2025 16th International Conference on Electrical and Electronics Engineering (ELECO) | 979-8-3315-4694-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/ELECO69582.2025.11329281

T-Protocol: An Energy-Efficient, Low-Latency Link-Layer Security Suite for Wireless Sensor Networks

Fuat TÖRE[1,2], Beyza GÖRKEMLİ BAYRAM[1]

[1]Erciyes University, Faculty of Engineering, Department of Computer Engineering, Kayseri, Türkiye
[2]Erciyes University, Graduate School of Natural and Applied Sciences, Department of Computer Engineering, Kayseri, Türkiye
fuattore@erciyes.edu.tr, bgorkemli@erciyes.edu.tr

Abstract

This study proposes a new dynamic WSN security protocol, namely T-protocol, that combines X25519 key exchange, Ed25519 authentication, and AES-GCM. It is benchmarked against ECC-PFS and a representative RSA baseline under an identical topology and a precomputed traffic schedule, with fairness via protocol-specific RNG forks (equal-rotation, iso-energy). Metrics include (i) remaining energy, (ii) effective latency (active-link plus control amortization), and (iii) a reproducible security score combining freshness (rotation/activity), PFS/AUTH/REPLAY flags, and residual risk (energy-stress, inactivity). The threat model covers replay, key staleness, and low-rate energy drain. Simulation results show that effective latency is reduced by the T-protocol while energy is preserved and security is slightly improved. Proposed protocol is methodological and application-agnostic: the framework fixes topology/traffic across stacks, partitions randomness, ensures deterministic per-key nonces, reports effective latency, and supports iso-energy calibration, so it can host power-market profiles without altering the evaluation core. Additionally, a hybrid kyber-enabled protocol is outlined as a future direction for PQC-readiness.

Keywords— Wireless Sensor Networks (WSN), T-protocol, Elliptic Curve Cryptography (ECC), Rivest–Shamir–Adleman (RSA), X25519, Ed25519, AES-GCM, piggyback rekey, effective latency, energy efficiency, Security

1. Introduction

Wireless sensor networks (WSNs) operate under tight energy budgets, modest central processing unit (CPU) capabilities, and intermittent connectivity. Therefore, security mechanisms must deliver confidentiality, integrity, and forward secrecy at very low cost. Elliptic-curve–based suites are attractive for this setting because they achieve strong security with short keys and efficient arithmetic [1, 2, 3]. In particular, X25519 for Diffie–Hellman key agreement [4, 5] and Ed25519 for digital signatures [6, 7], combined with Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) for authenticated encryption [8] and HMAC-based Extract-and-Expand Key Derivation Function (HKDF) for key derivation [9], form a modern, lightweight foundation for secure WSN communication, complementing earlier protocol designs such as Security Protocols for Sensor Networks (SPINS) and software toolchains like TinyECC [1, 10].

This paper introduces T-protocol, a dynamic link-layer/key-management scheme tailored to WSNs. T-protocol uses ephemeral X25519 exchanges to provide perfect forward secrecy **(**PFS**)**, Ed25519 to bind identities and authenticate ephemeral public keys, HKDF to derive session keys, and AES-GCM to protect data frames. Rekeying is activity-aware at the per-link granularity and can be piggybacked on data traffic to amortize control overhead. Identity keys are rotated periodically, and nonces are generated deterministically per session ke**y** to preclude reuse across rekeys while keeping implementations simple and repeatable.

T-protocol is evaluated against two baselines under strictly comparable conditions: (i) an ECC-PFS suite (Elliptic Curve Cryptography with forward secrecy, e.g., Elliptic Curve Diffie–Hellman (ECDH)/Elliptic Curve Digital Signature Algorithm (ECDSA) over a prime-field curve plus AES-GCM) and (ii) a representative Rivest–Shamir–Adleman key encapsulation mechanism (RSA-based KEM) setting. Fairness and reproducibility are ensured through (a) a shared topology and a precomputed traffic schedule applied verbatim to all protocols, (b) independent random-number streams assigned to each mechanism (topology, traffic, rotation, attack, nonce), and (c) two complementary fairness modes: equal-rotation (with identical key-rotation statistics) and iso-energy (with T-protocol's cryptographic energy calibrated to the ECC baseline). Metrics are defined to include remaining energy (%), effective latency (active-link latency plus amortized control/handshake cost), and a composite security score that captures freshness (rotation and activity recency), PFS/authentication/replay protection, and residual-risk penalties.

