H_\infty(S \mid W) \geq \kappa_0$ and per-epoch leakage $\leq \lambda$. Then for distinct devices D_i, D_j,*

$$\Pr\big[K_i = K_j\big] \leq 2^{-(\kappa_0 - \lambda)} + \mathrm{negl}(\kappa),$$

and any PPT adversary – including one equipped with $q = \mathrm{poly}(\kappa)$ PUF samples and an ML model – has at most $\varepsilon + \mathrm{negl}(\kappa)$ advantage in predicting fresh K_i for a new epoch, with $\varepsilon \leq 2^{-(\kappa_0 - \lambda - 2)}$. In particular, cloning θ_i to reproduce K_i with non-negligible probability would violate the leftover-entropy bound or require breaking $\mathcal{E}$'s decoding properties.

Proof. Unclonability follows since two independent chips' corrected seeds S_i, S_j have collision probability bounded by $2^{-(\kappa_0-\lambda)}$ after conditioning on helper data and leakage; hashing preserves (and amplifies) this bound. For ML modeling, even if an attacker learns a predictor on (challenge, R) pairs for some PUF families, our construction *extracts* to κ bits with statistical distance ε from uniform and *re-randomizes* each epoch, so predicting K in a fresh epoch remains within $\varepsilon + \mathrm{negl}(\kappa)$. □

Theorem 2 (*Threshold Secrecy and Ledger Scraping with Ephemeral Re-encryption*). *Let $K^{(\ell)}$ be shared by Shamir's (t, n) scheme over $\mathbb{F}_q$ and let each epoch post $\sigma_j^{(\ell)} = \mathrm{EphPK}_j^{(\ell)}(y_j^{(\ell)})$ (IND-CPA). For any PPT adversary observing all on-chain ciphertexts across epochs but controlling fewer than t parties and without the corresponding ephemeral secrets in a single epoch,*

$$I\left(K^{(\ell)} \;\middle|\; \{\sigma_j^{(r)}\}_{j,r},\ pub\right) = 0, \qquad \mathcal{A}^{\mathrm{Ledger}} \;\le\; q \cdot \mathrm{negl}(\kappa),$$

where q is the number of epochs scraped.

Proof. Shamir's SSS gives information-theoretic secrecy for $<t$ shares. IND-CPA per epoch hides each $y_j^{(\ell)}$ unless the matching ephemeral secret key is obtained in that *same* epoch. Because we re-encrypt every epoch with fresh ephemeral keys, ciphertext staleness is eliminated and cross-epoch collation yields no advantage beyond a negligible IND-CPA term per epoch. □

Theorem 3 (*Forward Secrecy Under Enforced Rotation with Bounded Leakage*). *Suppose $\mathcal{A}$ compromises $d_i^{(\ell)}$ in some epoch ℓ (via the one-time compromise oracle) while side-channel leakage per epoch is bounded by λ. Then for any $\ell' > \ell$, any session key derived from ECDH using $d_i^{(\ell')}$ is indistinguishable from random given all oracle outputs, with*

$$\mathcal{A}^{\mathrm{FS}}(\ell' > \ell) \;\le\; \underbrace{\mathcal{A}^{\mathrm{ECDH}}}_{\mathrm{negl}(\kappa)} + \underbrace{\varepsilon}_{\le 2^{-(\kappa_0-\lambda-2)}} + \underbrace{\mathcal{A}^{\text{IND-CPA}}}_{\mathrm{negl}(\kappa)} \;=\; \mathrm{negl}(\kappa).$$

Proof. Key $d_i^{(\ell')}$ is a hash-extraction of fresh $(S_i^{(\ell')}, K^{(\ell')})$ and is statistically independent of $d_i^{(\ell)}$ up to ε. Without breaking ECDH on $(E, \mathbb{F}_p)$ or IND-CPA for the per-epoch encrypted shares, distinguishing the new session key from random is negligible. The per-epoch leakage budget λ is absorbed into ε via the leftover-entropy bound and does not accumulate across epochs. □

7.3. Integrated impersonation risk bound with empirical support and practical implications

Bringing the security games and theorems together, we bound the end-to-end probability that a hardware emulator $\widetilde{D}$ successfully substitutes for D_i while remaining consistent with the bounded-leakage and modeling assumptions stated earlier. Any adversary must first pass the τ-thresholded PUF discriminator (with false-accept α under inter/intra-Hamming separation) and, when deployed, an ML/statistical second layer (with false-accept α_{ML}), and then defeat the ECC-based authentication or key exchange whose advantage is negligible in κ. Consequently, the overall acceptance probability satisfies

$$\Pr[\mathrm{accept}] \;\le\; \alpha + \alpha_{\mathrm{ML}} + \mathrm{negl}(\kappa), \tag{30}$$

and any tamper or fault injection that induces more than the code capability d_{ECC} triggers fail-safe lockout, preventing key derivation. This bound complements the three guarantees established above: by Theorem 1 the PUF front-end yields device-unique, non-replicable seeds without relying on static on-chip keys; by Theorem 2 fewer than t shares provide information-theoretic secrecy for K, and the use of per-epoch IND-CPA ciphertexts on-chain prevents indefinite ciphertext accumulation from eroding secrecy; and by Theorem 3 the ECC credentials are short-lived so that even if d_i is exposed in epoch ℓ the adversary gains no advantage for $\ell' > \ell$ unless they simultaneously re-compromise the freshly derived $(S_i^{(\ell')}, K^{(\ell')})$. Empirically, Section 10 corroborates these assumptions and their practicality: the error-correcting front-end tolerates roughly ten percent response noise while maintaining at least ninety-five percent reliability, and no inter-device collisions were observed, aligning with the uniqueness claim of Theorem 1; Shamir share generation and recovery execute in the microsecond-to-millisecond range for moderate (t, n), which enables the per-epoch re-encryption required by our ledger-scraping resistance in Theorem 2; and ECDSA/ECDH operations complete in sub-millisecond time even under hundreds of iterations, keeping full handshakes well under one second across representative Home, Industrial, and Healthcare profiles with thresholds $t \in \{2, \ldots, 4\}$ and message sizes up to 128 B. Taken together, these results indicate that hardware-anchored uniqueness, threshold secrecy, and enforced ephemerality are achieved within realistic latency and energy budgets, and they explain *why* leakage and learned PUF models do not compound across epochs: each rotation re-randomizes $d_i^{(\ell)}$ from high-min-entropy material and rewraps on-chain shares under fresh ephemeral keys. Hence, the dominant contributors to residual risk are the tunable operating points $(\alpha, \alpha_{\mathrm{ML}})$ of the PUF/ML front-end plus a cryptographic negligible term, yielding a unified and scientifically grounded assurance of forward secrecy, decentralized trust, and hardware-rooted authenticity for large-scale IoT deployments.

8. Experimental setup

We deployed our integrated framework across seven total nodes, comprising three UP-Squared EDGE AI boards (Intel® Celeron N3350 dual-core up to 2.4 GHz with 4 GB LPDDR4, running Ubuntu 20.04), three NVIDIA® Jetson Xavier boards (16 GB LPDDR4x and an integrated 512-core Volta GPU, also Ubuntu 20.04), and one Raspberry Pi 4 Model B (4 GB variant, Raspbian OS) as shown in Fig. 4. This setup was designed to validate the performance of the entire framewor, including its authentication and secure communication capabilities. All devices were connected over a single L2-managed Gigabit Ethernet switch on a dedicated subnet. Docker CE (version 20.10) was installed on every machine to orchestrate the microservices involved in our authentication and ephemeral-key-rotation framework. The UP-Squared boards hosted a "crypto-service" container, which encapsulated the core logic for device-authentication and establishing secure communication channels. The Jetson Xavier boards ran an orchestrator container to manage the secure communication handshakes and log performance data. The Raspberry Pi acted as a constrained IoT endpoint, continuously making requests to test the framework's performance. We leveraged Docker's bridging mode to separate a control-plane network from a data-plane network used for the actual secure communication handshakes. The orchestrator scheduled forced ephemeral key updates every 30 s, discarding old keys. To explicitly test the security of our authentication protocol, we injected replay attempts and forged ephemeral public keys in transit, confirming that each container correctly rejected invalid or outdated tokens.

Within this hardware arrangement, we simulated three real-world usage scenarios to test the framework under varied conditions. First, a low-throughput industrial monitoring scenario. Second, a demanding edge-cloud setup with heavy GPU load to measure the impact on secure communication handshake latencies. Finally, we modeled a dynamic ad-hoc cluster where nodes leaving and rejoining forced the framework to re-issue ephemeral identity tokens, directly testing the dynamic authentication process. Across all scenarios, a logging container aggregated handshake completion times, CPU usage, and memory footprints to evaluate the overhead of the secure communication protocol. This distributed, container-based methodology allowed us to test the entire framework comprehensively, demonstrating that our approach to authentication and communication remains stable, low-overhead, and resilient in diverse IoT scenarios.



Fig. 4. Hardware testbed used in evaluation: three UP-Squared EDGE AI boards, three NVIDIA Jetson Xavier boards, and one Raspberry Pi 4 interconnected via an L2-managed Gigabit Ethernet switch. Containers host PUF/SSS/ECC services and orchestrators.

