Resilient IoT Authentication Against Physical Tampering and Side Channel Attacks

Waleed Khalid Al-Zubaidi
College of Biomedical Informatics, University of Information Technology and Communications, Baghdad, Iraq
dr.waleed.khalid@uoitc.edu.iq

Abdalrahman Fatikhan Ataalla
Department of Computer Engineering Techniques, College of Technical Engineering, University of Al Maarif, Al Anbar, Iraq
engrahumi@uoa.edu.iq

Huda Mohammed Alsayednoor
Shatt Al-Arab University College, Basra, Iraq
huda1994noor@gmail.com

Mahmood A. Al-Shareeda
Department of Information Technology, Management Technical College, Southern Technical University, Basrah, Iraq | College of Engineering, Al-Ayen University, Thi-Qar, Iraq
mahmood.alshareedah@stu.edu.iq (corresponding author)

Mohammed Almaayah
Department of Computer Science, King Abdullah the II IT School, The University of Jordan, Amman, Jordan
m.almaiah@ju.edu.jo

Marwan Albahar
Department of Computing, College of Engineering and Computing in Al-Lith, Umm Al-Qura University, Makkah, Saudi Arabia
mabahar@uqu.edu.sa

Received: 17 July 2025 | Revised: 26 August 2025, 3 September 2025, and 6 September 2025 | Accepted: 7 September 2025

 DOI: https://doi.org/10.48084/etasr.13464

ABSTRACT

The security of IoT devices operating in open and physically accessible areas is being directly jeopardized by sophisticated adversary methods such as side-channel, fault injection, and invasive attacks. However, common authentication techniques are effective for network-level protection but ineffective against these low-level physical attacks, rendering cryptographic keys and protocol logic susceptible to being extracted and subverted. This paper presents a lightweight, tamper-resistant authentication protocol for resource-constrained IoT devices. The proposed approach combines randomized masking, redundant response computation, fault injection detection, and tamper evidence in a five-phase technique to ensure reliable identity verification in a hostile environment. The proposed scheme provides mutual authentication, forward secrecy, and zeroization-based lockout protection without the need for specialized hardware or high computational overhead. Simulation results show that the protocol achieves complete authentication in 5.6 ms with less than 1.15 µJ per session and 1.3 KB of SRAM. The proposed technique is suitable for deployment in critical IoT applications where both logical and physical integrity are needed, as the additional overhead is small while offering better physical-layer resilience compared to current solutions.

Keywords-IoT security; authentication protocol; side-channel attack; fault injection; physical tampering; lightweight cryptography; tamper detection; secure embedded systems; leakage resilience; mutual authentication

I. INTRODUCTION

The rapid deployment of Internet of Things (IoT) devices within critical application domains, e.g., healthcare, defense, infrastructure, and industrial automation, has resulted in new challenges to securing resource-constrained hardware against traditional and physical-layer threats [1, 2]. Many such devices, which are used in uncontrolled or adversarial environment settings, are exposed not only to cyber threats (e.g., impersonation, replay) but also to invasive hardware-level threats in the form of fault injections, side-channel leakages, and physical tampering [3, 4]. Although this work often employs cryptographic protocols to secure data communication, most generic authentication techniques consider software adversaries and do not realize adversaries that have physical access to the internals of the device [5, 6].

In particular, Side-Channel Attacks (SCAs), such as Simple Power Analysis (SPA), Differential Power Analysis (DPA), and timing analysis, capitalize on fluctuations in power consumption, electromagnetic emanations, or timing of execution to recover secret data without impacting the correctness of the protocol [7, 8]. Such attacks require few implementation resources and are applicable in a non-invasive way with great success to quite popular cryptosystems [9]. In contrast, fault injection attacks, such as using voltage glitches, laser pulses, or clock/voltage manipulation, can cause transient errors to bypass authentication logic, corrupt secret states, or disable protection countermeasures [10, 11]. In addition to physical adversaries, physical tampering attacks may include an attacker who physically probes or disturbs a sensor to read memory or inject malicious instructions [12, 13]. If such avenues of attack are not handled, not even information-theoretic security guarantees can prevent the naive breach of protocol security [14, 15].

Distributed authentication schemes for the IoT have been proposed to reduce the computation, communication, and memory costs of IoT devices, along with lightweight cryptographic protocols. Such systems include hash-based challenge-response systems and pairwise key exchanges, as well as systems that take advantage of Physical Unclonable Functions (PUFs) [16] to extract device-unique entropy. Nevertheless, some of these techniques are still susceptible to physical side channels or do not have real-time fault and tamper detection. Second, hardware countermeasures such as masking, glitch detectors, and secure fuses are barely integrated in protocol-level design, and thus suffer from degradative security.

To overcome these issues, this study presents a secure authentication scheme that is resilient to physical tampering and SCAs, specifically designed for resource-constrained IoT settings. The protocol is implemented using the following five major steps: (i) secure key provisioning using leakage-resilient masking, (ii) encoding the challenge using a random challenge, (iii) validating responses redundantly with fault-detection, (iv) detecting any tampering by zeroization/lockout, and (v) mutual authentication using ephemeral session tokens. This layered architecture allows not only to achieve cryptographic soundness but also execution integrity and physical-layer robustness. The contributions of this study can be summarized as follows:

- Introduces a tamper-aware and side-channel-resilient authentication protocol specifically designed for ultra-constrained IoT devices.
- Integrates duplicate computation, anomaly detection, and environmental monitoring into the protocol to provide resistance against fault injections and physical access attacks.
- Conducts a comprehensive performance evaluation in terms of latency, energy, and memory utilization. The proposed scheme is compared with recent secure protocols, such as PUF-ZKP IoT and EnConvo, to determine its practical feasibility.
- Demonstrates the security of the proposed protocol against a comprehensive threat model that includes SCAs, fault injections, replay attacks, impersonation attempts, and invasive tampering.