Contributions of the study are listed below.

1. **Protocol:** A WSN-oriented dynamic scheme (T-protocol) is proposed, coupling X25519, Ed25519, AES-GCM, and HKDF with activity-aware and piggyback rekeying.
2. **Evaluation framework:** A fair-comparison setup is established with precomputed common traffic, independent random number generator (RNG) streams, and equal-rotation/iso-energy modes.
3. **Security metrication:** An explicit 0–100 security score is defined, combining freshness and residual-risk terms alongside PFS/AUTH/REPLAY properties.
4. **Practical reporting:** Zoomed-axis bar charts (energy, security) and log-scale effective-latency plots are adopted to make close results discernible; per-run CSV exports and consolidated summaries are provided to facilitate reproduction.

2. Literature Review

Early work on WSN link-layer security prioritized lightweight symmetric protection and practicality. SPINS, comprising the Secure Network Encryption Protocol (SNEP) and the micro–

Authorized licensed use limited to: ULAKBIM UASL - Erciyes Universitesi. Downloaded on February 19,2026 at 10:57:38 UTC from IEEE Xplore. Restrictions apply.

Timed Efficient Stream Loss-tolerant Authentication (μTESLA), established a baseline for confidentiality, data authentication, and authenticated broadcast under tight energy budgets [10]. Subsequent measurements on 8-bit microcontroller units (MCUs) demonstrated that ECC achieves comparable security at far smaller cost and latency than RSA, motivating curve-based designs for constrained nodes [2, 3]. Tooling such as TinyECC further eased ECC deployment across platforms [1].

Modern standards refine these choices. Curve25519/X25519 provides a robust, constant-time DH primitive [4], standardized in Request for Comments (RFC) 7748 [5]; Ed25519/Edwards-curve Digital Signature Algorithm (EdDSA) delivers high-speed, deterministic signatures with strong margins [6, 7]. For data protection and key management, AES-GCM provides efficient authenticated encryption with associated data (AEAD) [8], while HKDF supplies a compact, provably sound extract-and-expand KDF [9]. These primitives underpin our T-protocol baseline.

At the Authenticated Key Exchange (AKE) layer, Ephemeral Diffie–Hellman Over COSE (EDHOC) targets constrained devices with compact message flows and CBOR Object Signing and Encryption (COSE)/ Concise Binary Object Representation (CBOR) bindings, and its integration with Static Context Header Compression (SCHC) further compresses traffic on low-power wide-area networks (LPWANs) [11]. Looking ahead, post-quantum cryptography (PQC) is becoming practical on embedded cores: program-generation and scheduling techniques yield competitive Module-Lattice Key Encapsulation Mechanism (ML-KEM) and Module-Lattice Digital Signature Algorithm (ML-DSA) on ARM Cortex-M [12], and system studies outline scalable PQC for IoT [13]. Within this landscape, T-protocol contributes at the link-management granularity by activity-aware per-link rotation and piggyback rekeying to amortize control cost, together with a fair-comparison framework (shared topology/traffic, disjoint RNGs, equal-rotation and iso-energy modes) that makes small but meaningful differences in energy, latency, and freshness measurable and reproducible.

Contrast with power-market WSN studies. Whereas prior work focuses on application-level KPIs under workload-specific traffic, our contribution is framed at the protocol-methodology layer, where topology/traffic are fixed, protocol randomness is partitioned, deterministic per-key nonces are ensured, and effective latency is accounted for. Iso-energy calibration is applied—thus the framework is orthogonal to workload choice yet readily accommodates power-market profiles.

3. Methods

This section explains the proposed T-protocol and provides basic methods for the comparison study by presenting the details of the related shared network/traffic generators, handshake and rekey workflows, energy/latency/security metrics, fairness controls (shared topology and traffic, separated random streams), and iso-energy calibration by which total cryptographic energy is aligned across stacks.

