Journal of
Sensor and Actuator Networks

MDPI

Article

A Lightweight, Forward-Secure, and Cloning-Resistant Authenticated Encryption Scheme for Data Transmission in Heterogeneous IoT Networks

Fares Mezrag [1,*], Anwar Noureddine Bahache [1], Abderrahim Zemmit [2] and Pascal Lorenz [3,*]

[1] Preparatory Cycle Department, National Higher School of Mathematics, Sidi Abdellah, Algiers 16093, Algeria; anwar.bahache@nhsm.edu.dz
[2] Electrical Engineering Department, University of M'Sila, M'Sila 28000, Algeria; abderrahim.zemmit@univ-msila.dz
[3] Network and Telecommunication Research Group, University of Haute Alsace IUT, 34 Rue du Grillenbreit, 68008 Colmar, France
* Correspondence: fares.mezrag@nhsm.edu.dz (F.M.); pascal.lorenz@uha.fr (P.L.)

Abstract

The rapid growth of the Internet of Things (IoT) in safety–critical applications requires lightweight and robust security mechanisms for resource–constrained devices. Existing authentication and key agreement schemes, including that of Al-Hejri et al., suffer from critical weaknesses, such as state desynchronization, lack of sender-forward secrecy, non–atomic authenticated encryption, replay and device cloning vulnerabilities, and insufficient formal verification. This paper introduces FRACE–IoT, a lightweight, hardware–bound forward–secure, and cloning–resistant authenticated encryption scheme for heterogeneous four–tier IoT networks. The proposed approach combines Curve25519 cryptography, physical unclonable functions, authenticated ChaCha20–Poly1305 encryption, and HKDF–BLAKE3 key derivation within a single–pass key establishment architecture that achieves sender-forward secrecy and hardware-enforced protection against credential extraction. Security analysis and formal verification using the BAN logic and the AVISPA tool under the Dolev–Yao model confirm session key security, mutual belief, and robust resistance to major cyber-attacks. Experimental evaluation on real embedded hardware IoT boards, namely, FIT IoT–LAB M3 and Nordic nRF52840DK, demonstrates that FRACE–IoT achieves a reduced computation cost of 187.92 ms, communication overhead of 1344 bits, and a total energy consumption of 210.10 mJ, making it suitable for efficient and scalable IoT security deployment.

Keywords: Internet of Things; lightweight cryptography; physical unclonable function; elliptic curve cryptography; ChaCha20–Poly1305; sender forward secrecy; formal security verification; authenticated key agreement

Check for updates

Academic Editor: Guangjie Han

Received: 5 September 2026
Revised: 23 October 2026
Accepted: 23 October 2026
Published: 2 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The Internet of Things (IoT) has emerged as a foundational technology supporting a wide range of modern applications, including smart homes, industrial automation, healthcare, and transportation, by facilitating seamless connectivity and data exchange among billions of devices. Nevertheless, this extensive integration presents substantial security challenges owing to the inherent characteristics of IoT ecosystems. These challenges arise from the vast heterogeneity of devices, their limited computational and energy resources,

and their deployment in potentially hostile environments. Ensuring the confidentiality, integrity, and authenticity of data transmitted within these networks is crucial, as any compromise can result in severe consequences, including privacy breaches, operational disruptions, and physical harm. Traditional cryptographic solutions, often designed for more powerful computing devices, are frequently too resource-intensive for constrained IoT devices, leading to a critical trade-off between security and operational efficiency of the devices. Although numerous lightweight authentication and key agreement schemes have been proposed to address these challenges, many still exhibit vulnerabilities to advanced attacks, including impersonation, replay attacks, and insider threats. Furthermore, the increasing sophistication of adversaries, particularly those capable of physically capturing devices, highlights the urgent need for security mechanisms that can resist hardware-based cloning and guarantee forward secrecy, ensuring that past session keys remain secure even if long-term secrets are compromised. The development of robust yet resource-efficient security protocols capable of mitigating these multifaceted threats in heterogeneous IoT environments remains a significant research priority.

1.1. Problem Statement

Achieving strong security guarantees while maintaining the lightweight characteristics required by resource–constrained IoT devices remains a fundamental challenge. To address this issue, several authentication and key agreement schemes have been proposed in the literature. Among them, Al–Hejri et al. [1] introduced a lightweight, secure, and scalable data transmission scheme for heterogeneous IoT networks that combines symmetric–key cryptography with ECC–based key agreement to enhance confidentiality, integrity, and scalability. Although the authors claimed that their scheme provides both security and efficiency, our analysis identifies several critical security weaknesses. Specifically, the scheme suffers from vulnerability to state desynchronization attacks, lacks sender-forward secrecy, relies on non–atomic authenticated encryption mechanisms, and remains susceptible to replay attacks during data transmission. Furthermore, the scheme does not provide adequate protection against hardware–based cloning and impersonation attacks. These limitations collectively motivate the central research problem addressed in this paper: the design of a lightweight, formally verified authenticated encryption scheme for heterogeneous IoT networks that achieves hardware–bound device authentication, sender-forward secrecy, atomic authenticated encryption, and resistance to the principal classes of network–level attacks, while maintaining computational and communication costs suitable for deployment on severely resource–constrained devices operating across a multi–tier network architecture.

1.2. Contributions

The main contributions of this paper are summarized as follows:

1. A review of the scheme proposed by Al–Hejri et al. [1]: identifies six specific security weaknesses. These include vulnerability to state desynchronization, lack of sender-forward secrecy, non-atomic authenticated encryption, susceptibility to replay attacks during data transmission, vulnerability to device cloning and impersonation, and the absence of formal security verification.
2. Novel Scheme Design: We propose FRACE-IoT, a novel, lightweight, authenticated encryption scheme designed to ensure forward security and resistance to cloning, specifically for data transmission within heterogeneous IoT networks. The scheme simultaneously provides confidentiality, ciphertext integrity, mutual authentication, sender-forward secrecy, and robust resistance to replay, man-in-the-middle, state desynchronization, and hardware-level device cloning attacks.

3. Co-Designed Lightweight Architectural Synthesis: Rather than relying on conventional cryptographic handshakes, FRACE-IoT adapts the single-pass, zero-round-trip-time (0-RTT) paradigm, also employed in protocols such as TLS 1.3, to a unified, resource-efficient IoT architecture through three complementary design innovations:
   (a) Tailored 0-RTT Integration with Resistance to State Desynchronization: By structuring key establishment as a unidirectional, single-pass exchange authenticated using hardware-bound PUF tokens and cached nonces, FRACE-IoT eliminates interactive multi-round handshakes, reduces radio-listening energy consumption, and mitigates the risk of persistent state desynchronization associated with rolling-key schemes, such as that of Al-Hejri et al.
   (b) Hybrid Ephemeral–Static Offline Precomputation: By decomposing Diffie–Hellman key agreement into an offline-provisioned static component Z^{static} and a single online ephemeral scalar multiplication DH^{ES}, FRACE-IoT reduces the online elliptic-curve scalar-multiplication workload on constrained sensor nodes by 50%, while preserving sender-forward secrecy under the stated key-erasure and compromise assumptions.
   (c) Atomic Single-Pass AEAD Integration: By adopting ChaCha20-Poly1305 authenticated encryption with associated data (AEAD) instead of a conventional two-pass Encrypt-then-MAC construction, FRACE-IoT streamlines cryptographic processing, reduces intermediate buffering overhead, and avoids decryption-oracle vulnerabilities arising from unsafe decryption and authentication ordering.
4. Hardware-Bound Authentication: We integrate SRAM-based PUFs to bind each device's cryptographic identity to its unique silicon instance. This ensures that firmware extraction alone cannot enable device cloning or impersonation, as PUF responses cannot be reproduced on different physical hardware.
5. Lightweight Cryptographic Integration: FRACE-IoT integrates Curve25519 elliptic curve cryptography for key agreement with the ChaCha20-Poly1305 authenticated encryption algorithm, known for its high performance. Additionally, FRACE-IoT employs HKDF and BLAKE3 to provide robust security while maintaining minimal computational overhead, making it suitable for IoT devices with limited resources.
6. Formal Verification and Experimental Validation: We conducted formal security verification using both BAN logic and the automated AVISPA tool under the Dolev-Yao adversary model. We explicitly note that the formal AVISPA modeling and verification are strictly limited to the key establishment stage (Algorithms 1 and 2), confirming session key confidentiality, entity authentication, and resilience against replay and man-in-the-middle attacks, whereas the data transmission stage (Algorithm 3) is evaluated via computational and informal security analyses. We further evaluated the scheme on real IoT-embedded hardware platforms, specifically the FIT IoT-LAB M3 (ARM Cortex-M3) and Nordic nRF52840DK (ARM Cortex-M4). The evaluation revealed an efficient computation cost of 187.92 ms, an optimized communication overhead of 1344 bits, and significantly reduced energy consumption compared with recent similar schemes.
7. Comparative Analysis: We present a comparative analysis that demonstrates the superior security features and competitive performance of FRACE-IoT compared to various state-of-the-art lightweight authentication and encryption schemes for IoT.

Algorithm 1 Key Establishment: CIoT ↔ CD

Require: ECC params (E, G, q); PUF challenge C_{IoT}; $Z^{\text{static}}_{\text{IoT,CD}}$ (both parties) and $K^{\text{SPu}}_{\text{CD}}$ enrollment DB.
Ensure: $K^{\text{sh}}_{\text{IoT,CD}}$

CIoT Device(Initiator):

$K^{\text{EPr}}_{\text{IoT}} \xleftarrow{R} \mathbb{Z}_{2^{255}-19}$; $K^{\text{EPu}}_{\text{IoT}} \leftarrow K^{\text{EPr}}_{\text{IoT}} \cdot G$;
$R_{\text{IoT}} \leftarrow \text{PUF}(C_{\text{IoT}})$; $n_{\text{IoT}} \xleftarrow{R} \{0,1\}^{64}$ ▷ *Reproduce device PUF response*
$\tau_{\text{IoT}} = \text{BLAKE3}_{64}(R_{\text{IoT}} \| n_{\text{IoT}})$
$\text{DH}^{\text{ES}}_{\text{IoT}} = K^{\text{EPr}}_{\text{IoT}} \cdot K^{\text{SPu}}_{\text{CD}}$
$K^{\text{sh}}_{\text{IoT,CD}} = \text{HKDF-BLAKE3}(\text{DH}^{\text{ES}}_{\text{IoT}} \| Z^{\text{static}}_{\text{IoT,CD}}, \text{``FRACE-IoT-CD''}, n_{\text{IoT}})$
send $\mathcal{M}_1 = \{ID_{\text{IoT}}, K^{\text{EPu}}_{\text{IoT}}, \tau_{\text{IoT}}, n_{\text{IoT}}\} \rightarrow$ CD
Delete: $K^{\text{EPr}}_{\text{IoT}} \leftarrow \perp$; $\text{DH}^{\text{ES}}_{\text{IoT}} \leftarrow \perp$

Controller (Verifier, no reply sent):

$K^{\text{SPr}}_{\text{CD}} \leftarrow \text{HKDF-BLAKE3}(R_{\text{CD}}, \text{``static-key''}, ID_{\text{CD}})$
$(C_{\text{IoT}}, R_{\text{IoT}}) \leftarrow \text{DB.lookup}(ID_{\text{IoT}})$ ▷ *not found ⇒ reject, alert GW*
assert $n_{IoT} \notin W_{nonce}(ID_{IoT})$ ▷ *Anti-replay check within active W_{nonce}; fail ⇒ reject replayed M_1*
$W_{nonce}(ID_{IoT}).insert(n_{IoT})$
assert $\tau_{\text{IoT}} \stackrel{?}{=} \text{BLAKE3}_{64}(R_{\text{IoT}} \| n_{\text{IoT}})$ ▷ *fail ⇒ reject, alert GW*
$\text{DH}^{\text{ES}}_{\text{CD}} = K^{\text{SPr}}_{\text{CD}} \cdot K^{\text{EPu}}_{\text{IoT}}$
$K^{\text{sh}}_{\text{IoT,CD}} = \text{HKDF-BLAKE3}(\text{DH}^{\text{ES}}_{\text{CD}} \| Z^{\text{static}}_{\text{IoT,CD}}, \text{``FRACE-IoT-CD''}, n_{\text{IoT}})$ ▷ *no reply sent*
Delete: $K^{\text{SPr}}_{\text{CD}} \leftarrow \perp$; $\text{DH}^{\text{ES}}_{\text{CD}} \leftarrow \perp$

Algorithm 2 Key Establishment: CD ↔ GW

Require: ECC params (E, G, q); PUF challenge C_{CD}; enrollment DB; $K^{\text{SPu}}_{\text{GW}}$,$Z^{\text{static}}_{\text{CD,GW}}$ (both parties).
Ensure: $K^{\text{sh}}_{\text{CD,GW}}$

Controller (Initiator):

$K^{\text{EPr}}_{\text{CD}} \xleftarrow{R} \mathbb{Z}_{2^{255}-19}$; $K^{\text{EPu}}_{\text{CD}} \leftarrow K^{\text{EPr}}_{\text{CD}} \cdot G$;
$R_{\text{CD}} \leftarrow \text{PUF}(C_{\text{CD}})$; $n_{\text{CD}} \xleftarrow{R} \{0,1\}^{64}$ ▷ *Reproduce device PUF response*
$\tau_{\text{CD}} = \text{BLAKE3}_{64}(R_{\text{CD}} \| n_{\text{CD}})$
$\text{DH}^{\text{ES}}_{\text{CD}} = K^{\text{EPr}}_{\text{CD}} \cdot K^{\text{SPu}}_{\text{GW}}$
$K^{\text{sh}}_{\text{CD,GW}} = \text{HKDF-BLAKE3}(\text{DH}^{\text{ES}}_{\text{CD}} \| Z^{\text{static}}_{\text{CD,GW}}, \text{``FRACE-CD-GW''}, n_{\text{CD}})$
send $\mathcal{M}_2 = \{ID_{\text{CD}}, K^{\text{EPu}}_{\text{CD}}, \tau_{\text{CD}}, n_{\text{CD}}\} \rightarrow$ GW
delete: $K^{\text{EPr}}_{\text{CD}} \leftarrow \perp$; $\text{DH}^{\text{ES}}_{\text{CD}} \leftarrow \perp$

Gateway (Verifier, no reply sent):

$(C_{\text{CD}}, R_{\text{CD}}) \leftarrow \text{DB.lookup}(ID_{\text{CD}})$ ▷ *not found ⇒ reject, log alert*
assert $n_{CD} \notin W_{nonce}(ID_{CD})$ ▷ *Anti-replay check within active W_{nonce}; fail ⇒ reject replayed M_2*
$W_{nonce}(ID_{CD}).insert(n_{CD})$
assert $\tau_{\text{CD}} \stackrel{?}{=} \text{BLAKE3}_{64}(R_{\text{CD}} \| n_{\text{CD}})$ ▷ *fail ⇒ reject, log alert*
$\text{DH}^{\text{ES}}_{\text{GW}} = K^{\text{SPr}}_{\text{GW}} \cdot K^{\text{EPu}}_{\text{CD}}$
$K^{\text{sh}}_{\text{CD,GW}} = \text{HKDF-BLAKE3}(\text{DH}^{\text{ES}}_{\text{GW}} \| Z^{\text{static}}_{\text{CD,GW}}, \text{``FRACE-CD-GW''}, n_{\text{CD}})$ ▷ *no reply sent*
Delete: $\text{DH}^{\text{ES}}_{\text{GW}} \leftarrow \perp$

Algorithm 3 Data Transmission Phase with Ephemeral Sequence Lifecycle

Require: Data m; session keys $K^{sh}_{IoT,CD}$, $K^{sh}_{CD,GW}$
Ensure: m received and verified at GW

// State Initialization (Maintained in Volatile SRAM)
Upon session key derivation: initialize $seq_{tx} \leftarrow 0$, $seq_{rx} \leftarrow 0$ in volatile RAM.
// Power Failure and Reboot Strategy
On Power Loss / System Reboot:
a) Volatile RAM is cleared: $K^{sh} \leftarrow \perp$, $seq_{tx} \leftarrow \perp$, $seq_{rx} \leftarrow \perp$.
b) Node re-executes Algorithm 1 (or Algorithm 2) to negotiate fresh session key $K^{sh\prime}$.
c) Re-initialize sequence counters in RAM: $seq_{tx} \leftarrow 0$, $seq_{rx} \leftarrow 0$.