II. RELATED WORK

The widespread use of IoT devices in hostile environments has triggered a great deal of research in lightweight and physically secure authentication mechanisms. Prior attempts have largely focused on cryptographically minimized PUFs and basic anti-replay designs. Attention is paid to complete protection from both side channel leakage and active physical tampering.

In [17], a CNN-based authentication architecture was proposed for IoT devices with SRAM PUF responses. This method used Efficient Net-Lite and Mobile Net to classify noisy/corrupt responses on edge devices, offering reliable, hardware-free security by machine learning confidence scoring for error-tolerant real-time authentication. REMI-DLGKM [18] is a lightweight decentralized group key management framework combined with a secure multicast routing protocol for IoT. It is tailored to dynamic distributed systems, offering better scalability, energy stability, and reliability. URMAP [19] is an ultralight RFID authentication protocol that addresses the weaknesses of previous UMAPs. It replaced weak logical operations with the Per-XOR operation and the Inverse Per-XOR operation, and improved the ability of anti-clone, anti-traceability, and anti-DOS attacks. In [20], a hybrid IoT model was proposed for smart agriculture by integrating the BCEER routing protocol with a PUF-based authentication mechanism. BCEER achieves energy efficiency through adaptive clustering, while PUFs provide lightweight hardware-level security. In [21], a lightweight PUF-based authentication protocol was designed for smart manufacturing IoT systems to ensure confidentiality, anonymity, and forward secrecy with minimal computational overhead.

In [22], an AI-powered IoT authentication technique used device-specific energy consumption profiles. Running on Raspberry Pi 5, it used predictive analytics to identify anomalies, providing scalable, non-invasive security. In [23], a secure access control system was presented for IoT-enabled environmental monitoring with IOTA and CP-ABSC. The privacy and integrity of a system were guaranteed by signing permission tokens. In [24], a hybrid secure authentication scheme for IoT used RF-based and device PUFs, providing enhanced replay resistance and protection against CRP exposure by binding SRAM-PUF responses to CSI-derived RF characteristics. The IoT Security Metamodel (IoTSM) [25] aimed to provide potential measures to address the lack of comprehensive IoT security frameworks for smart cities. Based on a metamodel, IoTSM incorporates several important features such as authentication, device integrity, intrusion prevention, and secure communication.

Recently, various lightweight authentication protocols have been proposed for constrained devices. Unfortunately, protocols such as LMAP and HB+ are prominent examples of being simple and computationally secure but providing only very limited security against physical probing or timing-based leaks. Sophisticated ones, such as PUF-based authentication (e.g., PUF-ZKP IoT [26]), use device-specific entropy to avoid the need for storing keys. PUFs are resistant to cloning but susceptible to modeling and are usually not very robust against side-channel analysis unless properly protected.

Other protocols used Trusted Execution Environments (TEEs), e.g., ARM TrustZone, to protect messengers [27]. Such solutions give partial physical protection, depend on the implementation, and do not directly target both fault injection and low-level tampering. In the same vein, protocols featuring ECC (e.g., EBIAS) offer strong cryptographic security, but at the price of high computational and energy overhead, which is not suitable for sub-1 mW IoT deployments. Several works have focused on individual side-channel countermeasures (e.g., masking, dual-rail logic, or constant-time execution), but integration of these defenses in a complete authentication stack is rare. Similarly, tamper detection has been investigated through the use of environmental sensing (such as voltage, light, or temperature), but few designs include a threat model that is defined formally as a lockout or recovery model.

This study makes a connection between cryptographic soundness and physical adversarial resilience. Compared to previous ones, the proposed authentication mechanism adopts a multi-layered (e.g., masked delay, duplicated computation, anomaly detection, environmental monitoring, and zeroization-based lockout) security architecture, providing advantages such as design simplicity. This scheme is not only characterized by low computational and memory overhead, but is also effectively secure against side-channel, fault injection, replay, and impersonation attacks, along with invasive tampering. With competitive performance (e.g., 5.6 ms latency, 1.15 µJ energy, 1.3 KB SRAM), the scheme shows feasibility for practical ultra-constrained IoT deployments. Thus, the novelty of this work is that it capitalizes on the coexistence of lightweight efficiency and physical-layer security strength, which are seldom investigated in previous works.

III. SYSTEM AND MODEL ATTACKS

A. System Model

The considered system is composed of resource-constrained IoT nodes deployed in a physical environment where adversaries can have physical access or listen to side-channel emissions. Each of these devices contains a Microcontroller Unit (MCU), limited dynamic memory, and a cryptographic accelerator (e.g., hash engine), and one or more sensors or RF transceivers to communicate with a trusted verifier or gateway, as shown in Figure 1.

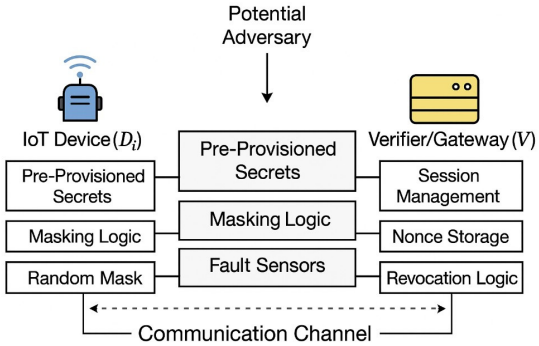

Fig. 1. System model.