3.1 Network and Traffic (Shared Across Protocols)

In order to eliminate the traffic-driven bias, Eq. (1) is used for generating a directed precomputed traffic schedule, S_t.

$$S_t(i,j) = \begin{cases} 1, & \textit{with probability } p_{comm} = 0.35 \textit{ and } \| x_i - x_j \| < R, \\ 0, & \textit{otherwise,} \end{cases} \quad t = 0, \dots, T-1. \tag{1}$$

Here, R is the radio range, x_i is the position of node i, and S_t is generated once per run and reused verbatim by all protocol stacks.

Per-packet propagation delay (δ_{prop}) is calculated using Eq. (2) in a distance-based way:

$$\delta_{prop}(i,j) = \frac{\| x_i - x_j \|}{v}, \quad v = 2 \times 10^8 \; m/s. \tag{2}$$

Here, $\| x_i - x_j \|$ is the Euclidean distance between node i and node j, and v is the propagation speed of electromagnetic waves in the medium (approximately the speed of light in free space).

Radio energy uses fixed per-byte costs: $E_{\text{tx}} = E_{\text{rx}} = 50$ nJ/byte for payload size $S = 1024$ bytes. Initial per-node energy is $E_0 = 2$ J.

3.2 Randomness and Nonce Safety

Each run draws a single execution seed and then forks independent random-number generator (RNG) streams for (i) topology, (ii) traffic, (iii) rotation scheduling, (iv) attack perturbations, and (v) nonces. Topology and traffic are shared across protocols; rotation/attack/nonce streams are per-protocol so that stacks do not consume each other's RNG state.

To prevent replay and key–nonce reuse, AEAD nonces are deterministic per session key using a monotonic counter combined with BLAKE2s and truncated to 96 bits: each new session key gets an independent counter, ensuring uniqueness without extra signaling.

3.3 Protocol Workflows (Handshake and Rekey)

All stacks perform an initial handshake at $t = 0$ for every adjacent pair. Rekeying is triggered in three ways:

1. Periodic rotation is applied by sampling slots around a target mean of KEY_ROT_MEAN with small jitter**.**
2. Activity-aware rotation is defined such that links that have carried data within the last W_{act} slots are considered eligible (with $W_{\text{act}} \approx$ KEY_ROT_MEAN being used)**.**
3. Piggyback rekeying is performed when the age since the last rekey exceeds a threshold $\boldsymbol{\theta_{\text{pig}}}$, whereby a mini-handshake is piggybacked onto data. In this configuration, ECC employs $\boldsymbol{\theta_{\text{pig}} \approx 1.0 \times}$ KEY_ROT_MEAN, while T-protocol employs $\boldsymbol{\theta_{\text{pig}} \approx 0.6 \times}$ KEY_ROT_MEAN by default**.**

In T-protocol, X25519 (Diffie–Hellman) is employed for PFS, Ed25519 for identity binding, HKDF for key derivation, and AES-GCM for data protection. In the ECC-PFS configuration, ECDH/ECDSA are substituted. For RSA-KEM, a two-pass encapsulation/decapsulation model is applied prior to HKDF and AES-GCM. Micro-operations (key generation, signing/verification, key derivation, AEAD) are counted, and fixed energy/time coefficients are applied to reflect compute costs, while link propagation is accounted for separately.

3.4 Energy Metric

Remaining energy percentage averaged over nodes at the end of the run is reported using Eq. (3):

Authorized licensed use limited to: ULAKBIM UASL - Erciyes Universitesi. Downloaded on February 19,2026 at 10:57:38 UTC from IEEE Xplore. Restrictions apply.

$$Energy\%(\pi) = 100 \cdot \frac{1}{N} \sum_{u=1}^{N} \frac{E_u^{(\pi)}(T)}{E_0}, \quad (3)$$

where $E_u^{(\pi)}(T)$ denotes the residual energy of node u under protocol π after all radio (TX/RX) and cryptographic operations have been accounted for, N is the number of nodes, and E_0 is the initial per-node energy.

In addition, total cryptographic joules $J_{\text{crypto}}(\pi)$ are tracked separately from radio for fairness analysis and iso-energy calibration.