CIoT Device (Sender):
1: $seq_{tx} \leftarrow seq_{tx} + 1$
2: $AD \leftarrow \{session_id, seq_{tx}, ID_{IoT}\}$; $n_{tx} \xleftarrow{R} \{0,1\}^{96}$
3: $(C_1, T_1) \leftarrow \text{AEAD}(K^{sh}_{IoT,CD}, n_{tx}, m, AD)$;
4: **send** $\mathcal{M}_3 = \{C_1, T_1, n_{tx}, seq_{tx}\} \rightarrow$ CD

Controller (Receiver, Verifier, and Aggregator):
5: **if** $seq_{tx} \leq seq_{rx}$ **then** ▷ *Strict anti-replay verification*
6: **reject** and alert GW ▷ *Replayed or stale transmission*
7: **end if**
8: **if** $T_1 \neq \text{Poly1305}(K^{sh}_{IoT,CD}, C_1 \parallel AD)$ **then** ▷ *INT-CTXT tag check*
9: **reject** and alert GW ▷ *Integrity forgery or context mismatch*
10: **end if**
11: $m \leftarrow \text{ChaCha20-Decrypt}(K^{sh}_{IoT,CD}, n_{tx}, C_1)$
12: $seq_{rx} \leftarrow seq_{tx}$ ▷ *Update expected sequence threshold in RAM*
13: $m_{agg} \leftarrow \text{Aggregate}(\{m\})$ ▷ *Compile validated CIoT cluster data*
14: $seq'_{tx} \leftarrow seq'_{tx} + 1$ ▷ *CD-to-GW sequence counter in RAM*
15: $AD' \leftarrow \{session_id', seq'_{tx}, ID_{CD}\}$
16: $n'_{tx} \leftarrow_R \{0,1\}^{96}$
17: $(C_2, T_2) \leftarrow \text{AEAD}(K^{sh}_{CD,GW}, n'_{tx}, m_{agg}, AD')$;
18: **send** $M_4 = \{C_2, T_2, n'_{tx}, seq'_{tx}\} \longrightarrow$ GW

Gateway (Final Verifier and Secure Storage):
19: **if** $seq'_{tx} \leq seq'_{rx}$ **or** $T_2 \neq \text{Poly1305}(K^{sh}_{CD,GW}, C_2 \parallel AD')$ **then**
20: **reject** and log alert ▷ *Replay or authenticity verification failed*
21: **end if**
22: $m_{agg} \leftarrow \text{ChaCha20-Decrypt}(K^{sh}_{CD,GW}, n'_{tx}, C_2)$
23: $seq'_{rx} \leftarrow seq'_{tx}$
24: Securely store m_{agg} into persistent database

1.3. Organization of the Paper

The remainder of this paper is organized as follows. Section 2 surveys related work in lightweight IoT authentication and secure data transmission. Section 3 reviews the preliminaries underlying the proposed scheme. Section 4 presents the network and threat models. Section 5 analyzes the vulnerabilities of a recent Al-Hejri et al.'s Scheme. Section 6 details the five phases of the proposed FRACE-IoT scheme. Section 7 provides comprehensive formal and informal security analyses, including automated verification via AVISPA under the Dolev–Yao threat model, mathematical verification of mutual belief and session key agreement using BAN logic, and thorough informal evaluations. Section 8 evaluates computational cost, communication overhead, and energy consumption through experimental measurements on embedded hardware. Finally, Section 9 concludes the paper.

2. Related Works

Recent research on IoT security has primarily focused on lightweight mutual authentication, key agreement, secure data transmission, privacy preservation, and resource-aware

communication. These aspects are particularly important in IoT environments, where devices often operate under strict constraints in terms of computational power, memory, energy consumption, and communication bandwidth. Consequently, security protocols must provide effective protection against impersonation, replay, insider, stolen-verifier, and desynchronization attacks without introducing excessive overhead. Recent studies have also expanded the scope of IoT security beyond conventional authentication mechanisms to include privacy-preserving data sharing, secure routing, edge-assisted protection, post-quantum authentication, and intelligent anomaly detection. The following review examines these developments by considering the main objectives, underlying mechanisms, and security contributions of the proposed approaches.

The earliest work in this set was conducted by Ju and Park [2], who proposed a provably secure lightweight mutual authentication and key agreement scheme for cloud-based IoT environments. Their approach addresses the challenge of establishing trusted communication between IoT devices and cloud-based services while limiting the computational burden imposed on resource-constrained nodes. Mutual authentication allows communicating entities to verify each other's legitimacy before exchanging sensitive information, whereas key agreement establishes a shared secret for protecting subsequent communications. The emphasis on provable security is particularly relevant because a protocol's use of lightweight cryptographic operations does not, by itself, guarantee resistance to sophisticated attacks. Similarly, Peivandizadeh et al. [3] introduced a secure authentication and key exchange scheme based on elliptic curve cryptography (ECC) and the elliptic curve Diffie–Hellman (ECDH) mechanism. ECC provides public-key operations with relatively compact key sizes, while ECDH enables communicating parties to derive a common secret over an insecure channel. This combination is intended to reduce computational and communication costs while preserving the security properties required for IoT communication. Nevertheless, the practical suitability of ECC-based schemes also depends on implementation efficiency, device capabilities, and the security assumptions associated with their underlying mathematical problems.

In 2024, several studies extended secure communication beyond conventional authentication and key agreement. Gangadharaiah and Bhajantri [4] proposed a secure data dissemination and routing framework for IoT using proxy re-encryption. This approach addresses an important challenge in distributed IoT environments: securely sharing encrypted information without requiring the original data owner to decrypt and re-encrypt the data for every recipient. Proxy re-encryption enables a designated intermediary to transform ciphertexts from one recipient's encryption context to another under controlled conditions, thereby supporting data sharing while reducing the cryptographic burden on constrained devices. Its security consequently depends on the re-encryption construction and the protection of the associated keys and authorization mechanisms. Al-Hejri et al. [1] presented a lightweight, secure, and scalable data-transmission scheme combining symmetric-key cryptography with ECC-based key agreement. The general design separates the establishment of cryptographic key material from the protection of transmitted data, allowing efficient symmetric encryption to be used for subsequent communication. Such a combination can reduce the cost of repeatedly applying public-key operations to large volumes of IoT data, while ECC facilitates secure key establishment. The approach is particularly relevant to heterogeneous IoT networks in which devices differ in computational capacity and communication requirements.

Attkan et al. [5] introduced a Pyra-cryptosystem-based model for lightweight data exchange, peer-to-peer authentication, and session-key management in IoT networks. By addressing authentication and session-key establishment alongside data exchange, this line of research considers the security of an entire communication process rather than

treating confidentiality as an isolated requirement. Peer-to-peer authentication is especially relevant in decentralized environments where devices may communicate directly without relying on a permanently available central authority. Cherbal [6] proposed 3S-ALDDT, a triple-security scheme intended to authenticate lightweight devices and secure data transmission. This work reflects the need to combine device authentication with protection of transmitted information, particularly when IoT nodes cannot support computationally expensive security mechanisms. Evaluating such a scheme requires considering not only its authentication procedure but also its resilience to attacks on the communication process and its computational, communication, and storage requirements. Saif et al. [7] developed DLSEA-64, a dynamic, lightweight, symmetric encryption algorithm targeting pervasive sensing data in the Internet of Medical Things (IoMT). Medical sensing environments require efficient protection of continuously generated data because devices may have limited energy resources while processing information that is sensitive from both a privacy and confidentiality perspective. Dynamic symmetric encryption is intended to support efficient data protection, although the strength of such an approach must be established through a rigorous analysis of its cryptographic construction, key management, and resistance to relevant attacks. In a related direction, Peivandizadeh et al. [8] proposed an authenticated key agreement protocol for IoT communication that emphasizes compact operations and efficient establishment of shared session keys. The use of Chebyshev polynomials and probabilistic physical unclonable functions (PUFs) in this approach is intended to support authentication and privacy in device-to-gateway communication while limiting the computational requirements imposed on IoT devices. PUF-based mechanisms derive device-specific responses from physical characteristics, whereas polynomial-based operations can be used to construct lightweight cryptographic procedures. Their security, however, depends on the unpredictability and reliability of the PUF responses and the precise mathematical properties of the protocol. Zeng et al. [9] extended the discussion to secure distributed semantic communication for heterogeneous IoT environments. Unlike conventional communication systems, which primarily focus on the reliable and confidential transfer of transmitted bits, semantic communication also raises questions concerning the integrity, authenticity, and trustworthiness of the information represented by the exchanged data. Their work highlights the need to protect semantic representations, model updates, and communication processes against manipulation, thereby extending IoT security considerations beyond traditional cryptographic channels.

Another important research direction concerns privacy-preserving authentication and secure data sharing across organizational and administrative boundaries. Chen et al. [10] proposed an anonymous authentication and secure data-transmission scheme based on lightweight trust and identity management. Such approaches seek to verify the legitimacy of participating devices or users without unnecessarily disclosing their real identities during routine communication. This is particularly important in IoT deployments where communication patterns and device identifiers can reveal user activities even when message contents are encrypted. The effectiveness of anonymous authentication therefore depends on balancing identity protection with accountability and the ability to revoke compromised credentials when necessary. Chai et al. [11] introduced BSCDA for secure cross-domain data access in IoT. Cross-domain access is challenging because devices and services belonging to different administrative domains may not share the same trust relationships, identity infrastructure, or access-control policies. A secure solution must consequently establish an appropriate basis for inter-domain trust while ensuring that only authorized entities can access protected resources. This problem becomes increasingly important in collaborative IoT applications in which data are shared among multiple service providers or institutions.

Building on the cross-domain security direction, Xingxing Chen et al. [12] proposed a data-transmission scheme for industrial IoT that integrates edge-cloud computing with blockchain-related mechanisms. Industrial environments often involve geographically distributed devices, heterogeneous networks, and multiple administrative entities, making centralized authentication and data management difficult to scale. Lightweight authentication and relationship-key construction can help establish secure communication relationships between participating entities while reducing the overhead associated with repeated inter-domain interactions. The integration of edge-cloud infrastructure also provides opportunities to distribute processing between nearby edge nodes and centralized cloud services. However, the resulting security and performance depend on the trust assumptions associated with the edge nodes, the key-management procedures, and the underlying blockchain architecture. Balraj et al. [13] addressed secure communication in wireless sensor networks through a lightweight blockchain-based scheme. Blockchain mechanisms can support shared records and improve the traceability of security-related operations in distributed networks, but their application to resource-constrained sensor nodes requires careful consideration of storage, communication, and consensus costs. A lightweight design must therefore ensure that the security and accountability benefits justify the additional infrastructure overhead.

Fu et al. [14] investigated energy-aware secure routing using two-way trust evaluation. In wireless sensor networks, compromised or unreliable forwarding nodes can disrupt communication, selectively drop packets, or degrade network performance. Trust-based routing attempts to reduce these risks by evaluating the behavior and reliability of participating nodes, while energy-aware optimization seeks to prevent excessive resource consumption and premature node failure. Combining these objectives is important because a routing strategy that maximizes security without considering residual energy may reduce the operational lifetime of the network. Nevertheless, trust-based mechanisms must account for inaccurate observations, collusion, and malicious nodes that behave correctly during an initial observation period. Ranjan and Kumar [15] proposed APPS, an authentication-enabled privacy-protection scheme incorporating authentication, encryption, and data-retrieval phases. This approach considers security throughout the data-access process, rather than limiting protection to the initial establishment of communication. Authentication determines whether a requester is legitimate, encryption protects information against unauthorized disclosure, and controlled retrieval supports secure access to stored data. For resource-constrained IoT devices, the main design challenge is to coordinate these functions without introducing excessive cryptographic overhead or weakening access-control guarantees.

In 2025, research increasingly addressed application-specific authentication requirements in medical IoT, transportation, and wireless sensor networks. Lee et al. [16] proposed PLAKA-MD, a PUF-based lightweight authentication and key agreement scheme for medical devices in the Internet of Healthcare Things (IoHT). The approach combines biometric information, PUFs, and fuzzy extractors to strengthen user and sensor authentication. Biometric information can contribute to user verification, while PUFs provide device-related characteristics that can be used to support device authentication without relying exclusively on conventional stored secrets. Fuzzy extractors are relevant because biometric measurements and physical responses may vary slightly between observations; they can derive reproducible cryptographic material from sufficiently similar inputs under appropriate security assumptions. Integrating these mechanisms can provide multiple sources of authentication evidence, but it also introduces challenges related to biometric privacy, measurement variability, helper-data protection, and the reliability of device-specific responses.

Consequently, evaluating the protocol requires considering both its security properties and its suitability for medical devices with limited processing and storage capabilities.

Lee et al. [17] subsequently introduced LAMT, a lightweight and anonymous authentication scheme for medical IoT services. In contrast to approaches that primarily focus on device legitimacy, anonymous authentication additionally seeks to limit the exposure of identifiable information during service access. This is especially important in healthcare, where communication metadata may disclose sensitive information about patients, medical devices, or patterns of treatment. The challenge is to preserve anonymity without preventing authorized entities from managing credentials, revoking compromised identities, or investigating misuse when necessary. Parameswarath et al. [18] proposed a quantum-safe authentication protocol for IoT-enabled transportation systems by combining post-quantum cryptography with lightweight cryptographic primitives. Transportation systems increasingly rely on connected devices that exchange operational and potentially safety-critical information, making long-term confidentiality and authentication important design requirements. Conventional public-key mechanisms based on problems such as integer factorization or elliptic-curve discrete logarithms may be vulnerable to sufficiently capable quantum computers. Incorporating post-quantum cryptography is therefore intended to address this threat, although the practical benefits depend on the selected algorithms, public-key and signature sizes, processing requirements, and compatibility with constrained devices. This work illustrates the growing importance of considering both current resource limitations and future cryptographic threats when designing IoT authentication protocols.

Gui et al. [19] presented Once, a unified, secure, and lightweight key agreement scheme designed for IoT deployments with both single-gateway and multigateway architectures. Gateway configuration affects how devices establish trust, exchange key material, and maintain secure sessions when communication is distributed across multiple access points. A unified protocol can reduce the need for separate security procedures across different network configurations, potentially simplifying implementation and deployment. However, its effectiveness must be assessed in terms of authentication guarantees, resistance to attacks involving multiple gateways, communication overhead, and the management of session keys when devices move between network segments. In the context of wireless sensor networks, Byun et al. [20] proposed an improved three-factor mutual authentication scheme incorporating biometrics and PUFs. Three-factor authentication aims to combine distinct categories of evidence, commonly a knowledge factor, a possession factor, and an inherence factor. The integration of biometrics and PUF-based device characteristics is intended to strengthen authentication and address weaknesses in earlier WSN schemes. Nevertheless, stronger factor combinations do not automatically guarantee stronger security: the protocol must also protect stored verification information, handle compromised factors, and resist impersonation and replay attacks. Lee et al. [21] proposed an EAP-based privacy-preserving authentication and key agreement protocol for WLAN-IoT devices. The approach combines one-time ECC-based registration with lightweight symmetric-key authentication to support privacy-preserving access and efficient reconnection. This design addresses a practical issue in wireless environments, where repeatedly performing expensive authentication procedures can increase latency and consume limited device resources. Its effectiveness depends on the security of the initial registration, the protection and renewal of symmetric credentials, and the ability to prevent replay or session-hijacking attacks during subsequent connections.