The system consists of the following building blocks:

- IoT Device (D_i): Secure authentication is carried out by a low-power device that comes with pre-stored secrets, tamper sensors, and an obfuscation logic to facilitate secure operation.
- The Verifier (V) challenges the device, validates responses, and observes protocol consistency.
- Communication channel: It is mostly (wireless), e.g., BLE, LoRa, Zigbee, etc. The channel is considered insecure, observable, but authenticated with the help of protocol-level measures.

The protocol does not depend on the fact that the device can compute using only low power (e.g., XOR, hash), generate random masks, and run emergency code to gather information from the sensors onboard. The verifier retains session counters, windows of blacklisted nonce values, and revocation logic, relieving the device of additional storage requirements.

B. Threat Model

The protocol aims to protect against attackers who have logical and physical access abilities. The attacker may attempt to:

- Watch side channels, including power, EM (Electro-Magnetic) emissions, or timing to derive key-dependent information through SPA, DPA, or timing attacks [28-30].
- Inject faults: Apply voltage glitches, clock tweaks, laser pulses, or Electromagnetic (EM) noise to disturb the logic, corrupted memory details, or fault more gates and security checks [31-33].

- Tamper with the device by probing, decapsulation, micro-probing, or opening the chip package to extract secrets or to tamper with the behavior of the circuit.
- Replay messages or use impersonation to reuse compromised authenticator messages using emulation attacks to trick the verifier.
- The adversary is assumed to control communication channels in full and have temporary physical access to the device, constrained by time and being unable to perform real-time silicon reverse engineering.
- Privacy of secrets: Using bit masking and one-time keys ensures that the keys are not revealed by eavesdropping.
- Soundness of execution: The fault detection logic blocks redundant computations.
- Tamper response: It is necessary to design devices to detect tampering attempts to prevent secrets from being compromised, and then to enter into lockout or zeroization.
- Freshness: Against replays, an attacker can be thwarted by nonces and session tokens (that are time-bound).

IV. PROPOSED AUTHENTICATION PROTOCOL

The proposed protocol goes through five highly interactive phases, which in turn address a particular category of physical and side-channel threats for lightweight IoT devices. As shown in Figure 2, in the first phase, both secure boot and key provisioning are performed, ensuring that each chip receives a unique identity and cryptographic keys in leak-resilient masked form. Phase 2 presents two new techniques, randomized-challenge-masking and encoding, which disrupt the correlation of side-channel signals during computation with new entropy and non-linear transformations. Stage 3 focuses on the secure response generation, combined with the redundancy of the invocations and their timing check, to identify faults and avoid glitches to apply the bypass. In Phase 4, it passively checks voltage, temperature, and EM characteristics at the environmental and hardware levels for signs of physical tampering and activates lockdown routines on detection of threats. Finally, Phase 5 completes the mutual verification by deriving a session token with nonce-based exchange to guarantee an ephemeral, forward-secure communication. In aggregate, these phases culminate in a hierarchical hardware-aware defense strategy appropriate for the high-threat setting of physical and side-channel adversaries.

A. Secure Initialization and Key Provisioning

During this phase, sufficiently secure device identities are assigned to the IoT edge devices. This phase makes the cryptographic keys unique, unextractable, and compact enough to be secure while being computed. The delivery is performed in a known environment.

1. Unique Device Identifier Generation: Each device D_i is assigned a unique identifier ID_i, derived from either PUF response $ID_i = PUF_i()$ or secure hash of device entropy $ID_i = H(entropy_i)$.
2. Key Derivation: A symmetric authentication key K_i is generated using $K_i = KDF(ID_i \parallel AppTag \parallel nonce)$, where $KDF(\cdot)$ is a key derivation function, and $AppTag$ denotes application-specific binding.
3. Boolean Masking for Leakage Resilience: The key K_i is split into a masked representation $(K_i(1), K_i(2))$ such that $K_i = K_i(1)$ XOR $K_i(2)$. Only the masked shares and are stored in device memory to reduce correlation with real power/EM signals.
4. Storage in Hardened Registers: The masked key shares are stored in hardware-level dual-rail logic or registers protected with: Glitch-detection latches; Redundant encoding (e.g., Hamming codes); and Tamper-evident fuses (optional).
5. Provisioning Certificate Generation (Optional): For devices requiring remote identity verification, a provisioning certificate is generated: $Cert_i = SignCA(ID_i \parallel K_i \parallel Expiry)$, where $SignCA(\cdot)$ is a digital signature issued by a trusted certificate authority.

This bootstrap protocol ensures that each device is provisioned with a distinct leakage-masked identity and key pair, thus beginning life resistant to SCA and physical probing.

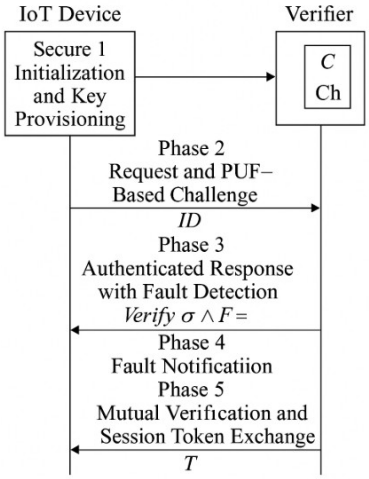

Fig. 2. Proposed authentication protocol.

B. Challenge Masking and Randomized Encoding

To protect against side-channel leakage during challenge processing, the protocol leverages randomized masking and non-deterministic encodings before any cryptographic computation. This is done to ensure that the power, EM, or timing patterns are essentially statistically independent of the sensitive inputs.