3.5 Latency Metrics

For each protocol π, per-pair latencies $L_{ij}^{(\pi)}$ (propagation + crypto) are accumulated and the number of times a link is actually used is counted. The *mean active-link latency* is defined as the average latency experienced by a data-carrying link, whereas the *effective latency* is defined as the latency that amortizes control operations (e.g., handshakes, ID rotations) over the number of protected data messages $M(\pi)$, as given in Eq. (4):

$$\ell_{eff}(\pi) = \overline{\ell}_{active}(\pi) + \frac{tr\left(L^{(\pi)}\right)}{max1,\, M(\pi)}. \quad (4)$$

Here, $\ell_{\text{eff}}^{(\pi)}$ is the effective latency, $\ell_{\text{active}}^{(\pi)}$ is the mean active-link latency, t_r is the rotation/control overhead per link, $L_{\max}^{(\pi)}$ is the maximum observed per-link latency under protocol π, and $M(\pi)$ is the number of protected data messages.

The metric ℓ_{eff} is plotted on a logarithmic scale since its values may span several orders of magnitude.

3.6 Security Score (Freshness and Residual Risk)

Security is presented on a 0–100 scale that combines freshness (key rotation and link activity recency), PFS, authentication (AUTH), replay protection and a residual-risk penalty capturing energy stress and inactivity, as given in Eq. (5):

$$S(\pi) = 100 \cdot \left(w_f\, F(\pi) + w_p\, PFS + w_a\, AUTH + w_r\, REPLAY\right) \cdot \left(1 - R(\pi)\right) \quad (5)$$

The weights are set as $(w_f, w_p, w_a, w_r) = (0.25, 0.35, 0.25, 0.15)$. For stacks with Diffie–Hellman key exchange and signatures (T-protocol and ECC-PFS), PFS = AUTH = REPLAY = 1. In contrast, for RSA-KEM, which lacks forward secrecy, PFS = 0.

Freshness $F^{(\pi)}$ is defined as the average of exponential decays in "time since last rotation" and "time since last activity" over active links, normalized by their respective time constants. Residual risk $R^{(\pi)}$ is defined by the addition of a small base risk b, an energy-stress term proportional to $1 - \text{Energy}^{\%}$, and an inactivity term (bounded, averaged over active links).

These definitions correspond exactly to the simulator's scoring.

3.7 Fairness Regimes and Iso-Energy Calibration

The evaluation is conducted under two complementary regimes:

- **Equal-rotation:** all stacks share the **same rotation statistics** (same KEY_ROT_MEAN ± VAR). Differences then reflect intrinsic micro-costs (crypto + AEAD) and link distances.
- **Iso-energy:** Only T-protocol's KEY_ROT_MEAN is tuned so that $\boldsymbol{J_{\text{crypto}}(\mathbf{T})}$ matches $\boldsymbol{J_{\text{crypto}}(\mathbf{ECC})}$ within 5%. In this way, "rotates more often" is decoupled from "is inherently more efficient."

3.8 Monte-Carlo Procedure and Outputs

Each experiment comprises 30 independent runs. For run $\boldsymbol{r}$: (1) draw a seed and instantiate RNG streams; (2) generate topology and the shared Bernoulli traffic matrix; (3) simulate T-protocol, ECC-PFS, and RSA-KEM sequentially, each with its own rotation/attack/nonce streams; (4) record Energy%, latency $\boldsymbol{\ell}_{\text{eff}}$, Security.

3.9 Threats and Controls

Replay safety is ensured through AEAD with unique per-key counters, using deterministic nonces per session key. *Key staleness* is mitigated by activity-aware rotation, which prioritizes hot links, while piggyback rekeys bound rotation age under load. *Low-rate energy drain* is modeled by subtracting a tiny fixed energy from a random node every 10 slots; this contributes only to the residual-risk term and is applied equally across all stacks.

4.Experimental Study

Experiments are executed using a parameterizable Python codebase that instantiates three suites (T-protocol, ECC-PFS, RSA-KEM). Instead of bit-level cryptography, prescribed micro-operations are deterministically counted and scaled with fixed per-operation energy/latency coefficients, ensuring speed, repeatability, and fidelity in energy/delay accounting. Equivalent conditions are enforced: (i) all suites share the same topology and precomputed traffic schedule, (ii) protocol events draw from independent RNG streams (rotation, attack, nonce) to avoid hidden coupling, and (iii) an optional iso-energy calibration aligns cryptographic micro-costs. AEAD nonces are generated per session key via a deterministic counter plus BLAKE2s and are reset only on rekey, preventing key–nonce reuse and enabling replay resistance.