More recent work has also investigated edge-assisted authentication, specialized device-to-gateway protocols, and intelligent threat detection. Swamy et al. [22] proposed a hybrid device-authentication algorithm for edge-based IoT networks, using edge com-

puting to support constrained devices and improve smart-home security. Edge-assisted security can move some computationally demanding operations closer to the devices, reducing the need for continuous interaction with a remote cloud server. This arrangement may improve response times and reduce communication overhead, but it also introduces trust and availability requirements for the edge infrastructure. A secure design must consider compromised edge nodes, unauthorized devices, and the consequences of losing connectivity to the authentication service. In a complementary direction, Pu et al. [8] proposed an authenticated key agreement protocol for device-to-gateway communication using Chebyshev polynomials and probabilistic PUFs. Device-to-gateway authentication is a fundamental requirement in IoT networks because gateways often provide access to broader networks and cloud services. Establishing a shared session key after verifying both communicating parties can protect subsequent exchanges against unauthorized access and passive observation. The use of PUF-based mechanisms is intended to associate authentication with device-specific physical characteristics, while the polynomial-based construction supports the protocol's key-agreement operations. A thorough assessment should examine the construction's assumptions, the unpredictability of PUF responses, resistance to replay and impersonation, and the overhead associated with the complete protocol rather than individual cryptographic operations alone.

Zeng et al. [9] further emphasized security challenges in distributed semantic communication. This direction is particularly relevant to IoT applications that employ distributed learning or semantic representations to reduce communication costs. Protecting such systems requires more than encrypting transmitted messages: attackers may manipulate model updates, introduce misleading semantic information, or exploit weaknesses in the usage phase of a communication system. Accordingly, integrity protection, trust management, and robust authentication must be considered across model development, update distribution, and operational deployment. These requirements complement conventional authentication and key-agreement mechanisms, illustrating that the security of emerging IoT architectures depends on protecting both communication channels and the higher-level processes that use the transmitted information.

Finally, several studies have focused on strengthening authentication, energy-aware communication, and anomaly detection in specialized IoT environments. Jhariya et al. [23] proposed a mutual authentication framework combining digital device fingerprinting with graph-based anomaly detection for dynamic machine-to-machine (M2M) environments. Device fingerprinting seeks to distinguish legitimate devices through identifiable characteristics, while graph-based analysis can reveal suspicious relationships or behavioral patterns among communicating entities. Integrating these mechanisms offers a broader security perspective by combining identity verification with the detection of potentially malicious interactions. Such a framework may complement cryptographic authentication, although fingerprint stability, false-positive rates, graph-update costs, and the effectiveness of detection against adaptive adversaries require careful evaluation. Verma and Jha [24] proposed a secure and energy-aware data-transmission framework for IoT-enabled wireless sensor networks based on cluster-oriented secure optimal routing. Clustering can organize sensor nodes into groups and reduce the communication burden associated with direct transmissions to a central destination. By incorporating security and energy considerations into routing decisions, the approach seeks to improve transmission reliability and extend network lifetime. The effectiveness of this strategy depends on the selection of cluster heads, the cost of maintaining routing information, and the ability to prevent compromised nodes from exploiting their position in the network.

Sudhakar et al. [25] introduced a lightweight two-factor authentication scheme for IoT-enabled wireless sensor networks using ECC and a fuzzy verifier. Two-factor authen-

tication aims to prevent an attacker from gaining access solely by compromising one authentication factor, while ECC supports public-key operations with comparatively compact parameters. A fuzzy verifier can accommodate controlled variation in verification inputs, depending on its construction, thereby supporting authentication in settings where exact input matching may be impractical. However, the security benefits depend on how verification information is stored and protected, how verification errors are bounded, and whether password-guessing and offline-verification attacks are adequately addressed. Shang et al. [26] proposed a three-factor ECC-based authentication and key agreement protocol for wireless medical sensor networks, considering stronger threat models that include insider attacks and ephemeral-secret leakage. The latter threat is particularly important because temporary secret values used during session establishment may be exposed even when long-term credentials remain protected. A robust protocol should therefore limit the consequences of ephemeral-secret compromise and establish explicit guarantees concerning session-key security, forward secrecy where applicable, and resistance to insider manipulation. This is especially significant in medical sensor networks, where compromised authentication can undermine the confidentiality and integrity of sensitive patient information.

Overall, the reviewed studies demonstrate a progression from lightweight authentication and key agreement toward more integrated IoT security frameworks combining privacy preservation, secure data sharing, energy-aware routing, device-specific authentication, edge computing, and intelligent anomaly detection. Nevertheless, these approaches address different threat models and deployment assumptions, and their security properties cannot be compared solely on the basis of their cryptographic primitives or claimed computational efficiency. A meaningful comparison should examine the authentication guarantees, resistance to relevant attacks, formal verification methods, computational and communication costs, storage requirements, and suitability for the intended application environment. In particular, medical IoT and healthcare authentication require careful attention to patient privacy, constrained medical devices, secure session-key establishment, and the consequences of compromised credentials. Furthermore, although ECC-based and other classical lightweight mechanisms remain attractive for constrained environments, the emergence of quantum-capable adversaries motivates investigation of post-quantum alternatives. These observations provide a basis for identifying the limitations of existing approaches and motivating authentication protocols that combine rigorous security verification with practical efficiency and, where required, resistance to quantum attacks.

3. Preliminaries

This section presents the foundational concepts and preliminaries associated with elliptic curve cryptography, physical unclonable functions, ChaCha20-Poly1305 AEAD, HKDF, and the BLAKE3 hash function.

3.1. Elliptic Curve Cryptography (ECC)

ECC is a public-key cryptography algorithm based on elliptic curves over finite fields. It has attracted considerable attention as an effective means of ensuring security in resource-constrained environments. This cryptosystem provides the same level of protection as the RSA cryptosystem but with significantly shorter key lengths, thereby involving substantially less computational overhead [27]. In this subsection, the foundational algebraic concepts of ECC are outlined. Let $\mathbb{F}_q$ denote a finite field of prime order $q > 3$. An elliptic curve E defined over $\mathbb{F}_q$, denoted as $E/\mathbb{F}_q$, is given by the simplified Weierstrass equation [28]:

$$y^2 = x^3 + ax + b \tag{1}$$

where $a, b \in \mathbb{F}_q$ and the discriminant satisfies $\Delta = -16(4a^3 + 27b^2) \neq 0$ to guarantee that the curve is non-singular. The set of $\mathbb{F}_q$-rational points on E, denoted by $E(\mathbb{F}_q)$, consists of all affine coordinate pairs $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$ satisfying Equation (1), together with a special point called the point at infinity, denoted by $\mathcal{O}$:

$$E(\mathbb{F}_q) = \left\{(x, y) \in \mathbb{F}_q \times \mathbb{F}_q \mid y^2 = x^3 + ax + b\right\} \cup \{\mathcal{O}\} \tag{2}$$

Under the standard chord-and-tangent geometric addition law, the set of points $(E(\mathbb{F}_q), +)$ forms an additive Abelian group, where the point at infinity $\mathcal{O}$ acts as the identity element such that $P + \mathcal{O} = P$ for all $P \in E(\mathbb{F}_q)$. Given a base point $P \in E(\mathbb{F}_q)$ and a scalar integer $k \in \mathbb{Z}_q^*$, elliptic curve scalar multiplication (point multiplication) is defined as the repeated group addition of P to itself k times:

$$kP = \underbrace{P + P + \cdots + P}_{k \text{ times}} \tag{3}$$

The cryptographic security of ECC rests upon the computational intractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP): given a base generator point $G \in E(\mathbb{F}_q)$ of prime order n and a public point $Q = kG$, it is computationally infeasible for any probabilistic polynomial-time (PPT) adversary to determine the scalar k.

3.2. Physical Unclonable Functions (PUFs)

A PUF is a hardware-based security primitive that exploits manufacturing variations in integrated circuits to generate unique device-specific responses. PUFs provide lightweight device authentication and secure key generation without permanently storing secret keys in the memory [29]. Formally, a PUF is characterized by a challenge–response pair mapping $F : \{0,1\}^n \rightarrow \{0,1\}^m$ such that:

- for any given challenge C, the response R = F (C) is reproducible on the same device across environmental variations;
- it is computationally infeasible to predict R for a new challenge or to reproduce R on a different physical device, even with full knowledge of the device's design [30].

The current literature extensively investigates SRAM-based PUF schemes in which the start-up values of uninitialized SRAM cells are exploited to generate unique device fingerprints. In particular, recent studies have demonstrated the efficacy of PUF-based challenge–response mechanisms for establishing lightweight, hardware-bound authentication in thing-to-thing (T2T) and autonomous drone-to-drone (D2D) IoT networks without permanently storing secret keys in non-volatile memory [31,32]. SRAM-PUFs are particularly suitable for constrained IoT devices because they can reuse existing SRAM components that are already present in most systems, avoiding the need for additional dedicated hardware resources [29].

3.3. ChaCha20-Poly1305 AEAD

ChaCha20-Poly1305 is a widely deployed scheme that encrypts and authenticates data in a single operation. It combines the ChaCha20 stream cipher for confidentiality and Poly1305 for integrity and authentication [33,34]. ChaCha20-Poly1305 uses a 256-bit key, 96-bit nonce, plaintext, and associated data. The output ciphertext $C = \text{ChaCha20}(K, \text{nonce}, m)$ is combined with an authentication tag $T = \text{Poly1305}(K, C \parallel AD)$ [33,35]. The resulting construction provides IND-CPA confidentiality and INT-CTXT integrity simultaneously, with a 128-bit authentication tag that renders forgery infeasible with a probability of at most 2^{-128}.

On software-only ARM Cortex-M class microcontrollers lacking AES hardware acceleration, ChaCha20-Poly1305 is empirically faster than AES-128-CBC with HMAC-SHA256 [36].

3.4. HKDF and BLAKE3

The HMAC-based Extract-and-Expand Key Derivation Function (HKDF) is a standardized cryptographic mechanism formalized in IETF RFC 5869 and recommended by NIST SP 800-56C [37]. Designed to convert shared secrets into cryptographically strong keying material, HKDF operates through two distinct sequential stages:

1. Extraction Stage: A pseudorandom key (PRK) is extracted from the input key material (IKM) using an optional salt to concentrate the entropy of potentially non-uniformly distributed secrets, such as Diffie–Hellman shared values:

$$PRK = \mathrm{HMAC}_{\mathrm{Hash}}(salt, IKM).$$

2. Expansion Stage: The extracted key (PRK) is expanded to generate output keying material (OKM) of the desired length, bound to an application-specific context parameter ($info$):

$$T(1) = \mathrm{HMAC}_{\mathrm{Hash}}\big(PRK,\ info \parallel 0x01\big).$$

For longer outputs, HKDF generates successive blocks as specified in RFC 5869:

$$T(i) = \mathrm{HMAC}_{\mathrm{Hash}}\big(PRK,\ T(i-1) \parallel info \parallel i\big),$$

and concatenates them to obtain the required output:

$$OKM = T(1) \parallel T(2) \parallel \cdots \parallel T(N).$$

A vital parameter in the HKDF expansion phase is the application-specific contextual information string ($info$). As described in NIST SP 800-56C, this parameter provides cryptographic domain separation, ensuring that independent session keys are derived for distinct protocol roles and operational contexts, even when the same input key material (IKM) is used. BLAKE3 is a modern cryptographic hash function optimized for high throughput and parallel processing [38]. It uses a tree-based hashing architecture and produces a 256-bit digest, providing a nominal 128-bit collision-resistance level and a 256-bit preimage-resistance level.

When instantiated within the HKDF framework using HMAC with BLAKE3 as the underlying hash function, denoted as HMAC-BLAKE3, the construction combines the extract-and-expand paradigm with the performance characteristics of BLAKE3. However, this instantiation should be distinguished from the specific hash-function instantiations explicitly standardized in RFC 5869 and NIST SP 800-56C. Its security depends on the suitability of HMAC-BLAKE3 as a pseudorandom function and on the security properties of the underlying implementation. On resource-constrained embedded platforms, including ARM Cortex-M microcontrollers without dedicated hash acceleration, HMAC-BLAKE3 may offer performance advantages over HMAC-SHA-256. However, the actual throughput improvement depends on the processor, implementation, and optimization level and should be established through empirical benchmarking.

4. System Models

4.1. Network Model

As shown in Figure 1, the proposed FRACE-IoT scheme is designed as a heterogeneous four-tier IoT network comprising the following layers:

- Tier 1 Constrained IoT Devices (CIoT): These are resource-constrained sensor nodes that monitor physical environments, encrypt their readings, and transmit them to their associated controllers. In our network model, we used a FIT IoT-LAB M3 device [39]

as a CIoT, which is equipped with an ARM Cortex-M3 microcontroller, 64 KB of RAM, and 256 KB of flash memory (Figure 2b). Furthermore, they function within battery-constrained energy budgets that may not allow recharging or replacement during the device's operational lifetime. Each CIoT device is embedded with an SRAM PUF and enrolled at the gateway server prior to network deployment.

- Tier 2 Controller Devices (CD): Controllers are cluster-head nodes with substantially greater computational resources than CIoT devices. Each controller manages a cluster of CIoT devices, collects and aggregates their individually encrypted transmissions, re-encrypts the aggregated data under the CD-GW session key, and forwards them to the gateway. Similar to CIoT devices, controllers are embedded with SRAM PUFs and enrolled at the gateway prior to deployment. In this architecture, we used a Nordic nRF52840DK [40] IoT device as a controller, which is equipped with an ARM Cortex-M4 processor, 1 MB of flash memory, and 256 KB of RAM (Figure 2a).
- Tier 3 Smart Gateway (GW): A trusted local server that initializes the system, registers new IoT devices via PUF enrollment, decrypts and stores received data, and mediates end-user data access. The gateway maintains a secure enrollment database mapping each device identifier to its enrolled PUF challenge–response parameters and derived public key. The gateway is assumed to be physically and computationally secure. We used a mobile workstation as the gateway. It was equipped with an Intel Core i7-11850H processor running at 2.50 GHz, 32 GB of RAM, and 1 TB of SSD. The gateway also maintains an up-to-date Revocation List (RL) and manages dynamic device joining and credential revocation during network operation, as described in Section 6.6.
- Tier 4 End-User Devices (EU): End-user devices are application-layer terminals such as laptops, smartphones, or management consoles that query the gateway for decrypted sensor data to support decision-making processes. They interact exclusively with the gateway tier and have no direct cryptographic relationship with CIoT devices or controllers.