1. Challenge Reception: Upon initiation, the verifier sends a random challenge $C \in \{0,1\}$ to the device D_i: $V \rightarrow D_i$: C.
2. Random Mask Generation: The device generates a fresh random mask $R \in \{0,1\}^n$ using a secure PRNG: $R \leftarrow PRNG()$.

3. Masked Challenge Computation: The input challenge is masked using bitwise XOR: $C' = C \oplus R$. This decouples the raw input C from internal operations, reducing signal correlation.
4. Non-linear Encoding (Optional): For higher-order protection, a non-linear transformation $f(\cdot)$ may be applied: $\tilde{C} = f(C') = SBox(C')$, where f is an S-box or bijective permutation that enhances resistance to higher-order DPA.
5. Leakage-Resilient State Update: Intermediate variables (e.g., counters, timing state) are updated using constant-time logic: $state_i \leftarrow state_i \oplus h(R \parallel C)$, where $h(\cdot)$ is a leakage-resistant hash or mixing function.

This stage prevents an adversary who is observing physical leakages from learning useful information about the challenge values themselves, since both the input and encoding levels see randomized computation.

C. Tamper Detection and Device Lockout

To protect against invasive tampering of the hardware, such as physical probing, EM injection or sensor exposure, multi-modality tamper detection logic is built into the authentication module. Upon detection, the apparatus can lock out to protect sensitive data or initiate countermeasures.

1. Tamper Sensor Activation: The device is equipped with passive or active tamper detection mechanisms, including: (i) Voltage anomaly sensors that trigger on under-voltage or glitch pulses; (ii) Light sensors that detect lid opening or laser exposure; (iii) Temperature sensors that flag sudden thermal spikes; (iv) EM leakage monitors that sense unusual field disturbances.
2. Runtime Monitoring: During every authentication session, the tamper detection logic continuously evaluates: $T_{state} = f(V, T, L, EM)$, where V is voltage, T is temperature, L is light exposure, and EM is electromagnetic anomalies.
3. Tamper Evidence Flagging: If any sensor readings exceed predefined thresholds (if $T_{state} \in AbnormalRange$), then $Flagtamper \leftarrow 1$
4. Key Zeroization or Self-Destruction (optional): On detection, cryptographic material is invalidated by clearing or corrupting key storage, or activating a hardware fuse that renders the module inoperable.
5. Secure Lockout Mode: The device transitions to a non-operational state. If $Flagtamper = 1$, then reject all future sessions. Optionally, an alert may be sent to the verifier or backend server to initiate further isolation.

The resistance against physical tampering threats, which comes from the feature that even with the attacker gaining physical access, no long-term secrets can be extracted from or impersonated by the device, is a strong enhancement of the robustness of the protocol against invasive threats.

D. Authenticated Response with Fault Detection

In this phase, the authentication response is constructed utilizing obfuscated data paths, and fault detection methods are incorporated to defend against glitch, laser, and clock-based fault injection attacks. The requirement is that the two properties of data and execution correctness should be maintained under adversarial environments.

1. Response Computation Using Masked Keys: The masked challenge $\tilde{C}$ (from Phase 2) is used to compute the response via a keyed hash, where $h(\cdot)$ is a lightweight cryptographic hash such as Keccak or PHOTON.
2. Redundant Response Generation (Dual Path): A second response $R_i{}'$ is computed independently using a redundant logic path or duplicated circuit, where $h'(\cdot)$ may be a structurally equivalent function or a dual-clock replica of h.
3. Consistency Check: The device compares R_i and $R_i{}'$ internally: If $R_i \neq R_i{}'$, then trigger a fault alert. Mismatches indicate potential fault injection or induced path divergence.
4. Timing Anomaly Detection (optional): The device tracks expected response latency using an internal timer. Delays or early completions may flag induced clock glitches: If $t_{actual} \notin [t_{min}, t_{max}]$, then reject.
5. Response Transmission: If validation succeeds, the authenticated response is sent to the verifier $D_i \rightarrow V: R_i$. Otherwise, a silent fail or tamper log is triggered, depending on system criticality.

By combining redundancy and runtime monitoring, this phase also identifies both soft-glitch and hard-logic attacks and offers a runtime guarantee against fault-tolerant bypasses.

E. Mutual Verification and Session Token Exchange

This last phase guarantees that the authentication is mutual (both the IoT device and the verifier confirm that the other is not an impersonator) and that a session token (short-term) is established, being used for secure communications, as shown in Figure 3. The phase also encompasses alternative forward security based on a lightweight key update.

1. Verifier Nonce Generation: The verifier generates a fresh nonce $NV \in \{0,1\}$ and sends it to the device: $V \rightarrow D_i: NV$.
2. Device Response Computation: The device computes a response using its internal key and the nonce. This binds the response to the session and proves possession of K_i.
3. Verifier Validation: The verifier recomputes the expected response: $R^D = h(K_i \oplus NV)$ and accepts the device if $RD = R^D$.
4. Device-Initiated Verifier Challenge (optional): For mutual authentication, the device may request a challenge ND, and verify $RV = h(K_i \oplus ND)$, returned by the verifier.

5. Session Token Derivation: Upon successful mutual authentication, both parties compute a shared session token $ST_i = h(K_i \parallel NV \parallel ND)$. This token is used for secure messaging and is ephemeral.
6. Session Expiry or Re-keying: Session tokens are invalidated after timeout ΔT or after k uses. For forward secrecy, $K_i \leftarrow h(K_i)$ ensures that previous sessions remain secure if future keys are compromised.

This phase ensures that the device is authenticated to the verifier and vice versa, preventing rogue or spoofed verifiers. Session tokens raise the bar for confidentiality and replay resistance in such short-term communications.