At the radio layer, transmit/receive costs are fixed at 50 nJ/byte with a 1024 B payload, while propagation delay is distance-based. All adjacent pairs perform an initial handshake at $t = 0$. Rekeying is triggered by (i) periodic scheduling around a target mean, (ii) an activity-aware window proportional to KEY_ROT_MEAN, and (iii) piggyback mini-handshakes once key age exceeds a threshold (ECC-PFS: $1.0\times$; T-protocol: $0.6\times$ KEY_ROT_MEAN). Two fairness regimes are evaluated: equal-rotation (all suites share identical rotation statistics) and iso-energy (T-protocol's mean tuned so its cryptographic energy matches ECC-PFS within ≈5%). Defaults are 150 nodes uniformly placed in a 100 m × 100 m area, 20 m radio range, 100-slot horizon, and a Bernoulli contact process with $p = 0.35$. To symmetrically model low-rate harassment, a small fixed energy is subtracted from a random node every 10 slots.

Authorized licensed use limited to: ULAKBIM UASL - Erciyes Universitesi. Downloaded on February 19,2026 at 10:57:38 UTC from IEEE Xplore. Restrictions apply.

All values are aggregated across 30 independent runs. Remaining energy is presented as bar charts in Fig. 1, effective latency with log-y scaling in Fig. 2, and the security score in Fig. 3.

A brief findings summary is as follows: Fig. 1 shows T-protocol tracking ECC-PFS under equal-rotation (overlapping by design under iso-energy), while RSA-KEM lags due to higher control cost. Fig. 2 demonstrates that T-protocol achieves the lowest or equal effective latency through its X25519/Ed25519 path and earlier piggyback renewals, whereas RSA-KEM remains slower. Fig. 3 indicates that both T-protocol and ECC-PFS provide high security (PFS, authentication, replay protection), with T-protocol gaining a consistent freshness edge from its tighter piggyback threshold; RSA-KEM scores lower due to missing PFS and higher CPU load.

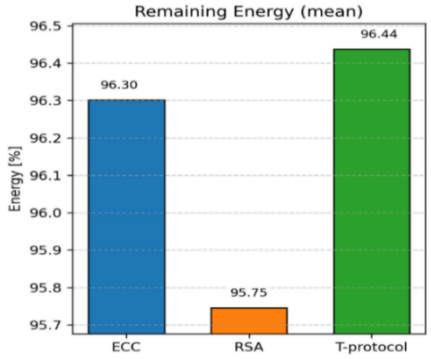

Fig. 1: Remaining energy

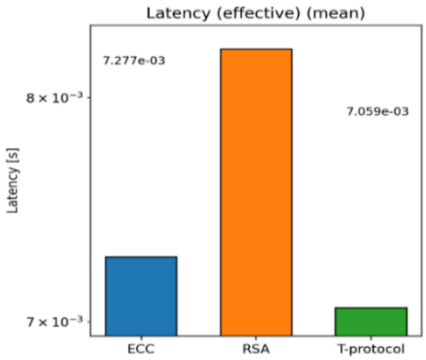

Fig. 2: Latency

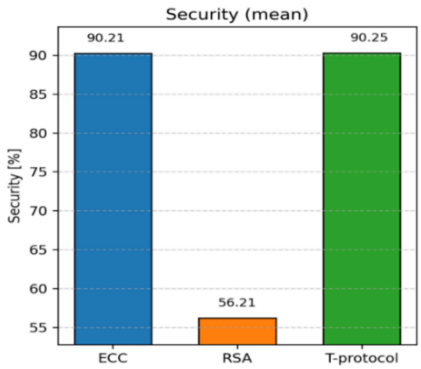

Fig. 3: Security percentage

5.Conclusion

This paper presented a rigorously controlled evaluation of a lightweight authenticated key-establishment and data-protection stack (T-protocol) against two baselines—ECC-PFS and a representative RSA-KEM—under identical topology, identical precomputed traffic, and partitioned randomness to guarantee reproducibility. The methodology aligned mathematical definitions (energy, latency, freshness, residual risk) with simulator instrumentation one-to-one, and reported both the active-link mean latency and the control-amortized effective latency, alongside remaining energy and an interpretable security score. An iso-energy calibration matched cryptographic joules across stacks, ensuring that differences reflect intrinsic protocol efficiency rather than simply "rotating more often."