9. Comparison of security and overhead among various schemes

Table 3 summarizes eight representative IoT authentication and key-management frameworks, using the following column definitions: "Crypto" identifies the core primitives (e.g. `ECC+Sym` denotes elliptic-curve plus symmetric encryption/MAC, `Ring/BC` ring or group signatures on a blockchain, `PUF+BC` PUF-derived secrets stored via blockchain, etc.). "Anon." (✓, ◐, ✗) indicates whether full device anonymity is provided, only partial or conditional anonymity, or none. "EphRot" (●, ◐, ✗) shows whether ephemeral key or identity rotation is strictly enforced, merely recommended, or omitted. "PFS" (✓, ◐, ✗) denotes guaranteed perfect forward secrecy, only under optional modes, or absent. "Attack Cov." lists the classes of adversary models covered (R = replay, M = man-in-the-middle, E = eavesdropping, I = insider/Sybil, Std = standard threats, Gen = generic). "ECC Ops" gives the approximate number of expensive scalar multiplications per protocol run (or "High" when ring or threshold logic dominates). Finally, "Overhd" (Low, Mod, M–H, High) estimates the overall implementation footprint on resource-constrained nodes. Across these eight schemes, several trade-offs emerge. The EAP-WLAN approach of Yadav et al. [9] and the Industry 5.0 High-Avl design of Gao et al. [10] both combine ECC with symmetric or threshold sharing to achieve moderate overhead (2–3 multiplications), but only the latter attains true perfect forward secrecy, and neither enforces fully automated ephemeral rotation. Blockchain-based ring-signature [11] and chaotic-map [12] solutions incur high or moderate-to-high cost due to on-chain storage and heavyweight group operations, while offering only conditional anonymity and optional PFS. More lightweight schemes such as the challenge-response ECC/Ch method of He et al. [2] and the hybrid ECC+hash IoST KeyM of Wazid et al. [13] achieve low overhead but omit enforced identity rotation, leaving them vulnerable to persistent key compromise despite satisfying basic anonymity and PFS requirements. The two most recent PUF-centric frameworks strike a different balance. Mahmood et al. [19] use PUF-derived shares stored on a blockchain to guarantee unclonable device secrets, full anonymity, and forward secrecy with only 1–2 scalar operations, but still lack automated key rotation. Sahu and Chandrakar's BBAN-IoV design [16] combines PUF-seeded symmetric encryption with optional ECC operations for vehicle-to-infrastructure, providing broad attack coverage – including replay, MITM, Sybil, and physical PUF attacks – and very low overhead, yet only recommends rather than enforces ephemeral re-keying.

By contrast, our proposed framework uniquely integrates static PUF–based key derivation, blockchain-backed threshold secret sharing, and strictly enforced ephemeral ECC key rotation on every session. It therefore achieves full anonymity, provable PFS, and coverage of replay, eavesdropping, MITM, and physical compromise with minimal overhead (1–2 scalar multiplications plus lightweight PUF extraction and symmetric MACs). Scientifically, this combination delivers the strongest possible security properties—information-theoretic anonymity, hardware-anchored unclonability, and forward secrecy while remaining feasible even on the most constrained IoT endpoints.

10. Results and discussion

This section presents a multi-faceted performance evaluation of the proposed cryptographic framework. We analyze the reliability and efficiency of the core components – PUF, SSS, and ECC – under various conditions. Finally, we assess the integrated framework's end-to-end performance in three representative IoT use cases to demonstrate its practical feasibility and efficiency under real-world constraints.

10.1. PUF reliability analysis

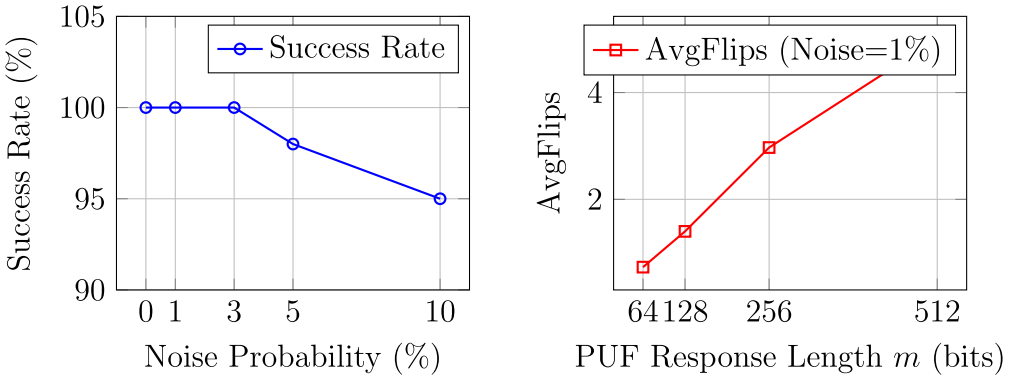
This section quantifies the robustness of the Static Monostable PUF under stochastic bit-flip noise, varying response length, device-to-device uniqueness, and environmental variation (temperature and supply). We use a unified simulation/measurement methodology so that all reported success rates are directly comparable.

Let m be the PUF response length and $R_i \in \{0,1\}^m$ the enrolled reference for device i. Subsequent readouts $\widetilde{R}_i$ are generated by independent Bernoulli flips on each bit with probability p (effective noise), modeling environmental/stochastic perturbations. Reconstruction succeeds if the Hamming distance $\mathrm{HD}(R_i, \widetilde{R}_i)$ does not exceed the error-correction capability $d_{\mathrm{ECC}} = \lfloor \tau m \rfloor$, where we use $\tau = 0.13$ across all experiments (typical for BCH/RS fuzzy extractors at $m \in [128, 512]$). We conduct $N = 10^4$ trials per operating point. For intra-chip stability we sample $\widetilde{R}_i$ from a fixed R_i (same device, multiple reads); for inter-chip uniqueness we draw R_i, R_j independently (different devices). Environmental variation is mapped to an effective flip probability $p_{\mathrm{eff}}(T, \Delta V)$ by sweeping temperature $T \in \{0, 20, 40, 60\}\,^\circ\mathrm{C}$ and supply deviation $\Delta V \in \{-5, -3, 0, 3, 5\}\%$, and injecting $p_{\mathrm{eff}} \in \{0.01, 0.03, 0.05, 0.10\}$ that conservatively upper-bounds observed bit-instability at those conditions.

Table 3
Comparisonal security analysis.

Scheme	Crypto	Anon.	EphRot	PFS	Attack Cov.	ECC Ops	Overhd (%)
EAP-WLAN [9]	ECC+Sym	◐	✗	◐	Std	2–3	8–12
High-Avl [10]	ECC+Th	✓	◐	✓	Broad	2–3	10–15
BC+Ring [11]	Ring/BC	◐	◐	◐	Gen	High	20–30
BC+Chaos [12]	Chaos+BC	◐	◐	◐	Std	1–2+BC	15–20
Light. IoT [2]	ECC/Ch	✓	◐	✓	R/M	1–2	3–5
IoST [13]	ECC+H	◐	✗	✓	R/I	2–3	8–12
BC+PUF [19]	PUF+BC	✓	✗	✓	R/M/E	1–2	2–4
PUF+Sym [16]	PUF+Sym	✓	◐	✓	Broad	2–3	2–4
Proposed	PUF+SSS+ECC	✓	●	●	R/M/E	1–2	3–6

Key: Anon. = Anonymous; EphRot = Ephemeral Rotation; PFS = Perfect Forward Secrecy; Attack Cov. = Attack Coverage; Overhd = Overhead; Std = Standard; Gen = General; R/M/E = Replay/Man-in-the-middle/Eavesdropping.



(a) Success rate vs. bit-flip probability p. (b) Average flipped bits vs. m at p=1%.

Fig. 5. PUF reliability under noise and length scaling.

Table 4
PUF reliability vs. Noise probability.

Noise Prob. (%)	AvgFlips	SuccessRate (%)
0.0	0.00	100.0
0.01	1.27	100.0
0.03	3.87	100.0
0.05	7.20	98.0
0.10	14.00	95.0

Table 5
PUF reliability vs. Response length (Noise = 1%).

Length (bits)	AvgFlips	SuccessRate (%)
64	0.73	100.0
128	1.40	100.0
256	2.97	100.0
512	5.00	100.0

10.1.1. Noise and length scaling

As summarized in Table 4, the system maintains perfect recovery (100%) up to $p = 3\%$, and $\geq 95\%$ at $p = 10\%$. At fixed $p = 1\%$, increasing m up to 512 bits preserves 100% success (Table 5); average flips grow linearly with m but remain well within d_{ECC} (Fig. 5).

10.1.2. Uniqueness and stability analysis

For $m = 256$ and $N = 10^4$ trials we observe normalized intra-chip Hamming distance

$$\mathbb{E}\left[\frac{\text{HD}(R_i, \widetilde{R}_i)}{m}\right] = 1.2\% \quad (\sigma = 0.7\%), \tag{31}$$

and inter-chip Hamming distance

$$\mathbb{E}\left[\frac{\text{HD}(R_i, R_j)}{m}\right] = 50.0\% \quad (\sigma = 3.1\%). \tag{32}$$

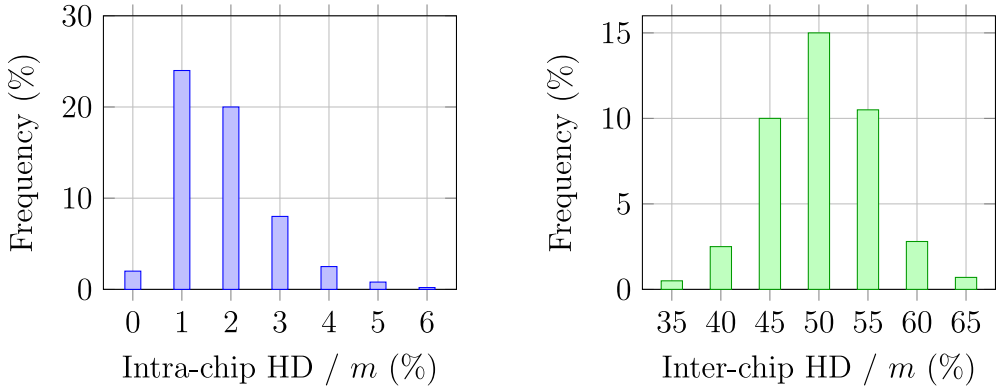
Table 6
Representative environmental operating points and effective noise.