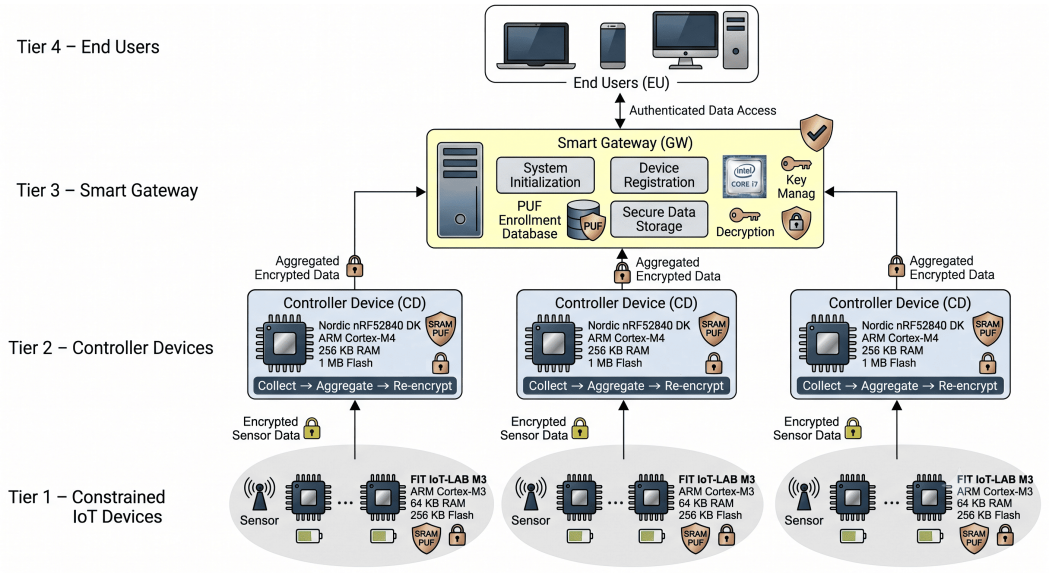
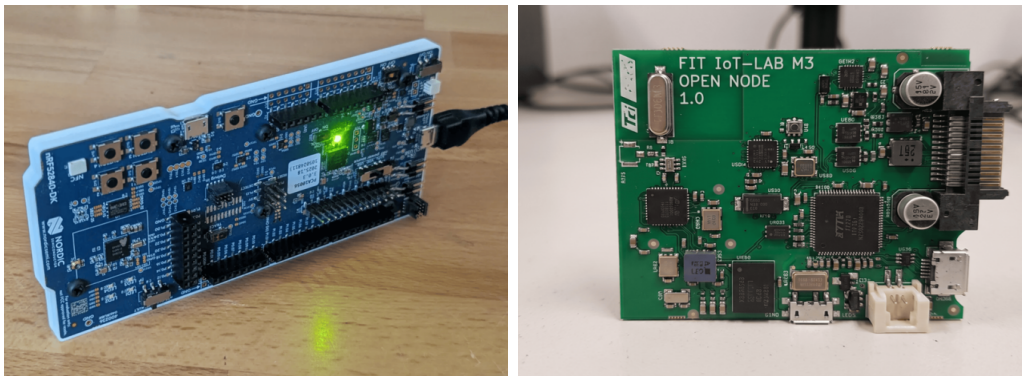


Figure 1. FRACE-IoT heterogeneous four-tier IoT network model.

4.2. Threat Model

The proposed scheme was evaluated within the standard Dolev–Yao adversary model [41], which was enhanced by incorporating a hardware threat. Specifically, adversary $\mathcal{A}$ is defined as

- Network capability: $\mathcal{A}$ has full control over the communication channel. It can intercept, read, modify, delete, delay, replay, and inject arbitrary messages on any link in the network.
- Computational bound: $\mathcal{A}$ is a probabilistic polynomial-time (PPT) algorithm. It cannot solve the ECDLP, invert a PUF function, break ChaCha20-Poly1305 AEAD, or mount a brute-force attack that requires more than 2^{128} operations.
- Key compromise: $\mathcal{A}$ may learn the session key K_{sh} of at most one completed session. Therefore, it cannot recover the session key of any other session (forward secrecy).
- Hardware threat: $\mathcal{A}$ may physically capture a device and attempt to clone it by copying its firmware. However, $\mathcal{A}$ cannot reproduce a PUF response on a different physical device because PUF responses are bound to the specific silicon instance of each device.
- Gateway trust: The gateway is assumed to be a trusted and tamper-resistant entity. The end-user devices are honest but outside the cryptographic perimeter.



(**a**) Nordic nRF52840DK board (**b**) Fit IoT-LAB M3 board

Figure 2. Experimental hardware platforms employed for network modeling and performance evaluation.

5. Cryptanalysis of Al-Hejri et al.'s Scheme

Although Al-Hejri et al. claimed that their lightweight secure and scaleable scheme [1] satisfies essential security requirements, a rigorous cryptanalysis revealed several critical vulnerabilities and design flaws. This section details these weaknesses, demonstrating how the scheme falls short in both the key-establishing and data transmission phases of communication. To provide a clear overview of the operational breakdown, Figure 3 illustrates the procedural workflow of Al-Hejri et al.'s scheme alongside the structural injection points of the identified vulnerabilities.

5.1. Vulnerability to State Desynchronization

The Al–Hejri et al. scheme suffers from a critical desynchronization vulnerability in its rolling pre–shared key mechanism between the BS and CH. During Key Establishment (phase 3), the CH encrypts a newly generated shared key using the current pre–shared key $K_{BS,CH}^{PSh}$, transmits it to the BS, and immediately deletes the current $K_{BS,CH}^{PSh}$ from its memory. Upon receiving this transmission, the BS generates a pre–shared key for the next round $K_{BS,CH}^{PSh'}$ and transmits it back to the CH. This architectural sequence introduces fatal flaws in desynchronization. If an active adversary intercepts, jams, or drops the return message from the BS (or if the packet is lost due to wireless interference), the CH will never receive the updated key. Because the CH prematurely deleted the old key, it was left without any valid cryptographic material to communicate with the BS in subsequent rounds. Consequently, an attacker can permanently lock any CH out of the network simply

by dropping a single packet during the key update process, making the network vulnerable to a DoS attack.

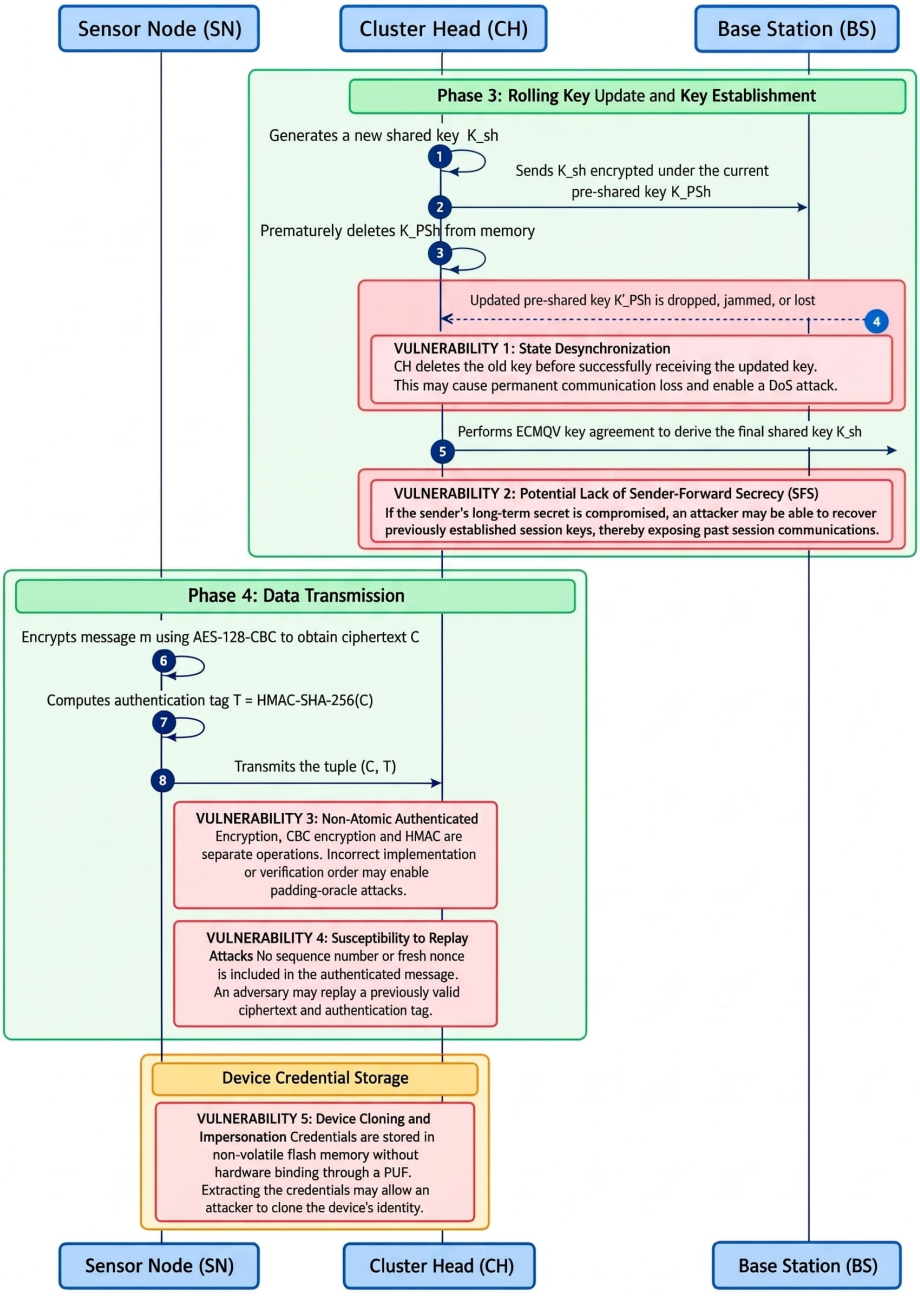


Figure 3. Protocol execution workflow of Al-Hejri et al.'s scheme and the corresponding cryptographic failure points: (1) state desynchronization caused by premature deletion of $K_{BS,CH}^{PSh}$; (2) lack of sender-forward secrecy; (3) non-atomic encrypt-then-MAC construction; (4) lack of sequence numbers or fresh nonces enabling payload replay; and (5) lack of hardware-bound silicon roots of trust enabling firmware cloning.

5.2. Lack of Sender-Forward Secrecy and Session Key Isolation

The Al–Hejri et al. scheme derives the session key K_{sh} directly from the unhashed ECMQV group element, omitting an extract-and-expand Key Derivation Function (KDF) capable of enforcing session-specific domain separation and cryptographic entropy extraction. This construction exhibits two critical security vulnerabilities regarding forward secrecy: First, although the ECMQV exchange incorporates ephemeral scalars, the absence of a session-binding KDF with independent, session-unique salts renders the established

session key a direct, deterministic algebraic function of the long-term static private keys stored in the devices' non-volatile memory. Because the communicating sensor nodes permanently store their long-term static private keys in unencrypted flash storage without hardware-bound silicon protection, an adversary who physically captures a node can readily extract its static credentials. Under the forward secrecy model formalized by Blake–Wilson et al. [42], a secure protocol must guarantee that the compromise of long-term credentials does not retroactively compromise past session keys. In Al-Hejri et al.'s design, an adversary possessing the extracted static key can retroactively compute past session keys directly from intercepted public transcripts, completely destroying session confidentiality. Second, the rolling pre-shared key mechanism between the cluster head and the base station links successive session secrets without independent, forward-secure ratcheting. If the current pre-shared key $K_{BS,CH}^{PSh}$ is compromised or exposed through side-channel leakage, the adversary can invert or trace the key generation sequence backwards, compromising previously exchanged sensor telemetry. Consequently, the scheme fails to provide key isolation across sessions, exposing past traffic to retroactive decryption upon credential compromise.

5.3. Non-Atomic Authenticated Encryption

The Al–Hejri et al. scheme's data transmission phase employs a two–pass cryptographic construction, encrypting messages via AES-128-CBC and subsequently authenticating the ciphertext using HMAC-SHA256. This architecture introduces three vulnerabilities to the Al-Hejri et al. scheme. First, it effectively doubles the computational overhead per message (requiring separate full passes for both AES-CBC and HMAC-SHA256), whereas AEAD schemes achieve confidentiality and integrity in a single pass. Second, the security of Al–Hejri et al.'s scheme is highly sensitive to the implementation ordering. Although encrypt-then-MAC is a provably secure paradigm, the Al–Hejri et al. scheme specification fails to strictly enforce it; an inadvertent MAC–then–encrypt implementation introduces a severe decryption oracle vulnerability [43]. Third, the authentication tag is not bound to the associated contextual data (such as a session identifier or sequence counter), making the Al-Hejri et al. scheme vulnerable to cross-session replay attacks.

5.4. Susceptibility to Replay Attacks in Data Transmission

The Al–Hejri et al. scheme lacks anti–replay mechanisms during its most frequently executed phase, that is, the data transmission phase. During this phase, a sensor node encrypts the sensed data, computes a message authentication code (MAC) using the session key, and then transmits the tuple $(C(m), V(m))$ to a Cluster Head (CH). Because this message format does not include monotonic sequence numbers or fresh nonces, an adversary can passively intercept a valid payload and retransmit it. Consequently, the CH successfully verifies the MAC and accepts the replayed message as legitimate, leading to incorrect processing of duplicate or stale data.

5.5. Vulnerability to Device Cloning and Impersonation

The Al–Hejri et al. scheme does not include any mechanism to link a device's cryptographic identity to its hardware. As a result, an adversary who obtains a copy of the device firmware, whether through a supply chain compromise, physical extraction, or software exfiltration, can replicate the device identity on arbitrary hardware and participate in the protocol as a legitimate node. This attack vector is particularly significant in IoT deployments, where devices are mass-produced and often operate in physically unprotected environments. Without hardware binding, Al–Hejri et al.'s claimed mutual authentication effectively reduces to the protection of software credentials, leaving it vulnerable to credential theft, with no inherent mechanism to mitigate this risk.

5.6. Absence of Formal Security Verification

The security analysis of the Al–Hejri et al. scheme is based entirely on informal claim-by-claim reasoning. Automated verification tools were not used to validate the protocol against a formal adversarial model. Experience in cryptographic protocol design shows that such informal analyses are often insufficient to detect subtle and logical flaws. Protocols that initially appeared secure under intuition have later been broken through formal methods. In particular, automated verification tools, such as AVISPA [44,45], have revealed vulnerabilities in many protocols that were previously supported only by informal security arguments.

6. Proposed FRACE-IoT Scheme

The FRACE-IoT scheme consists of five phases: PUF enrollment, key establishment between CIoT and controller devices, key establishment between controllers and the gateway, data transmission, and authenticated end–user access. The first phase is executed offline during device deployment. The key establishment phases occur at the beginning of each session, whereas the data transmission phase runs continuously during normal operation. The scheme is described below, with all notations defined in Table 1. Figure 4 illustrates the message sequence chart of the proposed FRACE-IoT scheme.

Table 1. List of notations.

Notation	Description
$CIoT$	Constrained IoT Device
CD	Controller Device
GW	Smart Gateway
K_x^{SPr}, K_x^{SPu}	Static private/public key of x
K_x^{EPr}, K_x^{EPu}	Ephemeral private/public key of x
$Z_{\mathrm{IoT,CD}}^{\mathrm{static}}$	Static DH share: $K_{\mathrm{IoT}}^{\mathrm{SPr}} \cdot K_{\mathrm{CD}}^{\mathrm{SPu}}$
$Z_{\mathrm{CD,GW}}^{\mathrm{static}}$	Static DH share: $K_{\mathrm{CD}}^{\mathrm{SPr}} \cdot K_{\mathrm{GW}}^{\mathrm{SPu}}$
$\mathrm{DH}_x^{\mathrm{ES}}$	Ephemeral-static DH share: $K_x^{\mathrm{EPr}} \cdot K_{\mathrm{peer}}^{\mathrm{SPu}}$
$K_{x,y}^{\mathrm{sh}}$	Session key between x and y
τ_x	PUF authentication token of x
n_x	Fresh nonce from x
$a \Vert b$	Bitstring concatenation
$\mathrm{AEAD}(K, N, m, A)$	ChaCha20-Poly1305 encryption
$\mathrm{HKDF\text{-}BLAKE3}(\cdot, \cdot, \cdot)$	HKDF-BLAKE3 key derivation
seq	Monotonic sequence counter
T	Poly1305 auth. tag (128 bits)
R_x	PUF response to challenge C
ID_x	Unique identifier of entity x
AD	Associated data

6.1. Phase 1: PUF Enrollment and Static DH Pre–Computation

Phase 1 is a one-time provisioning procedure executed before gateway network deployment, which serves as the architectural foundation of trust. This phase encompasses two critical responsibilities: enrolling all network devices and precomputing static Diffie–Hellman (DH) shares to facilitate a highly efficient single–pass key derivation across both communication links.