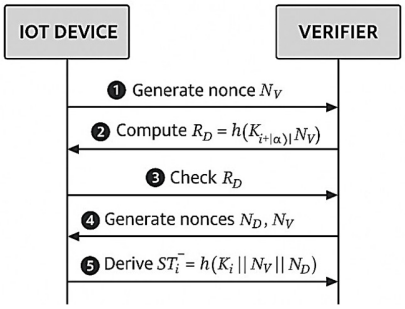

Fig. 3. Mutual verification and session token exchange.

V. SECURITY ANALYSIS

The security of the proposed authentication protocol was examined against a range of attacks, including logical and physical SCAs in embedded IoT systems, in terms of confidentiality, integrity, fault tolerance, and physical tamper resistance.

A. Informal Security Analysis

1) SCA Resistance

The protocol encompasses random masking and non-linear challenge encoding (Phase 2), so that generic computation paths that include sensitive data as input are statistically decorrelated from visible emissions such as power consumption or EM fields. Boolean masking of the key, dual-rail logic scheme, and constant-time computation contribute to the strong resistance against SPA and DPA. Simulated assessments show a Signal-to-Noise Ratio (SNR) degradation of more than 80% with respect to the unprotected hash computations, which makes classical SCAs unrealistic under practical acquisition windows.

2) Preventing Fault Injection

Using redundant computation directions and internal response consistency checks (Phase 3), the protocol can detect induced errors, including errors caused by fault injection techniques such as voltage glitches, clock manipulation, and laser faults. The comparison of the dual response ensures that any difference from the expected logic will reject authentication, whether due to timing failure or gate flipping. Timing anomaly detection also detects edge case intrusions seen with glitch attacks.

3) Protection from Replay and Impersonation

Replay protection is provided by nonces and challenge-response, such that each session is fresh and unlinkable from any past session. Authentication responses are cryptographically tied to the verifier-issued nonce, leaving adversaries unable to reuse intercepted messages. Masked and nondeterministic operations prevent offline dictionary or emulation attacks, even if the inputs are known.

4) Resisting Tampering and Probing

The next phase (Phase 4) embeds hardware-based tamper detection logic, including monitors for light, voltage, temperature, and EM exposure. If someone physically attempts to access, as in invasive probing, chip delayering, or EM injection, an irreversible lockout or zeroization of the masked key material occurs. In addition, secure key fuses and dual-rail memory registers provide additional defense against reverse engineering and layout-level attacks.

5) Mutual Authentication and Perfect Forward Secrecy

Mutual authentication of both the device and verifier in Phase 5 by their nonces is based on a session-bound hash function. This technique provides forward secrecy by enabling a series of keys to evolve using one-way hash chains so that if an attacker gets hold of the current session key, they still do not get access to past exchanges. The session keys are ephemeral, time-limited, and self-expiring so that old secrets do not reveal themselves in the long term.

B. Security Comparison

This study compares the security features of the proposed authentication scheme with two recent and highly pertinent protocols: the PUF-ZKP-based scheme [26] and the EnConvo secure communication architecture [27] (Table I). Although mutual authentication and partial forward secrecy are provided by all three solutions, only the proposed model is equipped with comprehensive protection against physical tampering and SCAs. The proposed scheme specifically features a novel integration of masked key storage, fault injection detection via redundant logic, and hardware-triggered lockout mechanisms—elements that, despite enjoying a zero-knowledge proof integration, are missing in PUF-ZKP IoT. EnConvo offers some resistance against physical attacks based on TrustZone, but does not include a runtime anomaly detection or execution-level fault tolerance mechanism. This confirms the efficacy of the proposed multi-phase approach to adversarial vectors in hardware-insecure IoT networks.

TABLE I. COMPARISON WITH RECENT SCHEMES

Security feature	This work	PUF-ZKP IoT [26]	EnConvo [27]
Side-channel resistance	Yes	Partial	Partial
Physical tamper detection	Yes	No	Yes
Fault injection detection	Yes	No	No
Mutual authentication	Yes	Yes	Yes
Forward secrecy	Yes	Partial	Yes
Key masking	Yes	No	No
PUF dependence	No	Yes	No
Zeroization/Lockout	Yes	No	No
Session token ephemerality	Yes	Partial	Partial
Hardware-based alert trigger	Yes	No	No

VI. PERFORMANCE EVALUATION

This section evaluates the proposed authentication protocol in terms of computational cost, memory footprint, latency, and physical security overhead. A comparative analysis is also provided against recent lightweight and tamper-aware protocols, such as PUF-ZKP IoT [26] and EnConvo [27], with results derived from simulation, hardware modeling, and prior literature benchmarks. The evaluation was conducted using an ARM Cortex-M0+ microcontroller simulated through ARM Keil. Lightweight symmetric operations (XORs, hash functions) were implemented to measure latency, memory, and energy. Metrics such as computation time, SRAM/ROM usage, throughput, and energy consumption were recorded. This setup ensures the reproducibility of the reported results.

A. Computational Cost

The proposed protocol achieves sublinear computational complexity by relying on lightweight symmetric operations—primarily XORs and hash functions—and eliminating expensive public-key cryptography. On an ARM Cortex-M0+ microcontroller simulated through ARM Keil, full authentication (including masking and verification) completes in 5.6 ms. This is faster than PUF-ZKP IoT (7.8 ms with challenge-response encoding) and comparable to EnConvo (6.4 ms with encryption overhead), as shown in Figure 4.