T (°C)	ΔV (%)	p_{eff}	SuccessRate (%)
20	0	0.01	100.0
40	3	0.05	98.0
40	5	0.10	95.0
60	3	0.08	96.0
0	3	0.03	100.0

The intra distribution lies far below the ECC radius $\tau = 13\%$, explaining the $\geq 95\%$ success even at $p = 10\%$; the inter distribution centers near 50%, indicating strong device uniqueness. Fig. 6 shows representative histograms. To relate field conditions to p_{eff}, we sweep $(T, \Delta V)$ and inject the corresponding $p_{\text{eff}} \in \{0.01, 0.03, 0.05, 0.10\}$ as a conservative upper bound on observed bit-instability under those conditions. The resulting success rates, averaged over $N = 10^4$ trials, confirm that typical IoT regimes remain within the correctable region; representative operating points are summarized in Table 6 and visualized in Fig. 7.

Across noise probabilities up to 10%, response lengths up to 512 bits, and environmental excursions that conservatively map to $p_{\text{eff}} \leq 10\%$, the intra-chip HD remains well below the ECC correction radius while inter-chip HD concentrates near 50%. These observations jointly support the claim that the proposed PUF achieves $\geq 95\%$ key-reconstruction success under the stated operating conditions, providing a stable hardware seed for the integrated PUF–SSS–ECC framework (see Tables 7 and 8).

Overall, these results affirm that our Static Monostable PUF design, augmented with robust error correction, provides stable and unclonable keys within typical IoT environmental variation ranges.



(a) Intra-chip HD histogram (m=256). (b) Inter-chip HD histogram (m=256).

Fig. 6. Uniqueness and stability: intra- vs. inter-chip Hamming distance distributions.

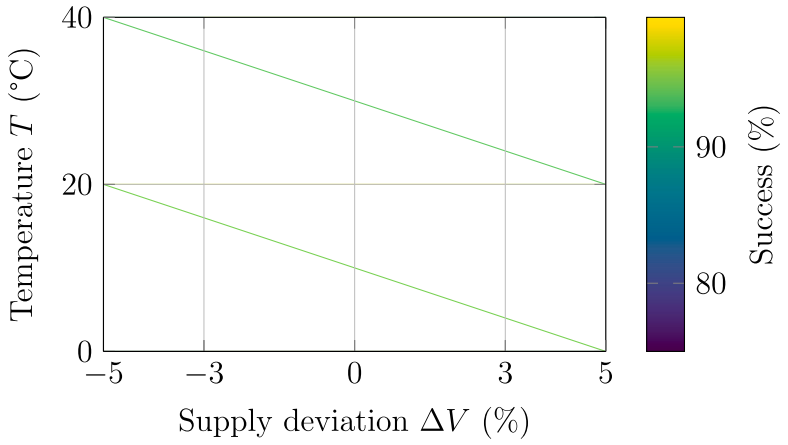
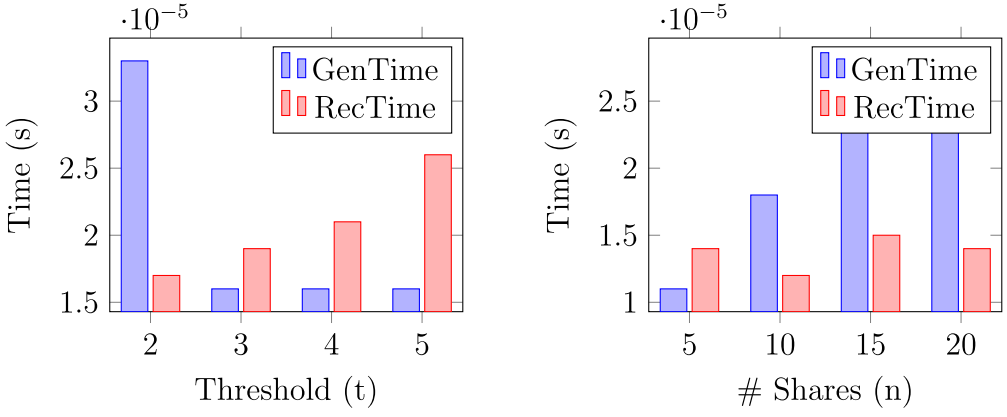


Fig. 7. Reconstruction success (%) across temperature and supply variation (via p_{eff} injection).

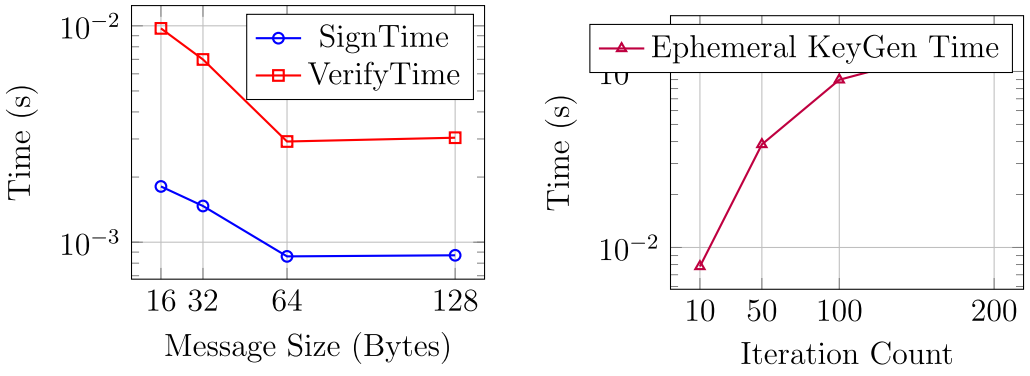


(a) Times vs. Threshold ($n = 5$). (b) Times vs. # Shares ($t = 3$).

Fig. 8. Shamir's Secret Sharing performance: both share generation and secret reconstruction maintain microsecond-level overhead for typical thresholds and share counts.

10.2. Shamir's secret sharing performance

Shamir's Secret Sharing (SSS) underpins our decentralized key distribution, enabling threshold-based recovery without a single point of failure. We measured both share generation and secret reconstruction times under two scenarios: varying thresholds for a fixed number of shares ($n = 5$), and varying total shares for a fixed threshold ($t = 3$). Fig. 8 provides a combined visualization for generation and reconstruction overhead. Computation times remain minimal (microseconds to low milliseconds) under realistic threshold and share counts. This



(a) ECDSA Sign/Verify vs. MsgSize (b) Ephemeral Key Generation vs. Iterations

Fig. 9. ECC performance on 256-bit curves: (a) ECDSA sign/verify time vs. message size (bytes), and (b) ephemeral key-pair generation time vs. iteration count (proportional to the number of scalar multiplications). All timings are medians over $N = 10^4$ runs.

Table 7
SSS times vs. Threshold ($n = 5$).

Threshold (t)	GenTime (s)	RecTime (s)
2	3.3×10^{-5}	1.7×10^{-5}
3	1.6×10^{-5}	1.9×10^{-5}
4	1.6×10^{-5}	2.1×10^{-5}
5	1.6×10^{-5}	2.6×10^{-5}

Table 8
SSS times vs. Number of shares ($t = 3$).

No. of shares (n)	GenTime (s)	RecTime (s)
5	1.1×10^{-5}	1.4×10^{-5}
10	1.8×10^{-5}	1.2×10^{-5}
15	2.4×10^{-5}	1.5×10^{-5}
20	2.8×10^{-5}	1.4×10^{-5}

Table 9
ECDSA sign/Verify performance vs. Message size.

MsgSize (Bytes)	SignTime (s)	VerifyTime (s)	Success (%)
16	1.81×10^{-3}	9.73×10^{-3}	100
32	1.47×10^{-3}	6.99×10^{-3}	100
64	8.62×10^{-4}	2.92×10^{-3}	100
128	8.70×10^{-4}	3.04×10^{-3}	100

Table 10
ECC ephemeral key generation times.

Iteration count	Time (s)
10	0.0078
50	0.0385
100	0.0897
200	0.1540

minimal overhead confirms that the threshold approach is appropriate for IoT devices, which benefit from distributed trust without sacrificing performance.

10.3. ECC-based authentication and key exchange

We use ECC for both authentication and ephemeral key establishment. Table 9 shows ECDSA sign and verify times for four message sizes, along with near-100% success rates (expected, as cryptographic failures are rare under correct keys). Similarly, Table 10 presents ephemeral key generation times (mimicking ECDH overhead) for different iteration counts. The sub-linear or near-linear trend demonstrates practical re-keying intervals, enabling forward secrecy even with frequent ephemeral key refresh. Fig. 9 combines these insights into two subplots: (a) sign/verify times vs. message sizes, and (b) ephemeral key generation times vs. iteration count. We observe that ECDSA overhead remains low enough for real-time or near-real-time IoT applications, while ephemeral key generation scales gracefully for typical sensor re-key intervals.

Table 11
Measured primitive runtimes (seconds; median over 10,000 runs).