Step 1: Device Enrollment

For every device d within the network (both CIoT devices and controllers), the gateway executes the following enrollment protocol:

1. The GW applies a unique device-specific challenge C_d to the PUF of the device to obtain the cryptographic response: $R_d = \text{PUF}(C_d)$;
2. The device's static private key is deterministically derived using the HKDF–BLAKE3 function, binding the PUF response to the device's unique identity: $K_d^{\text{SPr}} = \text{HKDF-BLAKE3}(R_d, \text{“static-key”}, ID_d)$;
3. The corresponding static public key is computed via a single scalar multiplication: $K_d^{\text{SPu}} = K_d^{\text{SPr}} \cdot G$;
4. The GW stores the resulting public tuple $(ID_d, C_d, K_d^{\text{SPu}})$ in its secure enrollment database, where $R_d = PUF_d(C_d)$ serves as the pre-measured reference response used by authorized verifiers (CD and GW) to validate runtime authentication tokens without requiring physical access to the device's silicon.

Crucially, long-term asymmetric private keys (K_d^{SPr}) are never stored in non-volatile storage on the devices. Instead, K_d^{SPr} is dynamically reconstructed on demand from the SRAM-PUF response at runtime and immediately expunged from volatile RAM after key derivation. Because key reconstruction relies on HKDF-BLAKE3, this operation introduces negligible computational latency (0.75 ms on the ARM Cortex-M4 platform, as benchmarked in Section 8.1). Furthermore, on CIoT sensor nodes, live derivation of K_{IoT}^{SPr} is bypassed entirely during online sessions by utilizing the pre-computed static share $Z_{IoT,CD}^{\text{static}}$, thereby ensuring that the resource-constrained tier incurs zero asymmetric scalar multiplication overhead for static key recovery.

Step 2: Static DH Share Pre–Computation

To enable lightweight single–pass key derivation at both architectural links, the gateway precomputes static DH shares as follows:

1. For each CIoT and CD pair, the shared static DH component is defined as: $Z_{\text{IoT,CD}}^{\text{static}} = K_{\text{IoT}}^{\text{SPr}} \cdot K_{\text{CD}}^{\text{SPu}}$
2. For each CD and GW pair, the shared static DH component is defined as follows: $Z_{\text{CD,GW}}^{\text{static}} = K_{\text{CD}}^{\text{SPr}} \cdot K_{\text{GW}}^{\text{SPu}}$
3. The GW loads the specific challenge C_{IoT}, the Controller's static public key $K_{\text{CD}}^{\text{SPu}}$, and the pre–computed static DH share $Z_{\text{IoT,CD}}^{\text{static}}$ onto the CIoT device through a secure, offline enrollment provisioning channel.
4. Similarly, the GW loads the specific challenge C_{CD}, along with both necessary static DH shares $Z_{\text{IoT,CD}}^{\text{static}}$ and $Z_{\text{CD,GW}}^{\text{static}}$ onto the Controller device through the secure offline enrollment provisioning channel.

6.2. Phase 2: Single-Pass Key Establishment (CIoT ↔ Controller)

While 0-RTT key exchange is an established principle in high-level transport protocols such as TLS 1.3, its application to severely constrained IoT tiers has traditionally been hindered by susceptibility to replay and state-synchronization vulnerabilities. In FRACE-IoT, this phase adapts the 0-RTT paradigm to resource-limited sensor links through a unidirectional, non-interactive architecture that combines ephemeral Diffie–Hellman dynamism with hardware-enforced SRAM-PUF verification. The CIoT device initiates a secure session without awaiting a handshake response from the CD, thereby conserving scarce radio transceiver energy. The CIoT device first generates an ephemeral Curve25519 key pair and a random 64-bit session nonce to guarantee session freshness (Steps 1 and 2 in Algorithm 1). It then computes a 64-bit truncated PUF token using the BLAKE3 hash of its PUF response and nonce (Step 3 in Algorithm 1). Subsequently, the CIoT device calculates a DH shared ephemeral-static secret $\text{DH}_{\text{IoT}}^{\text{ES}}$ by combining its ephemeral private key with the static public key of the controller (Step 4 in Algorithm 1). Using the HKDF-BLAKE3 expand function, the CIoT device derives the final session key $K_{\text{IoT,CD}}^{\text{sh}}$ by combining the ephemeral-static

secret $DH_{\text{IoT}}^{\text{ES}}$, the pre-computed static DH share $Z_{\text{IoT,CD}}^{\text{static}}$, the session nonce n_{IoT} , and the explicit domain separation string (Step 5 in Algorithm 1). This domain separation string cryptographically isolates the Tier 1-Tier 2 link, ensuring that the derived key space cannot collide with or be substituted into any other protocol phase or tier. It concludes the first part of this phase by transmitting a single initiation message $\mathcal{M}_1$ containing the device identifier, the ephemeral public key, the PUF token and the nonce to the CD and securely erasing its ephemeral private key and the DH secret to ensure forward secrecy (Steps 6–7 in Algorithm 1).

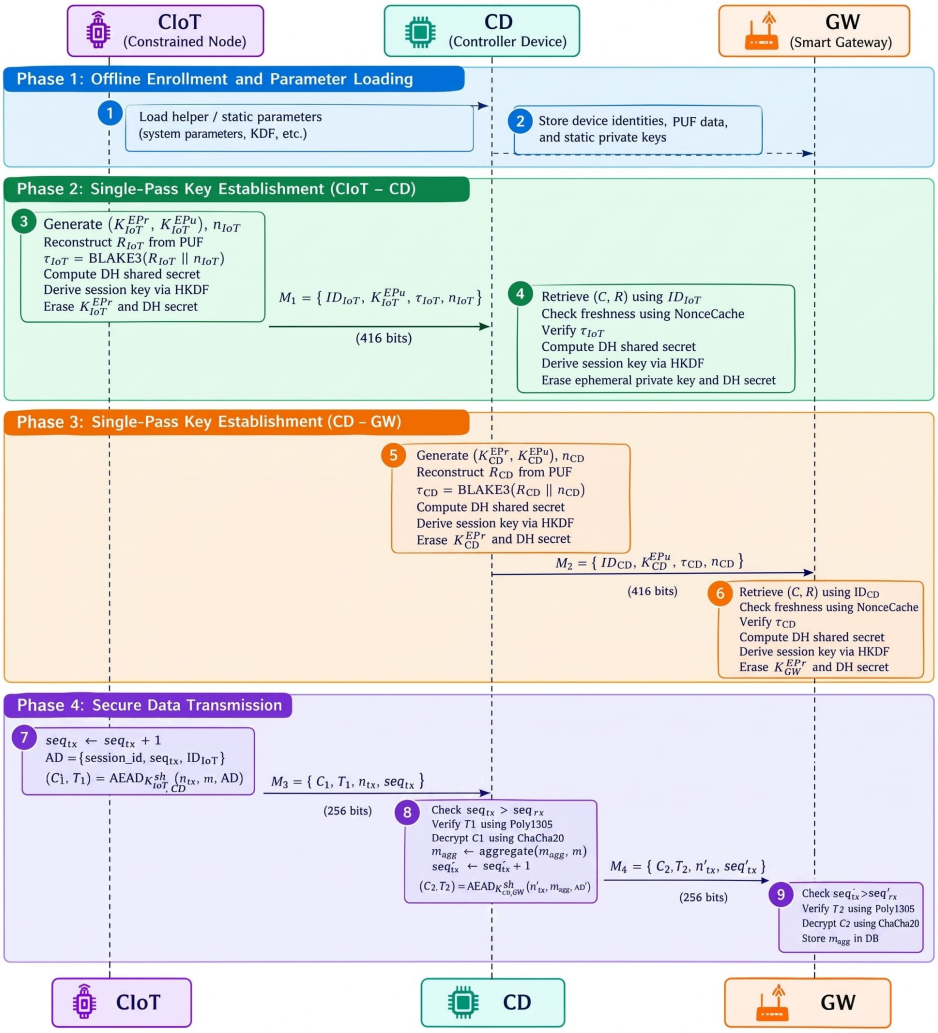


Figure 4. Message Sequence Chart illustrating the chronological operational flow of the proposed FRACE-IoT scheme across the phases: offline enrollment, single-pass key establishment (CIoT-CD and CD-GW), and secure data transmission.

Upon receiving the $\mathcal{M}_1$ message, the CD processes the request and derives $K_{\text{IoT,CD}}^{\text{sh}}$ without sending a response to the CIoT. The CD first derives its static private key on demand and retrieves the corresponding PUF challenge from the CIoT device from the enrollment database (Steps 8–9 in Algorithm 1). The device is then authenticated by reconstructing the PUF token and verifying it against the received token; if the device is not found or the verification fails, the connection is rejected, and the gateway is alerted (Step 11 in Algorithm 1).

To prevent the re-derivation of previously established session keys through replayed M_1 messages, the CD verifies the freshness of the received nonce n_{IoT} against an in-memory

64-bit sliding-window nonce cache, W_{nonce}. If n_{IoT} is already present in the cache, the message is rejected as a replay (Step 10, Algorithm 1), thereby preventing redundant session-key derivation and avoiding unnecessary elliptic-curve scalar multiplication overhead.

After successful authentication, the CD computes the shared ephemeral-static Diffie–Hellman secret using its static private key $K_{\text{CD}}^{\text{SPr}}$ and the received ephemeral public key of the CIoT: $DH_{\text{CD}}^{ES} = K_{\text{CD}}^{\text{SPr}} \cdot K_{\text{IoT}}^{\text{EPu}}$. (Step 11, Algorithm 1). By the commutative property of elliptic curve point multiplication, $K_{\text{IoT}}^{\text{EPr}} \cdot \left(K_{\text{CD}}^{\text{SPr}} \cdot G\right)$, which guarantees that $DH_{\text{CD}}^{ES} = DH_{\text{IoT}}^{ES}$. Consequently, both entities derive identical session keys $K_{\text{IoT,CD}}^{\text{sh}}$ using HKDF-BLAKE3. Finally, the CD securely erases all sensitive key material from memory (Step 13 in Algorithm 1).

6.3. Phase 3: Single–Pass Key Establishment (Controller ↔ Gateway)

This phase establishes a fresh and secure session key $K_{\text{CD,GW}}^{\text{sh}}$ between the CD and GW. The scheme proceeds analogously to Phase 2, but with the CD assuming the role of the initiator. The steps of this phase are illustrated in Algorithm 2. The CD first generates an ephemeral ECC key pair and a 64-bit nonce. It then evaluates the PUF associated with the CD challenge and produces the authentication token τ_{CD} (Steps 1–3 in Algorithm 2). Subsequently, the CD computes an ephemeral-static Diffie-Hellman secret $\text{DH}_{\text{CD}}^{\text{ES}}$ using its ephemeral private key $K_{\text{CD}}^{\text{EPr}}$ and the static public key of the GW $K_{\text{GW}}^{\text{SPu}}$. The resulting secret is combined with $Z_{\text{CD,GW}}^{\text{static}}$, the nonce n_{CD}, and the distinct domain separation string to derive $K_{\text{CD,GW}}^{\text{sh}}$ via HKDF-BLAKE3 (Steps 4–5 in Algorithm 2). This tag ensures strict cryptographic domain separation between the Tier 2-Tier 3 backbone link and the lower-tier sensor links, preventing cross-protocol key reuse and reflection attacks. The CD then sends the initiation message $\mathcal{M}_2$ to GW and immediately erases its ephemeral private key and the DH secret ($K_{\text{CD}}^{\text{SPr}}$, $\text{DH}_{\text{CD}}^{\text{ES}}$) to maintain forward secrecy (Steps 6–7 in Algorithm 2). After receiving the message $\mathcal{M}_2$, the GW retrieves the enrolled CD challenge corresponding to the CD identifier received (Step 8 in Algorithm 2). The GW checks whether the 64-bit nonce n_{CD} exists in the controller's active sliding-window cache $W_{\text{nonce}}(ID_{\text{CD}})$. If present, M_2 is discarded as a replay. Otherwise, n_{CD} is recorded in the cache (Step 9 in Algorithm 2). The GW then recomputes and verifies the PUF token using the nonce received from the CD. If a token is valid, the GW computes the same Diffie–Hellman secret using its static private key and the controller's ephemeral public key (Steps 10–11 in the Algorithm 2). Owing to the commutative property of the elliptic curve, Diffie–Hellman, CD and GW obtain the same shared secret key. GW therefore derives the identical session key $K_{\text{CD,GW}}^{\text{sh}}$ using the same HKDF–BLAKE3 inputs and finally deletes the temporary Diffie–Hellman key $\text{DH}_{\text{GW}}^{\text{ES}}$ from memory (Steps 12–13 in Algorithm 2).

6.4. Phase 4: Data Transmission

Upon successful key establishment, secure data transmission begins using ChaCha20-Poly1305 Authenticated Encryption with Associated Data (AEAD), providing confidentiality, integrity, and authentication of transmitted payloads. The end-to-end data transmission process (Algorithm 3) comprises four stages. First, the CIoT node increments its RAM-based transmission counter, $\text{seq}_{tx} \leftarrow \text{seq}_{tx} + 1$; constructs the associated data $AD = \{\text{session_id}, \text{seq}_{tx}, ID_{\text{IoT}}\}$; and generates a fresh 96-bit nonce n_{tx}. It encrypts the payload m using ChaCha20–Poly1305 and the session key $K_{\text{IoT,CD}}^{sh}$, producing $(C_1, T_1) \leftarrow \text{AEAD}\left(K_{\text{IoT,CD}}^{sh}, n_{tx}, m, AD\right)$. The message $M_3 = \{C_1, T_1, n_{tx}, \text{seq}_{tx}\}$ is then sent to the controller (Steps 1–5 in Algorithm 3). Second, the controller checks $\text{seq}_{tx} > \text{seq}_{rx}$ to prevent replay attacks and verifies the authentication tag T_1. Upon successful authentication, it decrypts C_1, updates $\text{seq}_{rx} \leftarrow \text{seq}_{tx}$, and aggregates the validated payloads into m_{agg} (Steps 6–10 in Algorithm 3). Third, the controller increments its backbone transmission counter, $\text{seq}'_{tx} \leftarrow \text{seq}'_{tx} + 1$; constructs the associated data AD'; and generates a fresh nonce

n'_{tx}. It encrypts m_{agg} using $K^{sh}_{CD,GW}$, obtaining $(C_2, T_2) \leftarrow \text{AEAD}(K^{sh}_{CD,GW}, n'_{tx}, m_{\text{agg}}, AD')$, and transmits $M_4 = \{C_2, T_2, n'_{tx}, \text{seq}'_{tx}\}$ to the gateway (Steps 11–15 in Algorithm 3). Finally, the gateway verifies $\text{seq}'_{tx} > \text{seq}'_{rx}$ and authenticates T_2. If both checks succeed, it decrypts C_2, updates $\text{seq}'_{rx} \leftarrow \text{seq}'_{tx}$, and securely stores m_{agg} in its database (Steps 16–20 in Algorithm 3). Messages failing replay or authentication checks are rejected at either receiving stage.

Sequence Counter Lifecycle, Storage, and Reboot Recovery

To prevent message replay, reordering, and injection attacks with minimal hardware overhead, the sequence number seq is governed by the following strategies:

- Initialization and In-Memory Volatile Storage: Upon session key derivation ($K^{sh}_{\text{IoT},CD}$ or $K^{sh}_{CD,GW}$), both communicating endpoints initialize their sequence counters to zero, $\text{seq}_{tx} \leftarrow 0$, and $\text{seq}_{rx} \leftarrow 0$, directly in volatile SRAM. Storing active counters exclusively in volatile RAM avoids repetitive, energy-costly Flash memory write cycles on constrained micro-controllers, preventing storage degradation and write-latency bottlenecks.
- Power Failure and Reboot Handling: If a device experiences a sudden power failure, crash, or hardware reset, volatile SRAM is cleared, expunging the active session key $K^{sh} \leftarrow \perp$ and resetting the counters, $\text{seq} \leftarrow \perp$. The node is strictly prohibited from resuming transmissions with stale session state. Instead, it must re-execute Phase 2 (Algorithm 1) to establish a fresh session key $K^{sh'}$ via new ephemeral randomness, under which seq_{tx} is cleanly reinitialized to 0 in RAM.
- Cross-Restart and Cross-Session Replay Protection: Replay protection across power cycles and session boundaries relies on cryptographic key isolation. Following a reboot, the node must re-execute Algorithm 1 using a fresh ephemeral scalar $K^{\text{EPr}'}_{\text{IoT}}$ and a fresh nonce n'_{IoT}, thereby establishing a distinct session key $K^{\text{sh}'}$. Consequently, historical telemetry frames M_3 intercepted before the reboot cannot be successfully authenticated under $K^{\text{sh}'}$, except with a forgery probability bounded by $P_{\text{forgery}} \leq 2^{-128}$. Furthermore, as discussed in Section 7.3, the sliding-window filter W_{nonce} helps detect and reject replayed handshake messages (M_1) across sessions. Persistent epoch counters or equivalent mechanisms for evolving receiver state are additionally required to prevent session-key recurrence following cache eviction.