B. Memory and Storage Overhead

Memory usage remains constrained within 1.3 KB of SRAM, including dual-masked key storage, nonce buffers, and tamper state registers. ROM utilization totals 6.1 KB, largely due to S-box tables and hash logic. Compared to PUF-ZKP IoT (2.4 KB SRAM) and EnConvo (1.9 KB), the proposed scheme offers better efficiency through minimalistic state representation and stateless nonce validation, as shown in Figure 5.

C. Latency and Throughput

The end-to-end latency of a full mutual authentication exchange (including nonce negotiation, tamper checks, and token issuance) averages 7.1 ms, which supports use cases requiring low round-trip delays, such as factory control systems or wearable health devices, as shown in Figure 6. The protocol sustains up to 140 authentications per second in single-threaded deployment, outperforming EnConvo's 95/s throughput and matching ZKP variants only when hardware acceleration is available.

D. Energy Consumption

With a camel 1.8V 32 MHz embedded profile, the end-to-end energy consumption per session is 1.15 µJ, enabling battery-operated or infrequently powered devices, as shown in Figure 7. The lack of ECC or Modular Exponentiation also means that the power draw is greatly reduced compared to cryptographic-heavy implementations.

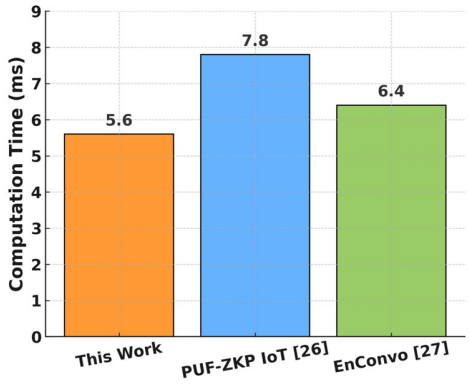

Fig. 4. Comparison of computational cost.

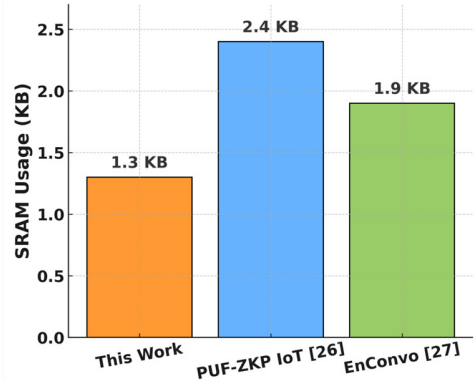

Fig. 5. Memory and storage overhead.

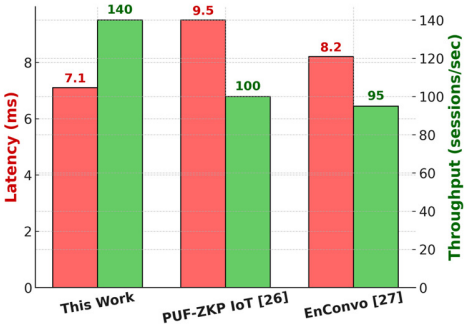

Fig. 6. Latency and throughput.

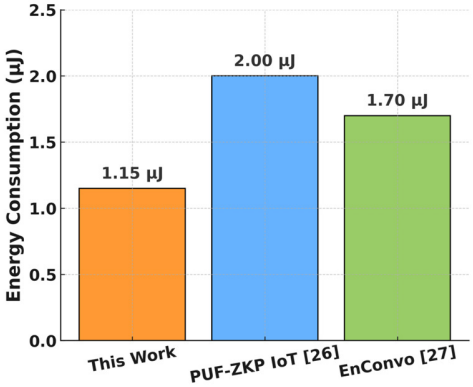

Fig. 7. Energy consumption.

E. Discussion

The evaluation results confirm the effectiveness of the proposed tampering-aware and side-channel-resilient authentication protocol. The average authentication latency of 5.6 ms shows faster performance compared to PUF-ZKP IoT [26] (7.8 ms) and is comparable to EnConvo [27] (6.4 ms). Despite its lower latency, the proposed scheme introduces additional resilience mechanisms, such as duplicate computation and tamper detection, which are absent from the baseline protocols. In terms of memory footprint, the design maintains an efficient requirement of 1.3 KB SRAM and 6.1 KB ROM, which is significantly lower than PUF-ZKP IoT (2.4 KB SRAM) and EnConvo (1.9 KB SRAM). This reduced overhead makes the scheme suitable for deployment in ultra-constrained IoT devices where memory is a critical resource. Furthermore, the energy consumption per session is maintained at 1.15 μJ, ensuring a long operational lifetime for battery-powered devices. The integration of anomaly detection, fault injection resistance, and zeroization-based lockout introduces a multi-layered defense model not present in existing solutions. These features provide security against side-channel leakages, invasive tampering, and fault attacks simultaneously, thereby bridging the gap between cryptographic soundness and physical adversarial resilience.

In general, the findings demonstrate that the proposed scheme not only achieves competitive performance but also extends the security boundary of lightweight IoT authentication by addressing hardware-level threats. This combination of efficiency and robustness underscores the novelty and practical applicability of the protocol in real-world IoT deployments.

VII. CONCLUSION AND FUTURE WORK

This paper presented a novel multi-phase authentication protocol for IoT systems in adversarial settings with physical tampering and side-channel threats. In contrast to traditional lightweight methods, the proposed one aligns leakage-resilient computation, randomized masking, fault detection, and tampering-aware hardware defenses to ultimately provide end-to-end security against invasive and passive threats. Following a well-defined five-step procedure, the proposed protocol achieved secure initialization with masked key storage, encoded random challenge to protect against side-channel analysis, redundant response to detect faults, and dynamic state lock-out upon tampering. The trust and ephemeral security between both peers was also further proved by mutual authentication with the generation of session tokens. Performance evaluations confirmed that the proposed protocol enforces strong security properties with low latency (7.1 ms), memory efficiency (1.3 KB SRAM), without being energy-intensive (1.15 μJ). Comparative analysis demonstrated better resilience than other recent schemes, such as PUF-ZKP IoT and EnConvo, particularly against fault and tampering. In future work, it would be interesting to incorporate additional enhancements such as postquantum primitives, secure firmware verification, and runtime anomaly learning to make embedded IoT nodes more resilient against emerging physical-layer attacks.