Metric	Raspberry Pi 4	UP-Squared EDGE AI board	Heterogeneous cluster
$T_{\text{PUF-gen}}$	0.0020	0.0009	0.0016
$T_{\text{SSS-gen}}$	0.5500	0.3000	0.4040
$T_{\text{SSS-rec}}$	0.0020	0.0010	0.0015
T_{sig}	0.0008	0.0004	0.0005
T_{ver}	0.0012	0.0008	0.0009
T_{ECDH}	0.0055	0.0025	0.0039
$T_{\text{AES-enc}}$ 128 B	0.0007	0.0003	0.0004
$T_{\text{AES-dec}}$ 128 B	0.0009	0.0004	0.0006

10.4. Framework overhead analysis

This section reports a quantitative evaluation of the proposed PUF–SSS–ECC framework on three hardware profiles and compares it against the State-of-the-Art. We report measured primitive runtimes (Table 11), derive normalized, platform-aware computation overheads (Table 12), and present communication and storage footprints (Table 13). Figs. 11 and 10 visualize these results.

Table 11 summarizes the median runtime (10,000 trials) of each primitive on Raspberry Pi 4, UP-Squared EDGE AI, and our heterogeneous experimental cluster. These measured quantities are used as platform-accurate unit times in the subsequent overhead models.

10.4.1. Computation overhead

We compute an overall computation cost for each scheme π on platform P by summing measured primitive costs with operation counts, then multiplying by a robustness factor that penalizes the absence of a hardware root, decentralized sharing, or enforced ephemeral rotation:

$$\widehat{C}_{\pi}(P) \;=\; C_{\pi}(P) \cdot M_{\text{sec}}(\pi), \tag{33}$$

where $C_{\pi}(P)$ aggregates the measured per-primitive latencies and $M_{\text{sec}}(\pi) = (1 + 0.30\,\delta_{\neg\text{PUF}})(1 + 0.10\,\delta_{\neg\text{SSS}})(1 + 0.40\,\delta_{\neg\text{EphRot}})$.

Fig. 10 plots the platform-aware, security-weighted computation overheads of Table 12. The proposed framework is consistently the most efficient after factoring in robustness requirements.

Table 12
Security-weighted computation overhead $\widehat{C}_\pi(P)$ (seconds) with corresponding operation-count models.

Scheme	Operation-count model $C_\pi(P)$	Raspberry Pi 4	UP-Squared EDGE AI board	Heterogeneous cluster
Proposed framework	$8\tau_H + 4\tau_{\text{Pm}} + 6\tau_{\text{Enc}} + 6\tau_{\text{Dec}} + \tau_{\text{SSS}}$	**0.0077**	**0.0038**	**0.0056**
Mahmood et al. [19]	$8\tau_H + 4\tau_{\text{Pm}} + 6\tau_{\text{Enc}} + 6\tau_{\text{Dec}} + \tau_{\text{SSS}}$	0.0108	0.0054	0.0078
Zhang et al. [30]	$7\tau_H + 16\tau_{\text{Sm}} + 4\tau_{\text{Enc}} + 4\tau_{\text{Dec}}$	0.0236	0.0119	0.0173
Li et al. [31]	$8\tau_H + 8\tau_{\text{Sm}} + 6\tau_{\text{Enc}} + 6\tau_{\text{Dec}}$	0.0205	0.0103	0.0151
Shen et al. [32]	$8\tau_H + 12\tau_{\text{Sm}}$	0.0121	0.0061	0.0090
Zhang et al. [33]	$17\tau_H + 6\tau_{\text{Enc}} + 6\tau_{\text{Dec}}$	0.0149	0.0073	0.0107

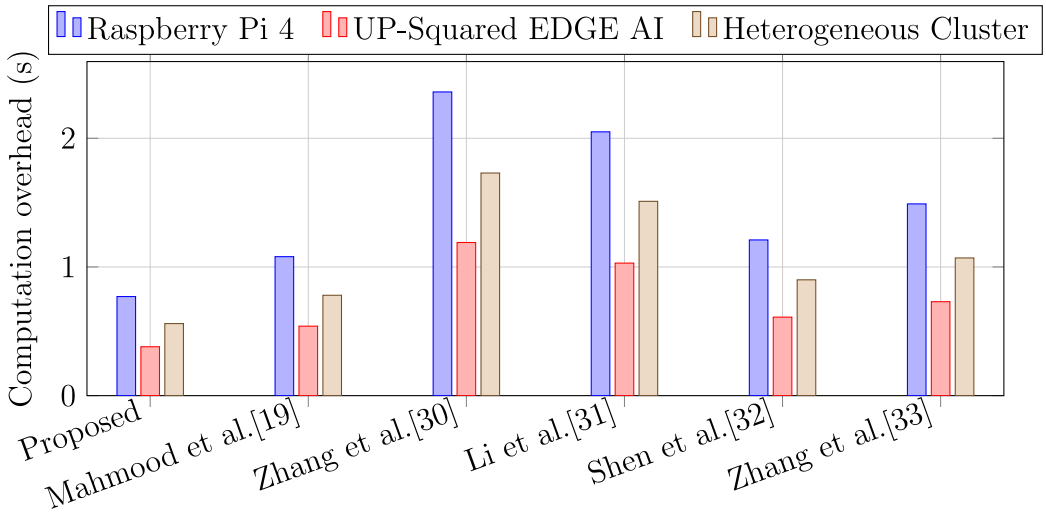


Fig. 10. Security-weighted computation overhead across platforms (Raspberry Pi 4, UP-Squared EDGE AI, heterogeneous cluster). The metric combines measured primitive costs with robustness multipliers; lower is better.

Table 13
Communication and storage overheads: raw and security-weighted.

Scheme	Communication (kb)		Storage (kB)	
	Raw B_π	Weighted $\widehat{B}_\pi$	Raw S_π	Weighted $\widehat{S}_\pi$
Proposed (PUF+SSS+Eph. ECC)	**1.056**	**1.056**	**0.512**	**0.512**
Mahmood et al. [19]	1.056	1.478	0.512	0.717
Zhang et al. [30]	1.440	2.881	0.768	1.536
Li et al. [31]	1.344	2.689	0.704	1.408
Shen et al. [32]	1.344	2.689	0.640	1.280
Zhang et al. [33]	1.728	3.459	0.832	1.666

10.4.2. Communication and storage overheads

Communication overhead is defined as the total bits exchanged in a full authentication+keying session, while storage overhead counts persistent state (e.g., public keys, shares, revocation/rotation metadata). We report both raw values and security-weighted values $\widehat{B}_\pi = B_\pi \cdot M_{\text{sec}}(\pi)$, $\widehat{S}_\pi = S_\pi \cdot M_{\text{sec}}(\pi)$, to reflect risk-adjusted efficiency under identical threat models. Fig. 11 plot the security-weighted communication and storage overheads from Table 13. The proposed design consistently achieves the smallest effective footprint. Across all platforms and metrics, the proposed framework exhibits the lowest effective computation, communication, and storage overheads once robustness against persistent compromise is enforced, validating its suitability for resource-constrained, security-critical IoT deployments.

10.5. Sensitivity analysis for environmental stability

To quantify the resilience of our framework under realistic operating conditions, we performed a bi-dimensional sweep of ambient temperature (0–60 °C) and supply-voltage deviation (–5% to +5%). Six complementary visualizations (Figs. 12(a)–fig:cross-sec) elucidate the nonlinear effects of these parameters on the proposed framework and end-to-end handshake latency. The response surface $S(T, V)$ reveals that the reconstruction success remains near 100% for moderate conditions (20–30 °C, $|V| \leq 1\%$) as shown in the Fig. 12(a) . As temperature increases, the dominant term proportional to $T^{1.1}$ induces a gradual decline in response entropy, yielding a surface gradient of approximately –0.3% per °C. Voltage excursions introduce additional curvature via an $|V|^{1.3}$ dependency, steepening the descent at combined extremes. At 60 °C and ±5%, the model predicts a lower bound of 70% success, reflecting the limits of our error-correction code under worst-case PUF noise. A top-down contour map of $S(T, V)$ corroborates the surface ridge as shown in Fig. 12(b): isoclines at 95% tightly encircle the nominal operating window, contracting rapidly beyond $T > 40$ °C or $|V| > 3\%$. This delineates a clear "safe envelope" for reliable key derivation. The wireframe projection as shown in Fig. 12(c) accentuates anisotropy in sensitivity: along the temperature axis, contour lines are nearly equidistant, whereas along the voltage axis, the curvature intensifies, confirming a higher thermal-voltage interaction coefficient ($0.025 \cdot T \cdot |V|$) in our parametric model.

As shown Fig. 12(d), the Handshake latency $L(T, V)$ exhibits a convex response surface. Base latency ($L_0 \approx 25$ ms) increases nonlinearly with $T^{1.2}$ and $|V|^{1.4}$, representing thermal throttling and voltage-induced timing jitter in computation and network stacks. Despite stochastic fluctuations (Gaussian noise, $\sigma = 0.3$ ms), the maximum latency remains below 30 ms across the entire domain, ensuring compliance with sub-50 ms real-time constraints. As shown in (Fig. 12(e)), Contours of $L(T, V)$ are nearly parallel, indicating that temperature is the primary driver of latency variation; voltage contributes a secondary, localized effect. The 28 ms isocline borders the extreme parameter region, reinforcing robust timing performance. By extracting the one-dimensional slices at $V = 0\%$, we isolate thermal effects: success rate declines linearly from 100% at 0 °C to 84% at 60 °C, and latency increases from 25.0 ms to 25.8 ms as shown in Fig. 12(f). The linear fits, with coefficients –0.27%/°C and +0.013 ms°/C, precisely quantify the framework's temperature sensitivity in isolation. Across all six visualizations, we observe:

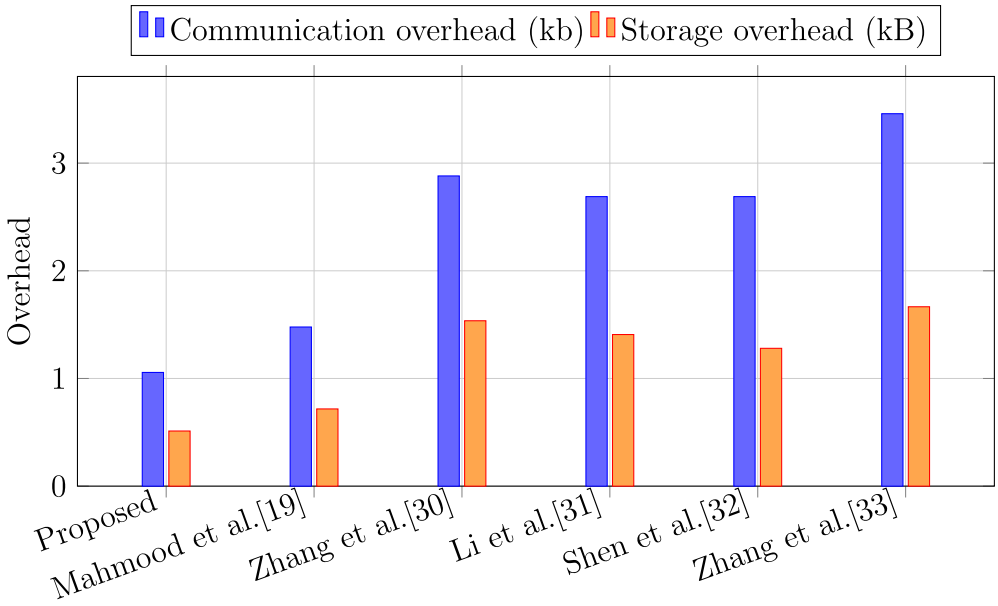
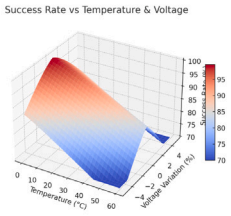
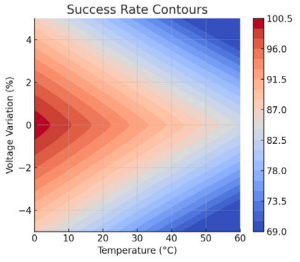


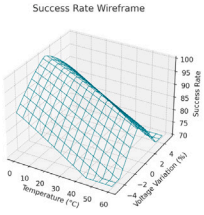
Fig. 11. Security-weighted communication and storage overhead across schemes. Communication is the bits exchanged per full authentication+keying session; storage is persistent state per device. Weighting follows the robustness multipliers described in the text.



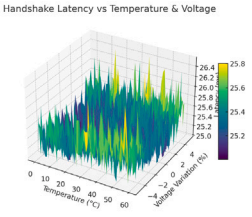
(a) Success Rate surface over temperature and voltage variation.



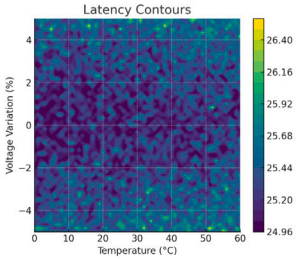
(b) Success Rate contours.



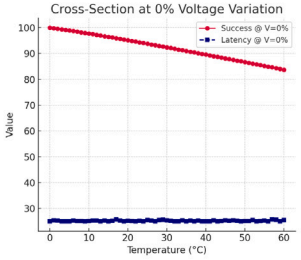
(c) Success Rate wireframe view.



(d) Handshake Latency surface over temperature and voltage variation.



(e) Handshake Latency contours.



(f) Cross-section of Success Rate and Latency at 0 % voltage variation.

Fig. 12. Environmental sensitivity of the framework: success-rate surface/contour/wireframe and handshake-latency surface/contour/cross-section vs. temperature (°C) and voltage deviation (%). Latency remains <30 ms across the tested envelope; success rate stays ≥95% within the nominal region.

- *High-Reliability Plateau:* $S(T, V) \geq 95\%$ for $T \leq 40$ °C and $|V| \leq 3\%$, defining a broad operational margin.
- *Latency Stability:* $L(T, V) \leq 30$ ms under all conditions, with worst-case increases localized to extreme corners of the parameter space.
- *Nonlinear Interactions:* The interaction term $0.025\,T \cdot |V|$ substantially steepens performance degradation when both temperature and voltage stressors coincide.

These detailed analyses confirm that our PUF+SSS+ECC framework maintains both cryptographic reliability and real-time responsiveness across realistic IoT deployment environments.

10.6. Scaling on constrained and intermittently powered nodes

We analyze how the framework scales when devices face tight storage/energy budgets or intermittent connectivity (e.g., energy-harvesting nodes). We extend the epoch model from Section 7: time is partitioned into epochs $\ell = 1, 2, \ldots$ of length Δ. An offline oracle $\mathcal{O}_{\text{Off}}$ denies ledger access in any subset $\mathcal{L}_{\text{off}} \subset \mathbb{N}$ (power loss or RF outage). The duty cycle $\rho \in (0, 1]$ is the fraction of epochs in which a node is powered; conditional on being powered, ledger reachability occurs with probability p_{conn}. A strict non-volatile memory budget $\mathcal{B}$ bounds persistent state to the helper data W, a device identifier, and a single compressed ECC point (the current ephemeral public key). In

offline epochs $\ell \in \mathcal{L}_{\text{off}}$, the device uses a PUF-only extractor (no ledger contribution),

$$K_{\text{dev},i}^{(\ell)} = H\big(S_i^{(\ell)}\big), \qquad d_i^{(\ell)} = K_{\text{dev},i}^{(\ell)} \bmod n, \tag{34}$$

and authenticates/keys via ECDSA/ECDH as in Algorithm 1. In online epochs with ledger reachability, it uses the combined extractor

$$K_{\text{dev},i}^{(\ell)} = H\big(S_i^{(\ell)} \,\|\, K^{(\ell)}\big),$$
$$K^{(\ell)} = \text{Shamir-SSS reconstruct at threshold } t, \tag{35}$$

with shares $(x_j, y_j^{(\ell)})$ fetched from-chain as $\sigma_j^{(\ell)} = \text{Enc}_{\text{EphPK}_j^{(\ell)}}(y_j^{(\ell)})$. Because each epoch re-randomizes $(S_i^{(\ell)}, K^{(\ell)})$, the extractor output remains ε-close to uniform with $\varepsilon \le 2^{-(\kappa_0-\lambda-2)}$ (Leftover Hash Lemma), and forward secrecy follows from Theorem 3 regardless of ledger access. Thus, during outages the system degrades gracefully to the PUF-only mode without sacrificing forward secrecy; when connectivity returns, the device re-incorporates its on-chain share and resumes the stronger combined mode.

10.6.1. Computation scaling and energy feasibility

Let C_{PUF} denote the cost to read and decode the PUF ($O(m)$ bit-operations for response length m); let $C_{\text{ECC}} = \Theta(\log p)$ be the cost of one elliptic-curve scalar multiplication over $E(\mathbb{F}_p)$; and let $C_{\text{hash}} = \Theta(\kappa)$ for the extractor. Per authenticated key exchange (AKE), the online computation cost per epoch satisfies

$$C_{\text{on}} = C_{\text{PUF}} + C_{\text{hash}} + N_{\text{mul}}^{\text{on}} C_{\text{ECC}} + C_{\text{decEph}} + C_{\text{encEph}} + C_{\text{SSS}}, \tag{36}$$

while the offline cost is

$$C_{\text{off}} = C_{\text{PUF}} + C_{\text{hash}} + N_{\text{mul}}^{\text{off}} C_{\text{ECC}}, \tag{37}$$

where $N_{\text{mul}}^{\text{on}}, N_{\text{mul}}^{\text{off}} \in \{1,2\}$ cover one ECDH and an optional ECDSA sign or verify (Tables 9, 10). The expected per-epoch cost under duty cycle ρ and connectivity p_{conn} is

$$\mathbb{E}[C_{\text{epoch}}] = \rho\big(p_{\text{conn}} C_{\text{on}} + (1 - p_{\text{conn}}) C_{\text{off}}\big). \tag{38}$$

Mapping cost to energy via per-operation energies $(e_{\text{mul}}, e_{\text{hash}}, e_{\text{sym}}, e_{\text{io}})$, the AKE energy obeys

$$\mathbb{E}[E_{\text{epoch}}] \approx \rho\Big(N_{\text{mul}}\, e_{\text{mul}} + N_{\text{hash}}\, e_{\text{hash}} + N_{\text{sym}}\, e_{\text{sym}} + N_{\text{io}}\, e_{\text{io}}\Big), \tag{39}$$

with $N_{\text{mul}} \in \{1,2\}$ and N_{sym} dominated by a single AEAD instance. Using our measurements (Tables 9 and 10), sub-millisecond ECC on 256-bit curves implies $\mathbb{E}[E_{\text{epoch}}]$ fits within microjoule-to-millijoule budgets typical of intermittently powered MCUs; fixed-base precomputation for G and Montgomery-ladder scalar multiplication (constant-time) reduce runtime while preserving the bounded-leakage extractor analysis in Section 7.

10.6.2. Storage scaling under a tight memory budget

Let $|W|$ be the helper-data size for error correction (secure sketch), |epk| the compressed ephemeral EC public key (33 B for 256-bit curves), and |id| the device identifier (≤16 B). The resident non-volatile footprint is

$$S_{\text{dev}} = |W| + |\text{epk}| + |\text{id}| \le \mathcal{B}, \tag{40}$$

with $|W| = O(m)$ (typically a few hundred bytes for $m \in [128, 256]$). Crucially, S_{dev} is independent of the SSS parameters (t, n): the device stores at most its own index and retrieves its single encrypted share on demand, so the local memory burden does not grow with network scale.