6.5. Phase 5: End–User Authenticated Access

Authorized end users securely access the aggregated and decrypted data through the GW using a stateless session token. Following the successful validation of user credentials, the GW provides a time–bound session token to manage the ongoing access as follows:

1. Token Generation: The GW computes a secure stateless token using the BLAKE3 hash function. This operation cryptographically binds the end–user identity (ID_{EU}) to the current timestamp and a gateway–local master secret Gateway ($\text{GW}_{\text{secret}}$) as follows: $\text{token} \leftarrow \text{BLAKE3}(ID_{\text{EU}} \parallel \text{timestamp} \parallel \text{GW}_{\text{secret}})$. Similarly, an expiration horizon is defined using a strict time–to–live (TTL) parameter: $\text{expiry} = \text{timestamp} + \text{TTL}$.
2. Stateless Verification: For all subsequent data requests, the end-user device must append this token. The GW statelessly authenticates the request by recomputing the BLAKE3 hash using its local master secret and validating it against the submitted token.

Crucially, this end-user access phase is completely decoupled from the constrained edge tier of the network. It does not involve CIoT devices or CDs, ensuring that no additional computational, transmission, or cryptographic burdens are imposed on the resource-limited IoT infrastructure.

6.6. Dynamic Device Joining and Revocation Procedures

In practical, large-scale IoT deployments, network topologies are dynamic: new CIoT nodes are added to expand monitoring coverage, additional controller devices are deployed to balance cluster workloads or replace faulty gateways, and compromised nodes must be promptly revoked. FRACE-IoT provides native support for these runtime events without requiring network-wide re-initialization.

6.6.1. Dynamic Joining of a CIoT$_{new}$

When a CIoT$_{\text{new}}$ is deployed into an operational cluster managed by CD$_j$, the following procedure is executed:

1. The node is provisioned offline by the GW following Phase 1. The GW queries the device's PUF to obtain $R_{\text{new}} = \text{PUF}(C_{\text{new}})$ and derives the static key pair $(K_{\text{new}}^{\text{SPr}}, K_{\text{new}}^{\text{SPu}})$.
2. The GW identifies the managing controller CD_j, precomputes the static Diffie–Hellman share $Z_{\text{new},CD_j}^{\text{static}} = K_{\text{new}}^{\text{SPr}} \cdot K_{CD_j}^{\text{SPu}}$, and updates its local enrollment database with $(ID_{\text{new}}, C_{\text{new}}, R_{\text{new}}, K_{\text{new}}^{\text{SPu}})$.
3. Over the active session channel established in Phase 3, the GW securely transmits the admission record to CD_j using authenticated encryption with associated data:

$$GW \rightarrow CD_j: \quad \text{AEAD}\left(K_{CD_j,GW}^{\text{sh}}, n_{\text{join}}, \{ID_{\text{new}}, R_{\text{new}}, Z_{\text{new},CD_j}^{\text{static}}\}, AD\right).$$

4. The controller CD$_j$ decrypts and authenticates the received message before adding the new node's record to its local authorized-device table. Consequently, CIoT$_{\text{new}}$ can immediately initiate secure sessions by executing Algorithm 1, without disrupting the operation of neighboring CIoT nodes.

6.6.2. Dynamic Joining of a CD$_{new}$

When a new controller device CD_{new} is introduced to handle an expanding cluster or establish a new physical zone:

1. The GW initializes CD_{new} via its physical SRAM-PUF. The GW evaluates $R_{CD_{\text{new}}} = \text{PUF}(C_{CD_{\text{new}}})$ and derives the controller's static asymmetric key pair ($K_{CD_{\text{new}}}^{SPr}, K_{CD_{\text{new}}}^{SPu}$).
2. The GW computes the static controller–gateway Diffie–Hellman share $Z_{CD_{\text{new}},GW}^{\text{static}} = K_{CD_{\text{new}}}^{SPr} \cdot K_{GW}^{SPu}$. The GW securely provisions $\left\{C_{CD_{\text{new}}}, K_{GW}^{SPu}, Z_{CD_{\text{new}},GW}^{\text{static}}\right\}$ into CD_{new}'s local non-volatile storage and stores the tuple $\left(ID_{CD_{\text{new}}}, C_{CD_{\text{new}}}, R_{CD_{\text{new}}}, K_{CD_{\text{new}}}^{SPu}\right)$ in the gateway's secure enrollment database.
3. Upon deployment, CD_{new} directly executes Phase 3 (Algorithm 2) to establish an authenticated, forward-secure session key $K_{CD_{\text{new}},GW}^{sh}$ with the GW.
4. The GW constructs the cluster enrollment list:

$$L_{\text{cluster}} = \left\{ \left(ID_i, R_i, Z_{i,CD_{\text{new}}}^{\text{static}}\right) \;\middle|\; CIoT_i \in \text{Cluster}(CD_{\text{new}}) \right\}.$$

The GW securely delivers L_{cluster} to CD$_{\text{new}}$, encrypted under $K_{CD_{\text{new}},GW}^{sh}$ using ChaCha20–Poly1305. The new CD is thus fully operational and can authenticate and aggregate data from its CIoT.

6.6.3. Compromised Device Revocation

When a CIoT or a CD is suspected of compromise, the GW initiates the revocation procedure to prevent further participation by the affected device.

1. Revocation Registration: The GW immediately adds the compromised device identifier ID_{revoked} to its centralized Revocation List ($\mathcal{RL}$).

2. Revocation of CIoT: The GW transmits an authenticated revocation command

$$\text{AEAD}\left(K^{sh}_{CD,GW}, n_{\text{rev}}, \{\text{“REVOKE-CIOT”}, ID_{\text{revoked}}\}, AD\right)$$

to the parent CD. The CD immediately evicts $\left(ID_{\text{revoked}}, R_{\text{revoked}}, Z_{\text{revoked}}, CD_{\text{static}}\right)$ from memory and drops all future initiation messages (M_1) matching ID_{revoked}.
3. Revoking CD: The GW marks ID_{CD} as revoked in the database and rejects any subsequent initiation messages (M_2) originating from that CD. The GW then commands the affected CIoT nodes, via broadcast beacon or during cluster maintenance, to associate with a neighboring fallback controller (CD_{alt}), securely disseminating new pairing shares $Z^{\text{static}}_{\text{IoT},CD_{\text{alt}}}$ to ensure seamless operational continuity.

7. Security Analysis

This section presents formal and informal security analyses of the proposed scheme to evaluate its effectiveness.

7.1. Formal Security Verification Using AVISPA

In this section, we present a formal security analysis of the proposed scheme using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [44,45]. AVISPA automatically verifies whether a cryptographic protocol achieves specific security goals, such as confidentiality and mutual authentication, and evaluates its resilience against passive and active threats, including eavesdropping, man-in-the-middle (MITM) attacks, and replay attacks. The tool relies on the High-Level Protocol Specification Language (HLPSL) to model protocols. HLPSL is a role-based language comprising two main role types: basic roles, which define the actions of individual participating entities, and composed roles, which describe the broader scenarios that instantiate these basic roles. Furthermore, HLPSL inherently supports the Dolev–Yao threat model [41], which assumes an adversary capable of intercepting, modifying, and replaying messages over public channels. During analysis, the HLPSL code was converted into Intermediate Format (IF) via the HLPSL2IF translator. This IF specification is then processed by AVISPA's four distinct back–ends—OFMC, CL–AtSe, SATMC, and TA4SP—to systematically verify the scheme's security properties and detect potential vulnerabilities.

1. **Specification of our scheme:** We implemented the FRACE-IoT scheme using HLPSL to formally model Algorithms 1 and 2. Figures 5 and 6 illustrate the detailed specifications of the basic roles of CIoT, CD, and GW. The composed roles, which comprise sessions, the environment, and the security goals, are presented in Figures 7 and 8.
    - Clarification of HLPSL Notation: Although the theoretical definitions and algorithms consistently use uppercase subscripts, such as CIoT, ID_{IoT}, n_{IoT}, and $K^{\text{sh}}_{\text{IoT,CD}}$, HLPSL syntax requires identifiers for constants, basic types, agent names, and security goals to begin with a lowercase letter, as illustrated by `ciot`, `id_iot`, `sec_k_iot_cd`, and `auth_n_iot`. The notational differences in Figures 5, 7 and 8 arise solely from HLPSL syntax requirements imposed by the AVISPA tool and do not alter the meaning of the corresponding mathematical notation defined in Table 1.
    - AVISPA scope and verification boundaries: The HLPSL modeling and AVISPA analysis are restricted to the key establishment stage (Algorithms 1 and 2) under the Dolev-Yao adversary model. The data transmission stage (Algorithm 3) is excluded from the AVISPA model, as its security relies on ChaCha20-Poly1305 AEAD and sequence-counter-based replay protection. Its confidentiality and ciphertext integrity are supported by the corresponding computational security

properties: IND-CPA and INT-CTXT. Accordingly, AVISPA formally verifies the specified confidentiality and mutual entity authentication goals for key establishment, while the security of data transmission and access control is addressed through the informal analyses in Section 7.3.

2. **Verification Results and Verified Security Properties:** Figure 9 presents the verification results generated by the OFMC and CL-AtSe back-ends. The "SAFE" verdict indicates that no reachable attack state violating the four formal security goals declared in the HLPSL environment specification was found within the explored state space.
   - Session Key Confidentiality (sec_k_iot_cd and sec_k_cd_gw): Both back-ends verified the declared secrecy goals within their respective search spaces. OFMC explored 9 nodes at a depth of 4, while CL-AtSe analyzed 24 states. The results indicate that an active Dolev–Yao intruder, with control over public communication channels and knowledge of public information, cannot derive or reconstruct the session keys $K^{\mathrm{sh}}_{\mathrm{IoT,CD}}$ and $K^{\mathrm{sh}}_{\mathrm{CD,GW}}$ under the assumptions encoded in the model.
   - Anti-Replay and Freshness (auth_n_iot and auth_n_cd): The authentication goals evaluate the correspondence between protocol executions and the fresh nonces n_{IoT} and n_{CD}. The absence of a counterexample indicates that the intruder cannot violate the specified authentication properties through replayed messages within the modeled protocol and explored state space.
   - Resistance to MITM and Masquerade Attacks: Since the Dolev–Yao intruder can intercept, modify, and inject messages, the absence of an execution trace violating `auth_n_iot` supports the protocol's resistance to the modeled impersonation and message-manipulation attacks. In particular, the results support the authenticity of the ephemeral public key K_x^{EPu} and the associated token τ_x, provided that these values are correctly bound to the authenticated protocol messages in the HLPSL specification.

   Therefore, the formal verification results support the session-key secrecy and authentication properties explicitly declared in the HLPSL model, as well as resistance to the modeled replay, impersonation, and man-in-the-middle attacks. These conclusions are restricted to the cryptographic abstractions, initial assumptions, and state space explored by the OFMC and CL-AtSe back-ends.

```
role role_CIoT(
    CIoT, CD    : agent,
    G           : text,
    hash        : hash_func
    Kcd_spu     : message,       % CD's static public key
    P_iot       : symmetric_key,  % Represents PUF(C_IoT)
    Z_iot_cd    : symmetric_key,  % Static shared secret Z_static
    ID_iot      : text,
    Frac_iot_cd : text,          % String constant "FRACE-IoT-CD"
    SND, RCV    : channel(dy))
played_by CIoT def=

    local
        State      : nat,
        K_iot_epr  : text,
        N_iot      : text,
        K_iot_epu  : message,
        DH_iot_es  : message,
        K_sh_iot_cd : message

    init State := 0

    transition
        1. State = 0 /\ RCV(start) =|>
           State' := 1
           /\ K_iot_epr' := new()
           /\ N_iot' := new()
           /\ K_iot_epu' := exp(G, K_iot_epr')
           /\ DH_iot_es' := exp(Kcd_spu, K_iot_epr')
           /\ K_sh_iot_cd' := Hash(DH_iot_es'.Z_iot_cd.Frac_iot_cd.N_iot')
           % Send M1 = (K_EPu, tau, n). ID_iot is included to facilitate DB lookup.
           /\ SND(ID_iot.K_iot_epu'.Hash(P_iot.N_iot').N_iot')
           /\ secret(K_sh_iot_cd', sec_k_iot_cd, {CIoT, CD})
           /\ witness(CIoT, CD, auth_n_iot, N_iot')

end role
```

```
role role_CD_Verif(
    CD, CIoT    : agent,
    G           : text,
    Hash        : hash_func
    Kcd_spr     : text,          % CD's static private key
    P_iot       : symmetric_key,  % Represents PUF(C_IoT) from DB
    Z_iot_cd    : symmetric_key,
    ID_iot      : text,
    Frac_iot_cd : text,
    SND, RCV    : channel(dy))
played_by CD def=

    local
        State      : nat,
        N_iot      : text,
        K_iot_epu  : message,
        DH_cd_es   : message,
        K_sh_iot_cd : message

    init State := 1

    transition
        % Pattern matching acts as the "assert" for the BLAKE3 tau tag
        1. State = 1 /\ RCV(ID_iot.K_iot_epu'.Hash(P_iot.N_iot').N_iot') =|>
           State' := 2
           /\ DH_cd_es' := exp(K_iot_epu', Kcd_spr)
           /\ K_sh_iot_cd' := Hash(DH_cd_es'.Z_iot_cd.Frac_iot_cd.N_iot')
           /\ request(CD, CIoT, auth_n_iot, N_iot')

end role
```

Figure 5. Role specifications of CIoT and CD in Algorithm 1 using HLPSL.

```
role role_CD_Init(
    CD, GW      : agent,
    G           : text,
    Hash        : hash_func,
    Kgw_spu     : message,        % GW's static public key
    P_cd        : symmetric_key,  % Represents PUF(C_CD)
    Z_cd_gw     : symmetric_key,  % Static shared secret Z_static
    ID_cd       : text,
    Frac_cd_gw  : text,           % String constant "FRACE-CD-GW"
    SND, RCV    : channel(dy)
played_by CD def=

    local
        State       : nat,
        K_cd_epr    : text,
        N_cd        : text,
        K_cd_epu    : message,
        DH_cd_es    : message,
        K_sh_cd_gw  : message

    init State := 0

    transition
        1. State = 0 /\ RCV(start) =|>
           State' := 1
           /\ K_cd_epr' := new()
           /\ N_cd' := new()
           /\ K_cd_epu' := exp(G, K_cd_epr')
           /\ DH_cd_es' := exp(Kgw_spu, K_cd_epr')
           /\ K_sh_cd_gw' := Hash(DH_cd_es'.Z_cd_gw.Frac_cd_gw.N_cd')
           % Send M2 = {K_EPu, tau, n, ID}
           /\ SND(K_cd_epu'.Hash(P_cd.N_cd').N_cd'.ID_cd)
           /\ secret(K_sh_cd_gw', sec_k_cd_gw, {CD, GW})
           /\ witness(CD, GW, auth_n_cd, N_cd')

end role
```

```
role role_GW(
    GW, CD      : agent,
    G           : text,
    Hash        : hash_func,
    Kgw_spr     : text,           % GW's static private key
    P_cd        : symmetric_key,  % Represents PUF(C_CD) from DB
    Z_cd_gw     : symmetric_key,
    ID_cd       : text,
    Frac_cd_gw  : text,
    SND, RCV    : channel(dy))
played_by GW def=

    local
        State       : nat,
        N_cd        : text,
        K_cd_epu    : message,
        DH_gw_es    : message,
        K_sh_cd_gw  : message

    init State := 1

    transition
        % Pattern matching acts as the "assert" for the BLAKE3 tau tag
        1. State = 1 /\ RCV(K_cd_epu'.Hash(P_cd.N_cd').N_cd'.ID_cd) =|>
           State' := 2
           /\ DH_gw_es' := exp(K_cd_epu', Kgw_spr)
           /\ K_sh_cd_gw' := Hash(DH_gw_es'.Z_cd_gw.Frac_cd_gw.N_cd')
           /\ request(GW, CD, auth_n_cd, N_cd')

end role
```