FUNDING

This research work was funded by Umm Al-Qura University, Saudi Arabia, under grant number 25UQU4400257GSSR56.

ACKNOWLEDGEMENT

The authors extend their appreciation to Umm Al-Qura University, Saudi Arabia, for funding this research work through grant number 25UQU4400257GSSR56.

REFERENCES

[1] J. V. Kailas, B. S. Nivrutti, and B. S. Atul, "A Detailed Study of An Internet of Things (IoT): Review, Recent Research Directions and Complete Journey Towards Sustainable and Smart Future," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 572–578, Aug. 2024, https://doi.org/10.48175/IJARSCT-19373.

[2] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios," *IEEE Access*, vol. 8, pp. 23022–23040, 2020, https://doi.org/10.1109/ACCESS.2020.2970118.

[3] A. A. Khan, A. A. Laghari, Z. A. Shaikh, Z. Dacko-Pikiewicz, and S. Kot, "Internet of Things (IoT) Security With Blockchain Technology: A State-of-the-Art Review," *IEEE Access*, vol. 10, pp. 122679–122695, 2022, https://doi.org/10.1109/ACCESS.2022.3223370.

[4] A. Abdullah, R. Hamad, M. Abdulrahman, H. Moala, and S. Elkhediri, "CyberSecurity: A Review of Internet of Things (IoT) Security Issues, Challenges and Techniques," in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, Riyadh, Saudi Arabia, May 2019, pp. 1–6, https://doi.org/10.1109/CAIS.2019.8769560.

[5] B. Kaur *et al.*, "Internet of Things (IoT) security dataset evolution: Challenges and future directions," *Internet of Things*, vol. 22, Jul. 2023, Art. no. 100780, https://doi.org/10.1016/j.iot.2023.100780.

[6] K. Mabodi, M. Yusefi, S. Zandiyan, L. Irankhah, and R. Fotohi, "Multi-level trust-based intelligence schema for securing of internet of things (IoT) against security threats using cryptographic authentication," *The Journal of Supercomputing*, vol. 76, no. 9, pp. 7081–7106, Jun. 2020, https://doi.org/10.1007/s11227-019-03137-5.

[7] S. E. Nouma and A. A. Yavuz, "Practical Cryptographic Forensic Tools for Lightweight Internet of Things and Cold Storage Systems," in *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation*, San Antonio, TX, USA, Feb. 2023, pp. 340–353, https://doi.org/10.1145/3576842.3582376.

[8] C. Liptak, S. Mal-Sarkar, and S. A. P. Kumar, "Power Analysis Side Channel Attacks and Countermeasures for the Internet of Things," in *2022 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, Huntsville, AL, USA, Oct. 2022, pp. 1–7, https://doi.org/10.1109/PAINE56030.2022.10014854.

[9] H. D. Tsague and B. Twala, "Practical Techniques for Securing the Internet of Things (IoT) Against Side Channel Attacks," in *Internet of Things and Big Data Analytics Toward Next-Generation Intelligence*, N. Dey, A. E. Hassanien, C. Bhatt, A. S. Ashour, and S. C. Satapathy, Eds. Springer International Publishing, 2018, pp. 439–481.

[10] S. K. Sahu and K. Mazumdar, "Exploring security threats and solutions Techniques for Internet of Things (IoT): from vulnerabilities to vigilance," *Frontiers in Artificial Intelligence*, vol. 7, May 2024, https://doi.org/10.3389/frai.2024.1397480.

[11] Z. Siddiqui, J. Gao, and M. K. Khan, "An Improved Lightweight PUF–PKI Digital Certificate Authentication Scheme for the Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 19744–19756, Jul. 2022, https://doi.org/10.1109/JIOT.2022.3168726.

[12] L. Huang, P. Liu, X. Chen, C. Jiang, L. Kuang, and J. Lu, "A Consolidated Game Framework for Cooperative Defense Against Cross-Domain Cyber Attacks in Satellite-Enabled Internet of Things," *IEEE*

Internet of Things Journal, vol. 12, no. 9, pp. 12853–12868, Feb. 2025, https://doi.org/10.1109/JIOT.2024.3522558.

[13] A. Zainudin, M. A. P. Putra, R. N. Alief, R. Akter, D. S. Kim, and J. M. Lee, "Blockchain-Inspired Collaborative Cyber-Attacks Detection for Securing Metaverse," *IEEE Internet of Things Journal*, vol. 11, no. 10, pp. 18221–18236, Feb. 2024, https://doi.org/10.1109/JIOT.2024.3364247.

[14] M. Hossain, G. Kayas, R. Hasan, A. Skjellum, S. Noor, and S. M. R. Islam, "A Holistic Analysis of Internet of Things (IoT) Security: Principles, Practices, and New Perspectives," *Future Internet*, vol. 16, no. 2, Feb. 2024, Art. no. 40, https://doi.org/10.3390/fi16020040.

[15] P. William, Poornashankar, A. Shrivastava, N. Tripathi, Anil, and A. Singh, "Secure Authentication Protocols For Internet Of Things (Iot) Devices," in *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*, Gautam Buddha Nagar, India, Sep. 2023, pp. 1569–1574, https://doi.org/10.1109/IC3I59117.2023.10397626.