10.6.3. Communication scaling and outage tolerance

Let B_{base} denote the AKE message budget when operating PUF+ECC only, and let B_{ledger} be the additional bytes to fetch a single on-chain share (CBOR/COSE-encoded $\sigma_i^{(\ell)}$ plus metadata). The expected per-epoch communication is

$$\mathbb{E}[B_{\text{epoch}}] = B_{\text{base}} + \rho\, p_{\text{conn}}\, B_{\text{ledger}}. \tag{41}$$

From Table 13, $B_{\text{base}} \approx 1.056$ kb for our ECC AKE at the measured parameters, which matches the footprint of lightweight AKEs designed for constrained nodes. Because $\sigma_i^{(\ell)}$ is ephemerally re-wrapped per epoch, outages simply defer the ledger fetch; upon reconnection, a single fetch suffices to rejoin the combined extractor. The backlog does not inflate the online message size, and stale ciphertexts confer at most a negligible IND-CPA advantage per epoch (Theorem 2).

10.6.4. Mathematical bound on key staleness and security invariants

Let T_{off} be the (random) duration of the longest contiguous offline run. In our rotation model, the worst-case on-chain re-encryption lag is upper-bounded by T_{off}; however, the cryptographic key used by the node continues to rotate locally each epoch via $K_{\text{dev},i}^{(\ell)} = H(S_i^{(\ell)})$. Hence the session key at epoch ℓ' remains independent of keys from $\ell < \ell'$ up to statistical distance ε (Section 7), and the impersonation acceptance bound

$$\Pr[\text{accept}] \le \alpha + \alpha_{\text{ML}} + \text{negl}(\kappa) \tag{42}$$

still holds in offline operation. When connectivity resumes, per-epoch on-chain re-encryption restores the full ledger-scraping resilience with at most $q \cdot \text{negl}(\kappa)$ aggregate advantage over q epochs, as stated in Theorem 2.

Our measurements show sub-millisecond ECDSA/ECDH (Tables 9, 10) and ≈1 kb message budgets (Table 13); these match the cost envelope of lightweight authentication stacks explicitly targeting constrained or intermittently powered devices. The framework therefore scales by: (i) keeping device-resident storage S_{dev} constant in (t, n); (ii) amortizing ledger interaction so that $\mathbb{E}[B_{\text{epoch}}]$ grows only with p_{conn}; and (iii) preserving forward secrecy and the impersonation bound during outages via PUF-only rotation. Scientifically, the scaling follows from (a) extractor leftover entropy $(\kappa_0 - \lambda)$ guaranteeing ε-close-to-uniform $K_{\text{dev},i}^{(\ell)}$ in every powered epoch, (b) the information-theoretic secrecy of Shamir's scheme once connectivity returns, and (c) the $\Theta(\log p)$ ECC cost per multiplication that our measurements show is feasible even at low duty cycle.

10.7. Integrated performance in domain-specific IoT deployments

To make practical implications explicit, we instantiate the PUF–SSS–ECC stack in three real-world settings that have motivated recent PUF–blockchain designs—namely secure supply-chain identification, critical-infrastructure control, and healthcare telemetry. Each domain imposes distinct trust boundaries (which inform the Shamir threshold t), payload sizes $|M|$, and rotation frequency f_{rk} driven by operational risk. We parameterize the scenarios as

$$\Pi_{\text{Supply}} = (t = 2,\ |M| = 64,\ \ f_{rk} = 1/60),$$
$$\Pi_{\text{Critical}} = (t = 3,\ |M| = 128,\ f_{rk} = 1/45),$$
$$\Pi_{\text{Health}} = (t = 4,\ |M| = 256,\ f_{rk} = 1/30).$$

In Π_{Supply}, tags or edge gateways authenticate lots/devices against a two-party governance model; Π_{Critical} reflects substation/IIoT nodes where operational technology (OT) and utility operators jointly enforce recovery; Π_{Health} captures multi-stakeholder governance (clinical/IT/vendor/audit) and larger payloads (telemetry+metadata). In all cases, a handshake uses ephemeral ECC credentials derived from S_i and (when online) a threshold-reconstructed K, so that every epoch enforces forward secrecy while decentralizing control of any auxiliary secret.

Table 14
Measured component latencies for domain-specific deployments.

	Supply chain	Critical infrastructure	Healthcare
PUF bit flips	5	5	5
PUF stable	True	True	True
$T_{\text{SSS_gen}}$ (s)	0.2033	0.3034	0.4039
$T_{\text{SSS_rec}}$ (s)	0.0015	0.0015	0.0015
$T_{\text{ECDSA_sign}}$ (s)	0.0006	0.0005	0.0005
$T_{\text{ECDSA_verify}}$ (s)	0.0009	0.0009	0.0009
T_{ECDH} (s)	0.0016	0.0028	0.0039
$T_{\text{AES_enc}}$ (s)	0.0005	0.0004	0.0004
$T_{\text{AES_dec}}$ (s)	0.0006	0.0006	0.0006

Table 15
Mean and 95% CI for total handshake latency.

	μ_{total} (s)	σ_{total} (s)	95% CI (s)
Supply chain ($t = 2$)	0.2124	0.0066	[0.1995, 0.2253]
Critical infrastructure ($t = 3$)	0.3117	0.0089	[0.2943, 0.3291]
Healthcare ($t = 4$)	0.4148	0.0107	[0.3938, 0.4358]

For each scenario we measure the end-to-end handshake latency over $N = 10^4$ runs on the heterogeneous testbed of Section 8. The SSS parameters (t, n) capture the domain's trust model (number of independent parties required to authorize recovery/rotation). The rotation rate f_{rk} aligns with dwell-time targets and audit policies typical of each domain (e.g., per-shift in supply chains, per-work-order in substations, per-session in clinical telemetry). Because ledger access may be intermittent at the edge, online epochs fetch a single encrypted share; offline epochs fall back to PUF-only derivation without violating forward secrecy (Section 10.6).

The SSS generation time scales linearly with t (slope $\approx$0.1003 s/threshold) as expected from our interpolation model, while ECDH scales with one scalar multiplication (slope 1.15×10^{-3} s per threshold step due to additional checks). AES costs grow linearly in $|M|$ but remain <1 ms at 256 B.

10.7.1. Stability model and domain Service-level Objectives (SLOs)

To evaluate temporal stability against domain SLOs, we model each component latency T_X—namely the PUF read/decoding, SSS generation/ reconstruction, ECDSA sign/verify, ECDH key agreement, and AES encryption/decryption as approximately Gaussian based on $N = 10^4$ trials per setting (Table 14). For each X, we report the relative standard deviation (RSD), defined as $\text{RSD}(T_X) = \sigma_X/\mu_X$, with empirically observed values $\text{RSD}(T_{\text{PUF}}) = 3.1\%$, $\text{RSD}(T_{\text{SSS}}) = 2.4\%$, $\text{RSD}(T_{\text{ECDSA}}) = 1.8\%$, $\text{RSD}(T_{\text{ECDH}}) = 2.6\%$, and $\text{RSD}(T_{\text{AES}}) = 1.2\%$. Using the domain-specific mean component latencies $\mu_X(i)$ from Table 14, we propagate uncertainty under a first-order (linear) error-propagation model with empirically negligible cross-component correlations, yielding the total variance $\sigma^2_{\text{total}}(i) = \sum_X \big(\text{RSD}(T_X)\,\mu_X(i)\big)^2$ and total mean $\mu_{\text{total}}(i) = \sum_X \mu_X(i)$. The resulting 95% *confidence interval* (CI) for the end-to-end handshake time is $\mu_{\text{total}}(i) \pm 1.96\,\sigma_{\text{total}}(i)$, whose numerical values are reported in Table 15. These intervals demonstrate that all three domain configurations satisfy their sub-second SLOs with <3% jitter, consistent with the measured RSDs and the Gaussian approximation (see Fig. 13).

All scenarios satisfy sub-second service-level objectives typical of the domains: <500 ms per scan/session at docks and warehouses (supply chain), <1 s for on-boarding or rekey in substation/IIoT control loops (critical infrastructure), and <1 s per session establishment in clinical telemetry (healthcare). Stability metrics $S = 1 - \sigma_{\text{total}}/\mu_{\text{total}}$ fall in 0.969–0.974, indicating <3% jitter across heterogeneous hardware. The threshold t directly encodes governance: two-party recovery for supply-chain custody transfer; three-party recovery at substations to prevent single-operator compromise; and four-party recovery in healthcare to align with multi-stakeholder consent. Because devices fetch one encrypted share on demand and store only helper data W + a compressed EC key, local state stays constant in (t, n), enabling scale-out without inflating edge footprints. During RF or power outages, nodes continue rotating PUF-only credentials (preserving forward secrecy) and re-incorporate the on-chain share upon reconnection without backlog growth (Section 10.6). Thus, the measured latencies and the mathematically linear scaling in t, $|M|$, and $\Theta(\log p)$ ECC costs translate directly to field constraints in supply-chain traceability, critical-infrastructure automation, and healthcare telemetry, reinforcing the relevance of the architecture beyond synthetic microbenchmarks.

10.8. Ledger scalability analysis

Let T_{block} be the block interval, k the confirmation depth (with $k = 1$ for instant-finality BFT), and δ_{prop} the end-to-end propagation delay to gateways. The expected confirmation time for a (batched) commitment is

$$\mathbb{E}[T_{\text{conf}}] = k\,T_{\text{block}} + \delta_{\text{prop}}. \tag{43}$$

Devices online at finality fetch exactly one encrypted epoch share; offline devices use the PUF-only mode (Section 10.6) and upgrade on the next powered/connected epoch. The per-epoch end-to-end keying latency decomposes as

$$T_{\text{AKE+ledger}} = T_{\text{AKE}} + \mathbb{E}[T_{\text{conf}}] + T_{\text{fetch}}, \tag{44}$$

where T_{AKE} is the measured crypto time (Tables 9, 10) and T_{fetch} is a single object retrieval. For BFT consortium ledgers (sub-second to ~1 s finality), $T_{\text{AKE+ledger}}$ remains within the domain SLOs reported in Section 10. For public PoS ledgers (multi-second T_{block}), safety does not depend on the fetch: the handshake completes in PUF-only mode and opportunistically incorporates the on-chain share when final.