Figure 6. Role specifications of CD and GW in Algorithm 2 using HLPSL.

```
role session(
    CIoT, CD, GW : agent,
    G            : text,
    Hash         : hash_func,
    Kcd_spr      : text,
    Kgw_spr      : text,
    P_iot, P_cd  : symmetric_key,
    Z_iot_cd     : symmetric_key,
    Z_cd_gw      : symmetric_key,
    ID_iot, ID_cd: text,
    Frac_iot_cd  : text,
    Frac_cd_gw   : text)
def=
    local
        SND1, RCV1, SND2, RCV2, SND3, RCV3, SND4, RCV4 : channel(dy)

    composition
        % Compose Algorithm 1
        role_CIoT(CIoT, CD, G, Hash, exp(G, Kcd_spr), P_iot, Z_iot_cd, ID_iot, Frac_iot_cd, SND1, RCV1)
     /\ role_CD_Verif(CD, CIoT, G, Hash, Kcd_spr, P_iot, Z_iot_cd, ID_iot, Frac_iot_cd, SND2, RCV2)
        % Compose Algorithm 2
     /\ role_CD_Init(CD, GW, G, Hash, exp(G, Kgw_spr), P_cd, Z_cd_gw, ID_cd, Frac_cd_gw, SND3, RCV3)
     /\ role_GW(GW, CD, G, Hash, Kgw_spr, P_cd, Z_cd_gw, ID_cd, Frac_cd_gw, SND4, RCV4)

end role
```

Figure 7. Specification of session roles using HLPSL.

```
role environment()
def=
    const
        ciot, cd, gw        : agent,
        g                   : text,
        h                   : hash_func,
        kcd_spr, kgw_spr    : text,
        p_iot, p_cd         : symmetric_key,
        z_iot_cd, z_cd_gw   : symmetric_key,
        id_iot, id_cd       : text,
        frac_iot_cd         : text,
        frac_cd_gw          : text,
        sec_k_iot_cd        : protocol_id,
        sec_k_cd_gw         : protocol_id,
        auth_n_iot          : protocol_id,
        auth_n_cd           : protocol_id

    intruder_knowledge = {ciot, cd, gw, g, h, id_iot, id_cd, frac_iot_cd, frac_cd_gw, exp(g, kcd_spr), exp(g, kgw_spr)}

    composition
        session(ciot, cd, gw, g, h, kcd_spr, kgw_spr, p_iot, p_cd, z_iot_cd, z_cd_gw, id_iot, id_cd, frc_iot_cd,frac_cd_gw)

end role

goal
    % Confidentiality of the derived shared keys
    secrecy_of sec_k_iot_cd
    secrecy_of sec_k_cd_gw
    % Authentication of identities against nonces
    authentication_on auth_n_iot
    authentication_on auth_n_cd
end goal

environment()
```

Figure 8. Specification of environment roles using HLPSL.

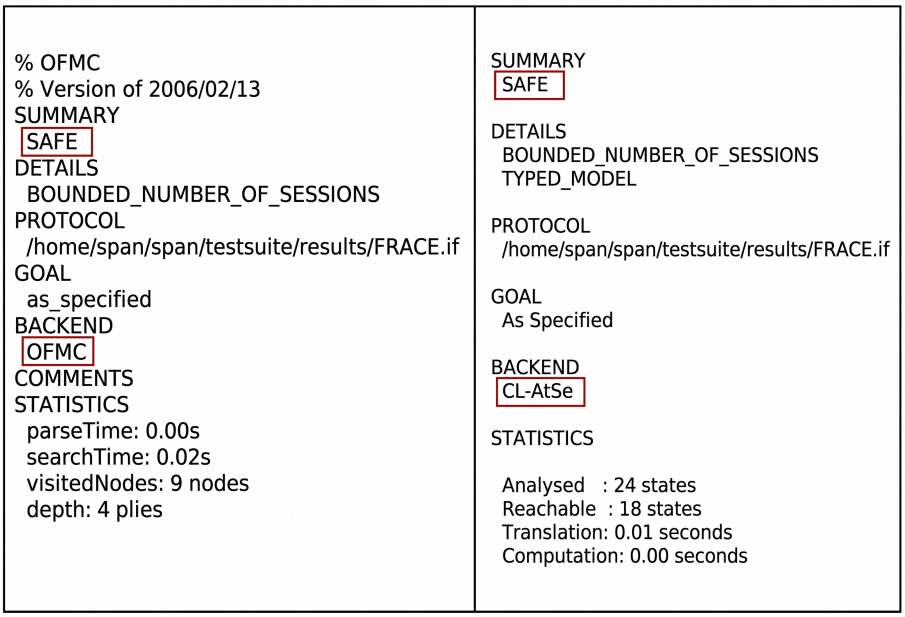


Figure 9. AVISPA simulation results for the proposed scheme under the OFMC (**left**) and CL-AtSe (**right**) backends. Red boxes highlight the security verification results (SAFE) and the backend engines used.

7.2. Formal Security Verification Using BAN Logic

To complement the automated symbolic model checking conducted via AVISPA under the Dolev–Yao model, we perform an analytical formal verification of the FRACE-IoT key establishment protocols using Burrows–Abadi–Needham (BAN) logic [46]. BAN logic is an axiomatic modal logic widely utilized to formally verify the authenticity, freshness, and mutual belief of shared session keys in cryptographic protocols.

7.2.1. Notations and Formal Inference Rules

Let P and Q denote protocol principals, X and Y denote cryptographic statements or data elements, and K denote an encryption key. The standard BAN logic notations are defined as follows:

- $P \models X$: Principal P believes that statement X is true.
- $P \triangleleft X$: Principal P sees (receives) message X.
- $P \sim X$: Principal P once said (transmitted) message X.
- $P \Rightarrow X$: Principal P has jurisdiction (authority) over statement X.
- $\#(X)$: Formula X is fresh (generated during the current protocol run).
- $P \overset{K}{\longleftrightarrow} Q$: Principals P and Q share secret key K, which is known only to them.
- $\langle X \rangle_Y$: Formula X is combined with secret parameter Y.
- $\{X\}_K$: Formula X is cryptographically encrypted or authenticated under key K.

The logical derivation relies on the following standard BAN logic inference rules:

1. **Message-Meaning Rule (MMR):**

$$\frac{P \models P \overset{K}{\longleftrightarrow} Q, \quad P \triangleleft \{X\}_K}{P \models Q \sim X} \tag{4}$$

2. **Nonce-Verification Rule (NVR):**

$$\frac{P \models \#(X), \quad P \models Q \sim X}{P \models Q \models X} \tag{5}$$

3. **Freshness Rule (FR):**

$$\frac{P \models \#(X)}{P \models \#(X, Y)} \tag{6}$$

4. **Jurisdiction Rule (JR):**

$$\frac{P \models Q \Rightarrow X, \quad P \models Q \models X}{P \models X} \tag{7}$$

5. **Belief Rule (BR):**

$$\frac{P \models (X, Y)}{P \models X}, \quad \frac{P \models Q \models (X, Y)}{P \models Q \models X} \tag{8}$$

7.2.2. Formal Security Goals

The primary objective of the FRACE-IoT key establishment protocol between the constrained sensor node (CIoT, denoted as I) and the controller device (CD, denoted as C) is to ensure mutual belief in the derived session key $K^{\text{sh}}_{\text{IoT,CD}}$. The formal security goals are formulated as:

- **Goal 1:** $C \models \left(I \overset{K^{\text{sh}}_{\text{IoT,CD}}}{\longleftrightarrow} C \right)$
- **Goal 2:** $C \models I \models \left(I \overset{K^{\text{sh}}_{\text{IoT,CD}}}{\longleftrightarrow} C \right)$
- **Goal 3:** $I \models \left(I \overset{K^{\text{sh}}_{\text{IoT,CD}}}{\longleftrightarrow} C \right)$
- **Goal 4:** $I \models C \models \left(I \overset{K^{\text{sh}}_{\text{IoT,CD}}}{\longleftrightarrow} C \right)$

Symmetrically, for Phase 3 between the controller C and the Smart Gateway G, the protocol must achieve:

- **Goal 5:** $G \models \left(C \overset{K^{\text{sh}}_{\text{CD,GW}}}{\longleftrightarrow} G \right)$
- **Goal 6:** $G \models C \models \left(C \overset{K^{\text{sh}}_{\text{CD,GW}}}{\longleftrightarrow} G \right)$
- **Goal 7:** $C \models \left(C \overset{K^{\text{sh}}_{\text{CD,GW}}}{\longleftrightarrow} G \right)$
- **Goal 8:** $C \models G \models \left(C \overset{K^{\text{sh}}_{\text{CD,GW}}}{\longleftrightarrow} G \right)$

7.2.3. Idealized Protocol Form

In BAN logic, transmitted messages are converted into idealized representations that convey only meaningful assertions. The initiation message M_1 in Algorithm 1 is formalized as:

$$M_1\ (I \to C): \quad \left(ID_{\text{IoT}},\ K^{\text{EPu}}_{\text{IoT}},\ n_{\text{IoT}},\ \langle I \overset{R_{\text{IoT}}}{\longleftrightarrow} C \rangle_{n_{\text{IoT}}} \right) \tag{9}$$

where $\langle I \overset{R_{\text{IoT}}}{\longleftrightarrow} C \rangle_{n_{\text{IoT}}}$ represents the PUF verification token $\tau_{\text{IoT}} = \text{BLAKE3}_{64}(R_{\text{IoT}} \parallel n_{\text{IoT}})$, binding the fresh nonce n_{IoT} to the hardware-bound secret response R_{IoT}.

7.2.4. Initial State Assumptions

Based on the secure Phase 1 enrollment and system parameters, the following initial assumptions are established:

$$A_1: \quad C \models \#(n_{\mathrm{IoT}}) \qquad \text{(Controller believes the session nonce is fresh)} \tag{10}$$

$$A_2: \quad I \models \#(K_{\mathrm{IoT}}^{\mathrm{EPr}}) \qquad \text{(CIoT believes its ephemeral private key is fresh)} \tag{11}$$

$$A_3: \quad C \models \left(I \overset{R_{\mathrm{IoT}}}{\longleftrightarrow} C\right) \qquad \text{(Controller holds authentic enrolled PUF response)} \tag{12}$$

$$A_4: \quad I \models \left(I \overset{Z_{\mathrm{IoT,CD}}^{\mathrm{static}}}{\longleftrightarrow} C\right) \qquad \text{(CIoT believes in the pre-computed static DH share)} \tag{13}$$

$$A_5: \quad C \models \left(I \overset{Z_{\mathrm{IoT,CD}}^{\mathrm{static}}}{\longleftrightarrow} C\right) \qquad \text{(Controller believes in the pre-computed static DH share)} \tag{14}$$

$$A_6: \quad C \models I \Rightarrow K_{\mathrm{IoT}}^{\mathrm{EPu}} \qquad \text{(Controller trusts CIoT over its ephemeral public share)} \tag{15}$$

$$A_7: \quad I \models C \Rightarrow K_{\mathrm{CD}}^{\mathrm{SPu}} \qquad \text{(CIoT trusts Controller's authentic static public key)} \tag{16}$$

7.2.5. Step-by-Step Formal Verification Derivation

The logical derivation proceeds through the following formal steps:

- **Step 1:** Upon receiving message M_1, the controller C observes the transmitted tuple:

$$C \triangleleft \left(ID_{\mathrm{IoT}}, K_{\mathrm{IoT}}^{\mathrm{EPu}}, n_{\mathrm{IoT}}, \langle I \overset{R_{\mathrm{IoT}}}{\longleftrightarrow} C\rangle_{n_{\mathrm{IoT}}}\right) \tag{17}$$

- **Step 2:** Using the Message-Meaning Rule (MMR) with the enrolled secret assumption A_3 (12) and the received token in (17):

$$\frac{C \models \left(I \overset{R_{\mathrm{IoT}}}{\longleftrightarrow} C\right), \quad C \triangleleft \langle I \overset{R_{\mathrm{IoT}}}{\longleftrightarrow} C\rangle_{n_{\mathrm{IoT}}}}{C \models I \sim \left(n_{\mathrm{IoT}}, K_{\mathrm{IoT}}^{\mathrm{EPu}}\right)} \tag{18}$$

  Equation (18) establishes that principal C believes that I originated the session nonce and the ephemeral public key.
- **Step 3:** By assumption A_1 (10), the controller believes n_{IoT} is fresh. Applying the Freshness Rule (FR):

$$\frac{C \models \#(n_{\mathrm{IoT}})}{C \models \#\left(n_{\mathrm{IoT}}, K_{\mathrm{IoT}}^{\mathrm{EPu}}\right)} \tag{19}$$

- **Step 4:** Applying the Nonce-Verification Rule (NVR) by combining freshness (19) with origin belief (18):

$$\frac{C \models \#\left(n_{\mathrm{IoT}}, K_{\mathrm{IoT}}^{\mathrm{EPu}}\right), \quad C \models I \sim \left(n_{\mathrm{IoT}}, K_{\mathrm{IoT}}^{\mathrm{EPu}}\right)}{C \models I \models \left(n_{\mathrm{IoT}}, K_{\mathrm{IoT}}^{\mathrm{EPu}}\right)} \tag{20}$$

  Applying the Belief Rule (BR) yields:

$$C \models I \models K_{\mathrm{IoT}}^{\mathrm{EPu}} \tag{21}$$

- **Step 5:** In Algorithm 1, the session key is computed via HKDF-BLAKE3 as:

$$K_{\text{IoT,CD}}^{\text{sh}} = \text{HKDF-BLAKE3}\left(DH_{\text{CD}}^{ES} \parallel Z_{\text{IoT,CD}}^{\text{static}}, \text{“FRACE-IOT-CD”}, n_{\text{IoT}}\right) \tag{22}$$

where $DHES_{\text{CD}} = K_{\text{CD}}^{\text{SPr}} \cdot K_{\text{IoT}}^{\text{EPu}}$. Because C holds $K_{\text{CD}}^{\text{SPr}}$ and shares $Z_{\text{IoT,CD}}^{\text{static}}$ with I (A_5), combining (21) with A_5 gives:

$$\frac{C \models I \models K_{\text{IoT}}^{\text{EPu}}, \quad C \models \left(I \overset{Z_{\text{IoT,CD}}^{\text{static}}}{\longleftrightarrow} C\right)}{C \models I \models \left(I \overset{K_{\text{IoT,CD}}^{\text{sh}}}{\longleftrightarrow} C\right)} \textbf{ (Goal 2)} \tag{23}$$

- **Step 6:** Applying the Jurisdiction Rule (JR) with assumption A_6 and (23):

$$\frac{C \models I \Rightarrow \left(I \overset{K_{\text{IoT,CD}}^{\text{sh}}}{\longleftrightarrow} C\right), \quad C \models I \models \left(I \overset{K_{\text{IoT,CD}}^{\text{sh}}}{\longleftrightarrow} C\right)}{C \models \left(I \overset{K_{\text{IoT,CD}}^{\text{sh}}}{\longleftrightarrow} C\right)} \textbf{ (Goal 1)} \tag{24}$$