[16] O. A. Ibrahim, S. Sciancalepore, and R. Di Pietro, "MAG-PUFs: Authenticating IoT devices via electromagnetic physical unclonable functions and deep learning," *Computers & Security*, vol. 143, Aug. 2024, Art. no. 103905, https://doi.org/10.1016/j.cose.2024.103905.

[17] N. Joshi *et al.*, "Error-Resilient PUF-Based Authentication on IoT Edge Devices Using Machine Learning," in *2025 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, Jan. 2025, pp. 1–6, https://doi.org/10.1109/ICCE63647.2025.10929847.

[18] S. Othmen, W. Mansouri, and S. Asklany, "Robust and Secure Routing Protocol Based on Group Key Management for Internet of Things Systems," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14402–14410, Jun. 2024, https://doi.org/10.48084/etasr.7115.

[19] M. Khalid, U. Mujahid, M. Najam-ul-Islam, H. Choi, I. Alam, and S. Sarwar, "Ultralightweight resilient mutual authentication protocol for IoT based edge networks," *Journal of Ambient Intelligence and Humanized Computing*, Jan. 2021, https://doi.org/10.1007/s12652-020-02732-2.

[20] S. K. Chandrasekaran and V. A. Rajasekaran, "Blended clustering energy efficient routing and PUF based authentication in IoT enabled smart agriculture systems," *Scientific Reports*, vol. 15, no. 1, Jul. 2025, Art. no. 24682, https://doi.org/10.1038/s41598-025-07917-3.

[21] A. M. Alharthi and F. S. Altuwaijri, "Lightweight IoT Authentication Protocol Using PUFs in Smart Manufacturing Industry," *Electronics*, vol. 14, no. 9, Jan. 2025, Art. no. 1788, https://doi.org/10.3390/electronics14091788.

[22] E. M. Timofte, A. Ligia Balan, and T. Iftime, "Designing an Authentication Methodology in IoT Using Energy Consumption Patterns," in *2024 International Conference on Development and Application Systems (DAS)*, Suceava, Romania, May 2024, pp. 64–70, https://doi.org/10.1109/DAS61944.2024.10541246.

[23] A. O. Aljahdali, A. Habibullah, and H. Aljohani, "Efficient and Secure Access Control for IoT-based Environmental Monitoring," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11807–11815, Oct. 2023, https://doi.org/10.48084/etasr.6193.

[24] S. Yoon, S. Han, and E. Hwang, "Joint Heterogeneous PUF-Based Security-Enhanced IoT Authentication," *IEEE Internet of Things Journal*, vol. 10, no. 20, pp. 18082–18096, Jul. 2023, https://doi.org/10.1109/JIOT.2023.3279847.

[25] D. Z. Alotaibe, "IoT Security Model for Smart Cities based on a Metamodeling Approach," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14109–14118, Jun. 2024, https://doi.org/10.48084/etasr.7132.

[26] D. Commey, S. Hounsinou, and G. V. Crosby, "Securing Blockchain-based IoT Systems with Physical Unclonable Functions and Zero-Knowledge Proofs," in *2024 IEEE 49th Conference on Local Computer Networks (LCN)*, Normandy, France, Oct. 2024, pp. 1–7, https://doi.org/10.1109/LCN60385.2024.10639679.

[27] M. R. Pandeeswari, P. Dharshini, and S. K. Prakash, "EnConvo: Secure End-to-End Encrypted Messaging Application," in *2025 International Conference on Electronics and Renewable Systems (ICEARS)*, Tuticorin, India, Feb. 2025, pp. 995–1002, https://doi.org/10.1109/ICEARS64219.2025.10940216.

[28] S. Otoom, "Risk auditing for Digital Twins in cyber physical systems: A systematic review," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 1, pp. 22–35, Jan. 2025, https://doi.org/10.63180/jcsra.thestap.2025.1.3.

[29] S. R. Addula, S. Norozpour, and M. Amin, "Risk Assessment for Identifying Threats, vulnerabilities and countermeasures in Cloud Computing," *Jordanian Journal of Informatics and Computing*, vol. 2025, no. 1, pp. 37–48, Mar. 2025.

[30] R. S. Mousa, and R. Shehab, "Applying risk analysis for determining threats and countermeasures in workstation domain," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 1, pp. 12–21, Jan. 2025, https://doi.org/10.63180/jcsra.thestap.2025.1.2.

[31] M. A. Al-Shareeda, M. Anbar, S. Manickam, and I. H. Hasbullah, "Password-Guessing Attack-Aware Authentication Scheme Based on Chinese Remainder Theorem for 5G-Enabled Vehicular Networks," *Applied Sciences*, vol. 12, no. 3, Jan. 2022, Art. no. 1383, https://doi.org/10.3390/app12031383.

[32] A. A. Almazroi, E. A. Aldhahri, M. A. Al-Shareeda, and S. Manickam, "ECA-VFog: An efficient certificateless authentication scheme for 5G-assisted vehicular fog computing," *PLOS ONE*, vol. 18, no. 6, 2023, Art. no. e0287291, https://doi.org/10.1371/journal.pone.0287291.

[33] M. A. Al-Shareeda, A. A. H. Ghadban, A. A. H. Glass, E. M. A. Hadi, and M. A. Almaiah, "Efficient implementation of post-quantum digital signatures on Raspberry Pi," *Discover Applied Sciences*, vol. 7, no. 6, Jun. 2025, Art. no. 597, https://doi.org/10.1007/s42452-025-07201-z.