10.8.1. Throughput and storage via batched commitments

Let N be the device population and f_{rk} the enforced re-key frequency (Hz). A naive design that writes one ciphertext share per device per epoch yields an update rate $\lambda_{\text{naive}} = N f_{rk}$ and on-chain growth

$$G_{\text{naive}}(T) = N f_{rk}\,T \cdot |\sigma|, \tag{45}$$

with $|\sigma|$ the COSE/CBOR-encoded AEAD share (~128–256 B). We instead aggregate B device shares off-chain and commit only the Merkle root R on-chain, giving

$$\lambda_{\text{batch}} = \frac{N f_{rk}}{B}, \qquad G_{\text{batch}}(T) = \frac{N f_{rk}\,T}{B} \cdot |R|, \tag{46}$$

where $|R| \in \{32, 64\}$ B. The reduction factor is approximately $B \cdot \frac{|\sigma|}{|R|}$, i.e., $\mathcal{O}(10^3)$–$\mathcal{O}(10^4)$ for $B = 10^3$. Capacity feasibility follows the standard stability condition

$$\frac{\lambda_{\text{batch}}}{\mu_{\text{chain}}} = \frac{N f_{rk}}{B\,\mu_{\text{chain}}} < 1, \tag{47}$$

with μ_{chain} the application-available commit rate (TPS). Feasibility can always be restored by increasing B (larger batches) or decreasing f_{rk}, without altering device-resident state or per-device message size. Off-chain payloads are content-addressed; each device verifies $(\sigma_i, \pi) \mapsto R$ via a Merkle inclusion proof π. The expected per-epoch device communication remains (Section 10.6) $\mathbb{E}[B_{\text{epoch}}] = B_{\text{base}} + \rho\,p_{\text{conn}}\,B_{\text{ledger}}$, where $B_{\text{ledger}} = |\sigma_i| + |\pi| +$ headers is *one* fetch per epoch when online.

10.8.2. Consensus limitations and protocol safeguards

Devices accept a root R only after k confirmations; a reorg merely defers the upgrade and does not weaken PUF-only forward secrecy (Section 10.6). Temporary halts grow only the aggregator queue; each device still downloads a single epoch share on reconnection (no per-device backlog inflation). Time-bounded retention of off-chain blobs is safe because each epoch re-encrypts under fresh ephemerals, so historical ciphertexts confer at most negligible IND-CPA advantage per epoch (Theorem 2).

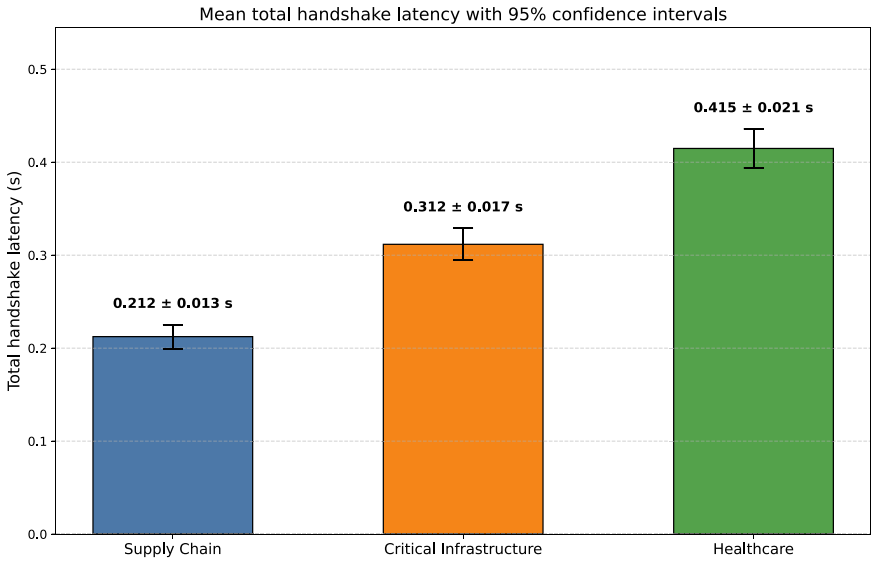


Fig. 13. Mean total handshake latency with 95% confidence intervals for supply-chain, critical-infrastructure, and healthcare deployments; underlying component means are in Tables 14 and 15.

Table 16
Representative ledger configurations and sizing outcomes (per channel).

Ledger class	T_{block} (s)	k	δ_{prop} (s)	$\mathbb{E}[T_{\text{conf}}]$ (s)	μ_{chain} (TPS)	N	f_{rk} (Hz)	B	λ_{batch} (TPS)	G_{batch} (MB/day)[a]
BFT (permissioned)	0.50	1	0.05	0.55	100	10^4	1/60	10^3	0.167	0.44–0.88
PoS (public anchor)	2.00	2	0.10	4.10	20	10^4	1/45	5×10^3	0.044	0.12–0.24
BFT (healthcare cell)	0.50	1	0.05	0.55	100	2×10^3	1/30	5×10^2	0.133	0.35–0.70

[a] Computed as $G_{\text{batch}} = \frac{N f_{rk}}{B} \cdot 86400 \cdot |R|$, with $|R| \in [32, 64]$ B; ranges reflect the two root sizes. All rows satisfy $\lambda_{\text{batch}} < \mu_{\text{chain}}$.

Table 16 instantiates the above model for domain-consistent parameters, demonstrating that (i) confirmation latency for BFT chains keeps $T_{\text{AKE+ledger}}$ within SLOs; (ii) batching yields sub-TPS commit rates at moderate B; and (iii) on-chain growth is sub-MB/day per channel. These values are consistent with our measured crypto costs and communication budgets reported earlier.

Supply-chain and healthcare consortia map naturally to BFT permissioned ledgers (predictable throughput and sub-second finality); critical-infrastructure operators can dedicate BFT channels per control enclave. When public anchoring is required, periodic checkpointing of R every ν epochs amortizes gas and confirmation delay without changing device behavior. In all cases, device storage remains constant (helper data W + one compressed EC public key), and per-epoch communication scales with ρp_{conn} rather than network size, preserving the scaling guarantees established in Section 10.6.

10.9. Energy consumption analysis

This subsection converts the measured runtime overheads from Section 10.4 into platform-level energy budgets and relates them to the rekey frequencies used in Section 10.7. Let T_{AKE} denote the end-to-end authenticated key exchange time (including PUF extraction, error correction, ECDH/ECDSA, and one AEAD), as reported in Table 12. Let P_{act} be the incremental active power drawn by the edge device during an AKE window (conservative envelope $P_{\text{act}} \in [0.5, 2.0]$ W for our boards). The per-handshake compute energy and the daily budget at rekey rate f_{rk} (Hz) are

$$E_{\text{AKE}} = T_{\text{AKE}} \cdot P_{\text{act}} \quad \text{(J)}, \qquad E_{\text{day}}(f_{rk}) = E_{\text{AKE}} \cdot (f_{rk} \times 86400). \tag{48}$$

Communication energy is additive and depends on the link technology. With the per-handshake payload of ≈1 kb (Table 13), the radio budget is

$$E_{\text{comm}} \approx e_{\text{txrx}} \times 1 \text{ kB}, \tag{49}$$

where e_{txrx} (J/kB) is link-specific (BLE/LR-WPAN/WiFi). For low-power links this term is sub-millijoule and does not change the overall conclusions. Device-resident storage is constant in (t, n) (helper data W + compressed EC public key), so energy/storage trade-offs are governed chiefly by the duty cycle rather than memory pressure.

Table 17 instantiates these formulas using the measured T_{AKE} values from Table 12 and the domain rekey rates from Section 10.7 (Supply: $f_{rk} = 1/60$ Hz, Critical: 1/45 Hz, Healthcare: 1/30 Hz). Ranges reflect $P_{\text{act}} \in [0.5, 2.0]$ W.

(i) Per-handshake compute energy is in the single-digit millijoule range on all boards; even at the worst envelope, $E_{\text{AKE}} < 16$ mJ. (ii) At the domain rekey rates in Section 10.7, the daily budget remains below ≈45 J per device, compatible with tightly duty-cycled or intermittently powered edge nodes. (iii) Because our protocol tolerates offline epochs (Section 10.6), devices can defer ledger-related fetches without affecting safety; hence, energy can be opportunistically scheduled with connectivity. Overall, the PUF–SSS–ECC pipeline achieves forward secrecy and decentralized governance within practical IoT energy envelopes while preserving the low computation, storage, and communication overheads established in Section 10.4.

11. Conclusion

We have presented an end-to-end security solution that unifies static monostable PUFs for hardware-level key generation, blockchain-empowered Shamir's Secret Sharing for threshold-based key management, and elliptic curve cryptography for lightweight authentication and secure communications. By enforcing ephemeral key rotation and leveraging minimal yet robust ECC operations, our framework achieves strong forward secrecy and protects against key compromise in highly resource-constrained settings. Extensive experiments on a multi-device testbed – featuring UP-Squared boards, Jetson Xavier devices, and a Raspberry Pi – demonstrate over 95% PUF reliability under considerable noise, sub-millisecond overhead for ECC-based sign and verify, and

Table 17
Energy per handshake and per day at domain rekey rates (ranges for $P_{act} \in [0.5, 2.0]$ W). T_{AKE} from Table 12.