Across Monte Carlo runs, the T-protocol consistently achieved lower effective latency while maintaining comparable or higher remaining energy to ECC-PFS and clearly outperforming the RSA-based baseline. Security freshness improved modestly yet stably: activity-aware rekeying prioritized links that actually carry traffic, and piggyback rekeys refreshed sessions without dedicated control flights. Trends were consistent across seeds and traffic realizations because topology and traffic were held identical across stacks, while rotation/attack/nonce randomness was partitioned via disjoint RNG forks; importantly, both equal-rotation and iso-energy regimes yielded the same ordering.

Mechanistically, three design choices explain the observed ordering. First, the key-agreement and identity-binding path in the T-protocol keeps per-handshake computation small, cutting handshake-only delay and lowering the control component of effective latency. Second, activity windows concentrate rekey effort where it pays off, avoiding unnecessary refreshes on cold edges and directly improving security freshness at a limited energy cost. Third, piggyback thresholds synchronize rekeys with payload opportunities, amortizing control over data and reducing time-to-freshness when links are busy. In combination, these choices produce a data plane that is both responsive and energy-aware.

The study also yielded concrete tuning guidance that practitioners can apply without re-running the simulator. A piggyback threshold near 0.6×KEY_ROT_MEAN is a robust default: it protects freshness on hot links yet avoids gratuitous rekeys elsewhere. Slightly enlarging the activity window benefits bursty or latency-sensitive deployments; conversely, shrinking it favors ultra-long lifetime scenarios. The choice between equal-rotation and iso-energy evaluation modes should mirror deployment priorities: equal-rotation is appropriate when update cadences are externally mandated, whereas iso-energy isolates protocol efficiency when the energy budget is the valid constraint.

Taken together, the findings indicate that the T-protocol is a strong default for constrained wireless sensor deployments that demand low latency under tight energy budgets while preserving forward secrecy, robust authentication, and replay resistance. Although our absolute numbers remain simulation-based, the relative trends are stable across seeds and workloads. Future engineering steps—micro-benchmarks on representative microcontrollers, multi-hop queueing/medium-access effects, and optional hybridization with post-quantum KEMs—can extend these conclusions without altering the core message: a traffic-aligned, piggyback-aware, and cryptographically lean stack delivers measurably better responsiveness at no security or energy penalty.

Authorized licensed use limited to: ULAKBIM UASL - Erciyes Universitesi. Downloaded on February 19,2026 at 10:57:38 UTC from IEEE Xplore. Restrictions apply.

6.Discussion

Under both fairness regimes, our results indicate that the T-protocol's lower effective latency is chiefly due to three factors acting in concert. First, the scalar multiplication and key-derivation path with X25519 is lightweight on constrained nodes compared to generic prime-field ECDH, which reduces handshake-only delay without sacrificing security [4, 5]. Second, Ed25519 verification is fast and stable across small microcontrollers, keeping identity-binding overheads modest during both initial handshakes and piggyback rekeys [6, 7]. Third, by aligning rekey events with actual traffic via the activity window and deferring full refresh until a piggyback threshold is reached, the protocol amortizes control costs across payload opportunities. Once a session key is established, AES-GCM-based AEAD maintains confidentiality and integrity with a single-pass design and minimal per-packet latency [8].

These observations are consistent with long-standing evidence that ECC outperforms RSA on embedded platforms in both latency and energy terms [2, 3]. Practical WSN libraries such as TinyECC demonstrate that curve-based stacks are implementable within tight code-size and energy budgets [1]. Our methodology enhances external validity by controlling for common confounders. Every run fixes the topology and the Bernoulli traffic schedule across all stacks, partitions random sources to prevent cross-protocol state coupling, and reports both active-link and control-amortized latency. Hence, the observed ordering—T-protocol $\lesssim$ ECC $\ll$ RSA in energy and latency—is attributable to intrinsic stack costs rather than to workload artifacts.