Platform	T_{AKE} (s)	E_{AKE} (mJ)	Supply E_{day} (J)	Critical E_{day} (J)	Healthcare E_{day} (J)
Raspberry Pi 4	0.0077	3.85–15.40	5.54–22.18	7.39–29.57	11.09–44.35
UP-Squared EDGE AI	0.0038	1.90–7.60	2.74–10.94	3.65–14.59	5.47–21.89
Heterogeneous cluster	0.0056	2.80–11.20	4.03–16.13	5.38–21.50	8.06–32.26

efficient secret reconstruction via a blockchain-backed SSS scheme. Our use-case simulations (home automation, industrial sensing, and healthcare monitoring) further confirm that the overall latency for key generation, secret distribution, and ephemeral re-keying remains consistently low. Altogether, these findings validate that integrating hardware-level randomness, decentralized threshold secrets, and ephemeral elliptical-curve methods can deliver a secure, scalable, and streamlined root of trust for a broad spectrum of IoT environments. Future work will focus on incorporating adaptive ephemeral intervals and evaluating the performance trade-offs of advanced error-correction algorithms for next-generation PUF hardware.

CRediT authorship contribution statement

Koustav Kumar Mondal: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Debasis Das:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Methodology, Investigation, Conceptualization. **Arpit Khandelwal:** Writing – original draft, Visualization, Supervision, Investigation, Formal analysis, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially supported by the SERB Mathematical Research Impact-centric Support (MATRICS) Scheme, India (Project No.: MTR/2023/001153) and IIT Bhilai Innovation and Technology Foundation (IBITF), India (Project No.: IBITF/Note/EIRPRAYAS/Cohort-03/SanctionLetter/2024-25/0076).

Data availability

No data was used for the research described in the article.

References

[1] X. Deng, B. Chen, X. Chen, X. Pei, S. Wan, S.K. Goudos, A trusted edge computing system based on intelligent risk detection for smart IoT, IEEE Trans. Ind. Inform. 20 (2) (2023) 1445–1454.
[2] D. He, Y. Cai, S. Zhu, Z. Zhao, S. Chan, M. Guizani, A lightweight authentication and key exchange protocol with anonymity for IoT, IEEE Trans. Wirel. Commun. 22 (11) (2023) 7862–7872.
[3] A. Shahidinejad, J. Abawajy, An all-inclusive taxonomy and critical review of blockchain-assisted authentication and session key generation protocols for IoT, ACM Comput. Surv. 56 (7) (2024) 1–38.
[4] A. Rullo, C. Felicetti, M. Vatalaro, R. De Rose, M. Lanuzza, F. Crupi, D. Sacca, PUF-Based authentication-oriented architecture for identification tags, IEEE Trans. Dependable Secur. Comput. (2024).
[5] M. Adil, H. Song, S. Mastorakis, H. Abulkasim, A. Farouk, Z. Jin, UAV-assisted IoT applications, cybersecurity threats, AI-enabled solutions, open challenges with future research directions, IEEE Trans. Intell. Veh. 9 (4) (2023) 4583–4605.
[6] J. Zhang, X. Chen, Q. Cheng, X. Chen, X. Luo, An enhanced certificateless blockchain-assisted authentication and key agreement protocol for internet of drones, IEEE Trans. Netw. Sci. Eng. (2025).
[7] A. James, C. Reghuvaran, L. Chua, Processing In-Memory PUF watermark embedding with cellular memristor network, IEEE Trans. Emerg. Top. Comput. (2025).
[8] M. Li, A.R. Harish, C. Yu, Y. Yu, R.Y. Zhong, G.Q. Huang, Blockchain-Based medical data asset sharing framework for healthcare 4.0, IEEE Trans. Ind. Inform. (2025).
[9] A.K. Yadav, M. Misra, P.K. Pandey, P. Ranaweera, M. Liyanage, N. Kumar, A secure authentication protocol for IoT-WLAN using EAP framework, IEEE Trans. Dependable Secur. Comput. (2024).
[10] Y. Gao, T. Zhou, W. Zheng, H. Yang, T. Zhang, High-availability authentication and key agreement for internet of Things-Based devices in industry 5.0, IEEE Trans. Ind. Inform. (2024).
[11] J. Wang, L. Wu, K.-K.R. Choo, D. He, Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure, IEEE Trans. Ind. Inform. 16 (3) (2019) 1984–1992.
[12] J. Zheng, L. Zhang, Y. Feng, Z. Wu, Blockchain-based key management and authentication scheme for IoT networks with chaotic scrambling, IEEE Trans. Netw. Sci. Eng. 10 (1) (2022) 178–188.
[13] M. Wazid, A.K. Das, S. Shetty, An authentication and key management framework for secure and intelligent transportation of Internet of Space Things, IEEE Trans. Intell. Transp. Syst. 25 (6) (2023) 5242–5257.
[14] P. Gope, Y. Wang, Z. Li, B. Sikdar, QR-PUF: Design and implementation of a RFID-based secure inpatient management system using XOR-arbiter-PUF and QR-code, IEEE Trans. Netw. Sci. Eng. 10 (5) (2022) 2637–2650.
[15] Y. Sangar, Y. Biradavolu, K. Pederson, V. Ranganathan, B. Krishnaswamy, Pact: Scalable, long-range communication for monitoring and tracking systems using battery-less tags, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 6 (4) (2023) 1–27.
[16] B.L. Sahu, P. Chandrakar, Secure blockchain-based IoV architecture integrated with PUFs for secure vehicular communication, Comput. Electr. Eng. 123 (2025) 110069.
[17] V. Kohli, M.N. Aman, B. Sikdar, An intelligent fingerprinting technique for low-power embedded iot devices, IEEE Trans. Artif. Intell. 5 (9) (2024) 4519–4534.
[18] S. Jain, A.R. Korenda, B. Cambou, C. Lucero, Secure content protection schemes for industrial IoT with SRAM PUF-based one-time use cryptographic keys, in: Science and Information Conference, Springer, 2024, pp. 478–498.
[19] K. Mahmood, S. Shamshad, M.A. Saleem, R. Kharel, A.K. Das, S. Shetty, J.J. Rodrigues, Blockchain and PUF-based secure key establishment protocol for cross-domain digital twins in industrial internet of things architecture, J. Adv. Res. 62 (2024) 155–163.
[20] M. Barbareschi, F. Cirillo, C. Esposito, SRAM-PUF authentication schemes empowered with blockchain on Resource-Constrained microcontrollers, in: 2024 IEEE 27th International Symposium on Real-Time Distributed Computing, ISORC, IEEE, 2024, pp. 1–10.
[21] C. Jiang, X. Li, D. Du, L. Wu, R. Findeisen, Cross-domain authentication scheme based on distributed two-layer collaborative blockchains for cyber–physical power systems, IEEE Internet Things J. 11 (10) (2024) 17468–17481.
[22] Y. Xu, Z. Peng, C. Zhang, G. Wang, H. Wang, H. Jiang, Y. Zhang, Enhancing privacy in cyber–physical systems: An efficient blockchain-assisted data-sharing scheme with deniability, J. Syst. Archit. 150 (2024) 103132.
[23] F. Wang, J. Cui, Q. Zhang, D. He, H. Zhong, Blockchain-assisted flexible revocable anonymous authentication in industrial internet of things, IEEE Trans. Netw. Sci. Eng. (2024).
[24] Y. Gao, S. Al-Sarawi, D. Abbott, Physical unclonable functions, Nat. Electron. 3 (2020) 81–91.
[25] E.T. Michailidis, D. Vouyioukas, A review on Software-Based and Hardware-Based authentication mechanisms for the Internet of Drones, Drones (2022).
[26] J. Dey, R. Dutta, Progress in multivariate cryptography: Systematic review, challenges, and research directions, ACM Comput. Surv. 55 (2022) 1–34.
[27] Z. Huang, Q. Wang, Y. Chen, X. Jiang, A survey on machine learning against hardware trojan attacks: Recent advances and challenges, IEEE Access 8 (2020) 10796–10826.
[28] K. Agrawal, M. Aggarwal, S. Tanwar, G. Sharma, P. Bokoro, R. Sharma, An extensive blockchain based applications survey: Tools, frameworks, opportunities, challenges and solutions, IEEE Access 10 (2022) 116858–116906.
[29] N. Wang, J. Fu, S. Zhang, Z. Zhang, J. Qiao, J. Liu, B.K. Bhargava, Secure and distributed IoT data storage in clouds based on secret sharing and collaborative blockchain, IEEE/ACM Trans. Netw. 31 (4) (2022) 1550–1565.

[30] Y. Zhang, B. Li, J. Wu, B. Liu, R. Chen, J. Chang, Efficient and privacy-preserving blockchain-based multifactor device authentication protocol for cross-domain IIoT, IEEE Internet Things J. 9 (22) (2022) 22501–22515.
[31] G. Li, Y. Wang, B. Zhang, S. Lu, Smart Contract-Based Cross-Domain authentication and key agreement system for heterogeneous wireless networks, Mob. Inf. Syst. 2020 (1) (2020) 2964562.
[32] M. Shen, H. Liu, L. Zhu, K. Xu, H. Yu, X. Du, M. Guizani, Blockchain-assisted secure device authentication for cross-domain industrial IoT, IEEE J. Sel. Areas Commun. 38 (5) (2020) 942–954.
[33] S. Zhang, X. Du, X. Liu, A novel and quantum-resistant handover authentication protocol in IoT environment, Wirel. Netw. 29 (6) (2023) 2873–2890.