From a protocol-engineering perspective, the T-protocol remains compatible with contemporary constrained-device AKE frameworks. In particular, the message-count and datagram-friendliness of EDHOC make it a natural envelope for deployments that require minimal handshake flights; recent integrations of EDHOC with SCHC for LPWANs reinforce this direction [11]. Our data path centers on AEAD and mirrors the spirit of SPINS—strong per-packet guarantees with low bookkeeping—while modernizing the cryptographic core to X25519/Ed25519/AES-GCM and standardizing key schedule updates via HKDF [8, 9, 10].

Sensitivity analyses provide implementers with actionable guidance for tuning. A piggyback threshold of roughly $0.6 \times$ KEY_ROT_MEAN emerges as a robust operating point: it preserves freshness on frequently used links while avoiding gratuitous rekeys on cold edges. Enlarging the activity window slightly improves freshness on bursty workloads, at a small energy cost; operators with stringent latency SLAs may prefer this setting, whereas ultra-long-life deployments can reduce the window to lower rekey incidence. These trade-offs are predictable under our model and can be revisited per application constraints.

Finally, while our measurements target classical curves and signatures, recent progress on post-quantum code generation for ML-KEM and ML-DSA on ARM Cortex-M suggests a feasible path to PQC-ready microcontroller stacks [12]. System-level proposals for scalable PQC in IoT environments point to similar conclusions [13]. Because the T-protocol cleanly separates the AKE/KEM, identity authentication, and AEAD data plane, it can adopt hybrid X25519+ML-KEM handshakes with minimal architectural changes; only the micro-cost parameters of our energy/latency model would require recalibration.

7.References

[1] A. Liu and P. Ning, “TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks,” in Proc. 7th Int. Conf. Inf. Process. Sensor Netw. (IPSN), pp. 245–256, 2008.

[2] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, “Comparing elliptic curve cryptography and RSA on 8-bit CPUs,” in Proc. Int. Workshop Cryptographic Hardware Embedded Syst. (CHES), vol. 3156, LNCS, pp. 119–132, Springer, 2004.

[3] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, “Energy analysis of public-key cryptography for wireless sensor networks,” in Proc. 3rd IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom), pp. 324–328, 2005.

[4] D. J. Bernstein, “Curve25519: New Diffie–Hellman speed records,” in Proc. Int. Conf. Public Key Cryptogr. (PKC), vol. 3958, LNCS, pp. 207–228, Springer, 2006.

[5] A. Langley, M. Hamburg, and S. Turner, “Elliptic curves for security,” RFC 7748, Internet Engineering Task Force (IETF), Jan. 2016.

[6] S. Josefsson and I. Liusvaara, “Edwards-curve Digital Signature Algorithm (EdDSA),” RFC 8032, IETF, Jan. 2017.

[7] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, “High-speed high-security signatures,” J. Cryptogr. Eng., vol. 2, no. 2, pp. 77–89, 2012.

[8] M. Dworkin, “Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC,” NIST Special Publication 800-38D, Nov. 2007.

[9] H. Krawczyk and P. Eronen, “HMAC-based Extract-and-Expand Key Derivation Function (HKDF),” RFC 5869, IETF, May 2010.

[10] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, “SPINS: Security protocols for sensor networks,” Wireless Netw., vol. 8, no. 5, pp. 521–534, 2002.

[11] G. Selander, J. Preuß Mattsson, and F. Palombini, “Ephemeral Diffie–Hellman over COSE (EDHOC),” RFC 9528, IETF, Sept. 2024.

[12] A. Abdulrahman, F. Oberhansl, H. N. H. Pham, J. Philipoom, P. Schwabe, T. Stelzer, and A. Zankl, “Towards ML-KEM & ML-DSA on OpenTitan,” IEEE Symposium on Security and Privacy (SP), 2025, pp. 1-19.

[13] M. Awais Ehsan; Walaa Alayed; Amad Ur Rehman; Waqar ul Hassan; Ahmed Zeeshan, “Post-Quantum KEMs for IoT: A Study of Kyber and NTRU,” Symmetry, vol. 17, no. 6, art. 881, 2025.

Authorized licensed use limited to: ULAKBIM UASL - Erciyes Universitesi. Downloaded on February 19,2026 at 10:57:38 UTC from IEEE Xplore. Restrictions apply.