- **Step 7:** On the initiator side, principal I generates fresh ephemeral scalar $K_{\text{IoT}}^{\text{EPr}}$ (A_2) and uses the authentic public key $K_{\text{CD}}^{\text{SPu}}$ (A_7) alongside the shared static component $Z_{\text{IoT,CD}}^{\text{static}}$ (A_4). Because the Diffie–Hellman operation satisfies the commutative property $K_{\text{IoT}}^{\text{EPr}} \cdot K_{\text{CD}}^{\text{SPu}} = K_{\text{CD}}^{\text{SPr}} \cdot K_{\text{IoT}}^{\text{EPu}}$, and only the genuine controller possessing $K_{\text{CD}}^{\text{SPr}}$ can compute $DHES_{\text{CD}}$, principal I directly deduces:

$$I \models \left(I \overset{K_{\text{IoT,CD}}^{\text{sh}}}{\longleftrightarrow} C\right) \quad \textbf{(Goal 3)} \tag{25}$$

- **Step 8:** During Phase 4 (Algorithm 3), the controller decrypts payload C_1 and verifies authentication tag $T_1 = \text{Poly1305}(K_{\text{IoT,CD}}^{\text{sh}}, \dots)$. The successful processing and subsequent payload aggregation under the matching key space provides implicit mutual verification to the session:

$$I \models C \models \left(I \overset{K_{\text{IoT,CD}}^{\text{sh}}}{\longleftrightarrow} C\right) \quad \textbf{(Goal 4)} \tag{26}$$

By symmetric derivation, substituting principals (C, G), ephemeral parameters $(K_{\text{CD}}^{\text{EPr}}, n_{\text{CD}})$, reference response R_{CD}, and pre-computed share $Z_{\text{CD,GW}}^{\text{static}}$ across Phase 3 (Algorithm 2), we successfully deduce:

$$G \models \left(C \overset{K_{\text{CD,GW}}^{\text{sh}}}{\longleftrightarrow} G\right) \quad \textbf{(Goal 5)} \tag{27}$$

$$G \models C \models \left(C \overset{K_{\text{CD,GW}}^{\text{sh}}}{\longleftrightarrow} G\right) \quad \textbf{(Goal 6)} \tag{28}$$

$$C \models \left(C \overset{K_{\text{CD,GW}}^{\text{sh}}}{\longleftrightarrow} G\right) \quad \textbf{(Goal 7)} \tag{29}$$

$$C \models G \models \left(C \overset{K_{\text{CD,GW}}^{\text{sh}}}{\longleftrightarrow} G\right) \quad \textbf{(Goal 8)} \tag{30}$$

The formal derivation confirms that both communicating entities at each architectural tier successfully achieve mutual belief in the freshness, authenticity, and exclusive possession of their shared session keys. This proves that FRACE-IoT is mathematically secure against masquerade, key-replication, and replay attacks.

7.3. Informal Security Analysis

In this subsection, we explain how the proposed scheme satisfies key security properties and evaluate its resilience against relevant attack vectors.

- Confidentiality: All data transmitted over the CIoT-to-CD and CD-to-GW links are encrypted using ChaCha20-Poly1305. ChaCha20 is IND-CPA-secure under the assumption that its keystream generator is a pseudorandom function, according to the RFC 8439 standard [33]. Because ephemeral key pairs are generated for each session and session keys are derived using HKDF-BLAKE3 with session-unique salts, the encryption performed in one session does not reveal any information about other sessions. An adversary who intercepts ciphertexts without knowledge of the session key cannot distinguish them from random bitstrings.
- Data Integrity and Authenticity: The Poly1305 tag included in each AEAD ciphertext provides INT-CTXT security with message forgery probability at most 2^{-128} [33]. The *AD* field, which includes the session identifier, device identifier, and sequence counter, is cryptographically linked to each ciphertext. Any modification to the ciphertext, associated data, or nonce causes tag verification to fail, and the receiving entity discards the message.
- Mutual Authentication: In the communication link between a CIoT device and a CD, a CIoT device is explicitly authenticated using the token τ_{IoT}, which can only be verified using the specific challenge C_{IoT}. Conversely, a CD is authenticated implicitly: only a legitimate CD holding $K_{\text{CD}}^{\text{SPr}}$ (derived from its unique PUF response) can derive the correct $K_{IoT,CD}^{sh}$. If an adversary attempting to impersonate the controller lacks the authentic $K_{\text{CD}}^{\text{SPr}}$, he will be unable to compute the valid shared session key $K_{IoT,CD}^{sh}$. This impersonation is inherently detected during Phase 4 upon the failure of the initial AEAD tag verification, thereby eliminating the need for an explicit challenge-response process. This identical dual-authentication logic applies symmetrically to the communication link between the CD and GW, relying on the token τ_{CD} and $K_{\text{GW}}^{\text{SPr}}$.
- MITM Attacks Resistance: MITM attacks require an adversary to impersonate one party to the other during the key establishment process (phase 2 and phase 3). In FRACE-IoT, any such attempt would require an adversary to (i) generate a valid PUF token τ_x for the impersonated device, which is infeasible without physical access to that device, and (ii) compute a valid shared session key, which requires knowledge of the static private key $K_{\text{x}}^{\text{SPr}}$ of the impersonated party, which can only be derived via the PUF. Because both conditions are infeasible under the stated threat model, MITM attacks are ruled out.
- Replay Attacks Resistance and Cross-Session Isolation: Replay mitigation in FRACE-IoT operates across two distinct protocol layers, each governed by precise security boundaries:
    1. Key Establishment Layer (Messages M_1 and M_2) and Cache Boundaries: In single-pass 0-RTT architectures, replaying historical handshake messages requires careful analysis of cache horizons:
        - Intra-Window Protection via W_{nonce}: The receiving entity maintains an in-memory sliding-window cache W_{nonce} of the most recent $W = 64$ session nonces. Replayed initiation messages residing within the active window are

dropped immediately (Algorithm 1, Step 10) prior to performing elliptic curve scalar multiplications, defeating DoS-oriented computational exhaustion.

- Post-Expiration Key Recurrence Analysis: If an adversary captures M_1 and replays it after n_{IoT} has been evicted from W_{nonce}, the verifier would re-derive the historical session key $K^{\text{sh}}_{\text{IoT,CD}}$ because all inputs to HKDF-BLAKE3 are identical. If unmitigated, replaying historical telemetry frames M_3 from that expired session could result in tag verification success since seq_{rx} reinitializes to 0.
- Architectural Mitigations for Cross-Session Immunity: To guarantee absolute cross-session isolation beyond the sliding window, FRACE-IoT provides two configurable hardening mechanisms:
  (a) Persistent Monotonic Epoch Counter: The 64-bit nonce is structured as $n_{\text{IoT}} = \langle epoch_{\text{IoT}}\,(32\text{ b}) \parallel rand_{32}\,(32\text{ b})\rangle$, where $epoch_{\text{IoT}}$ is incremented in non-volatile flash per session, and the controller stores the high-water mark $epoch_{\max}(\text{ID}_{\text{IoT}})$. Any message with $epoch \leq epoch_{\max}$ is permanently rejected, eliminating all cache-eviction vulnerabilities.
  (b) Receiver-Driven Entropy Injection (Stateful Key Evolution): In platforms where flash write-cycles must be avoided to conserve battery, the Controller mixes an internal monotonic counter σ_{CD} into the session key derivation:

$$K^{\text{sh}}_{\text{IoT,CD}} = \text{HKDF-BLAKE3}(DH^{\text{ES}} \parallel Z^{\text{static}} \parallel \sigma_{\text{CD}}, \text{“FRACE-IoT-CD”}, n_{\text{IoT}})$$

  Because σ_{CD} increments with every key negotiation, a replayed M_1 inevitably derives an uncorrelated session key $K^{\text{sh}''}$, causing historical data packets M_3 to fail Poly1305 authentication with probability $1 - 2^{-128}$.

2. Data Transmission Layer (Messages M_3 and M_4): During active telemetry exchange, payload replay is unconditionally prevented by binding each transmission to a strictly increasing sequence counter:

$$\text{AD} = \{\text{session_id}, seq_{\text{tx}}, \text{ID}_x\}$$

   The verifier enforces the invariant $seq_{\text{tx}} > seq_{\text{rx}}$ prior to decryption. Any duplicate, delayed, or reordered transmission results in immediate rejection at Step 5 of Algorithm 3. Furthermore, because seq_{tx} is authenticated within the Poly1305 associated data, an adversary cannot alter the counter without causing an authentication failure ($P_{\text{forgery}} \leq 2^{-128}$).

- Forward Secrecy and Asymmetric Compromise Resilience: In cryptographic protocol design, unidirectional single-pass (0-RTT) architectures exhibit distinct forward secrecy properties compared to interactive two-pass (1-RTT) exchanges. Because the responder does not transmit an online ephemeral public key, FRACE-IoT achieves Sender-Forward Secrecy rather than unconstrained bilateral PFS:

1. Resilience Against Initiator Static Compromise: If an adversary compromises the initiator's long-term static private key $K^{\text{SPr}}_{\text{IoT}}$ or static share $Z^{\text{static}}_{\text{IoT,CD}}$ at time $t > t_{\text{session}}$, the adversary remains computationally incapable of recovering past session keys $K^{\text{sh}}_{\text{IoT,CD}}$. Deriving $K^{\text{sh}}_{\text{IoT,CD}}$ requires the ephemeral-static Diffie–Hellman secret $DH^{\text{ES}}_{\text{IoT}} = K^{\text{EPr}}_{\text{IoT}} \cdot K^{\text{SPu}}_{\text{CD}}$. Because the initiator's ephemeral scalar $K^{\text{EPr}}_{\text{IoT}}$ is freshly sampled at each session and securely expunged from volatile RAM

immediately following transmission (Step 7, Algorithm 1), computing DH^{ES} from the intercepted public transcript ($K_{\mathrm{IoT}}^{\mathrm{EPu}}$, $K_{\mathrm{CD}}^{\mathrm{SPu}}$) reduces directly to the computational intractability of the ECDLP over Curve25519, providing 128-bit security.

2. Theoretical Boundaries of Single-Pass Responder Compromise: Conversely, as formally proven in the literature for 0-RTT key establishment (such as TLS 1.3, RFC 8446), if an adversary compromises the static private key of the responder $K_{\mathrm{CD}}^{\mathrm{SPr}}$, the adversary can use the publicly transmitted ephemeral public key $K_{\mathrm{IoT}}^{\mathrm{EPu}}$ to retroactively compute $DH_{\mathrm{CD}}^{\mathrm{ES}} = K_{\mathrm{CD}}^{\mathrm{SPr}} \cdot K_{\mathrm{IoT}}^{\mathrm{EPu}}$, thereby recovering past session keys. This represents an unavoidable fundamental trade-off of eliminating interactive round-trip handshakes to conserve transceiver energy.
3. Hardware-Enforced Mitigation via SRAM-PUF: FRACE-IoT mitigates this theoretical limitation by ensuring that static private keys are never stored in non-volatile memory on any device. As detailed in Section 6.1, $K_{\mathrm{CD}}^{\mathrm{SPr}}$ is dynamically reconstructed on-the-fly from uninitialized SRAM startup patterns only when actively required, and is purged from RAM immediately after session key derivation. Therefore, an adversary who physically captures a controller or sensor node cannot extract static private keys through non-volatile flash dumps, firmware extraction, or side-channel readout, effectively eliminating the static key leakage assumption in physical operating environments.

- Brute-Force Attack Resistance: FRACE-IoT provides resistance to brute-force attacks across three independent security layers. In the key agreement layer, the compromise of the session key would require solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) over Curve25519. This corresponds to a 128-bit security level, requiring approximately 2^{128} operations, which is beyond the practical computational feasibility. At the data encryption layer, ChaCha20 offers a key space of 2^{256} , thereby rendering exhaustive key search attacks computationally infeasible. At the authentication layer, the probability of successfully forging a Poly1305 tag without knowledge of the valid session key is bounded by at most 2^{-128} per attempt. Collectively, these security bounds meet or exceed the 2^{128} computational threshold defined in the threat model. Moreover, because session keys are strictly single-use and discarded immediately after use, any hypothetical brute-force compromise of a single session remains fully isolated and does not reveal information that could endanger other sessions.
- Atomic authenticated encryption: FRACE-IoT achieves confidentiality and integrity by employing the ChaCha20-Poly1305 authenticated encryption with associated data (AEAD) construction as a single atomic cryptographic primitive. During each data transmission, the ciphertext C and authentication tag T are generated simultaneously as $(C, T) = \mathrm{AEAD}(K_{sh}, n_{tx}, m, AD)$, where the AD includes the session identifier, sequence counter, and device identity, all of which are cryptographically authenticated without being encrypted. By integrating encryption and authentication into a single operation, FRACE-IoT eliminates implementation-ordering ambiguities inherent in multi-stage authenticated encryption constructions and ensures that ciphertext is processed only after successful authentication. Furthermore, because the authentication tag is cryptographically bound to both the ciphertext and the associated data for every invocation, any attempt to replay, reorder, modify, or reuse a ciphertext under a different session context results in authentication failure with negligible probability of successful forgery, thereby providing unified guarantees of message authenticity and session-context integrity.

- Resistance to State Desynchronization vs. Transient Packet Loss: In cryptographic protocols, a state desynchronization attack occurs when an adversary exploits message delivery failure to force communicating entities into divergent, irreconcilable internal states (such as rolling key counters or ratcheted pre-shared keys) resulting in permanent communication lockout (as demonstrated in our cryptanalysis of Al-Hejri et al. in Section 5.1). FRACE-IoT eliminates this vulnerability through its stateless single-pass architecture:
  1. Analysis of M_1 Interception: Consider an adversary who intercepts, alters, or drops the key initiation message $M_1 = \{ID_{\text{IoT}}, K_{\text{IoT}}^{\text{EPu}}, \tau_{\text{IoT}}, n_{\text{IoT}}\}$. Because the controller has not yet received M_1, its internal state remains completely unaffected. On the CIoT side, the ephemeral secret $K_{\text{IoT}}^{\text{EPr}}$ was generated strictly for that transient attempt. Neither entity has updated, overwritten, or discarded any long-term credentials or synchronized counters.
  2. Recovery Without Lockout: When the CIoT node initiates its subsequent transmission, it independently samples a fresh ephemeral scalar $K_{\text{IoT}}^{\text{EPr}\prime} \xleftarrow{\$} \mathbb{Z}_{2^{255}-19}$ and a fresh nonce n'_{IoT}, generating an entirely new message M'_1. Upon receiving M'_1, the controller reconstructs the identical static Diffie–Hellman (DH) secret, derives the new session key $K_{\text{IoT,CD}}^{\text{sh}\prime}$ and successfully decrypts subsequent payloads. Consequently, dropping or intercepting M_1 constitutes ordinary transient packet erasure inherent to any wireless medium and cannot induce cryptographic state desynchronization or persistent protocol failure.
- Hardware-Level Cloning Resistance: At the CIoT-CD communication, a cloned CIoT device operating on different hardware is unable to reproduce $\text{PUF}(C_{\text{IoT}})$, and consequently cannot compute a valid τ_{IoT}. Therefore, it is rejected in Step 10 of Algorithm 1. Similarly, at the CD-GW communication, a cloned CD cannot reproduce $\text{PUF}(C_{\text{CD}})$, cannot generate a valid τ_{CD}, and is rejected in Step 9 of Algorithm 2. In both scenarios, the hardware-level security guarantee remains intact even if an adversary fully extracts the device's non-volatile storage, because no stored data suffices to reproduce PUF responses on different physical hardware. This establishes uniform hardware-level resistance to cloning.

Table 2 provides a comparison of the security features of the proposed FRACE-IoT and the existing relevant schemes.

Table 2. Detailed comparison of the security properties of existing IoT authentication schemes and the proposed FRACE-IoT scheme.

Schemes	Security Objectives				Attack Resistance			Hardware	Formal
	Confidentiality	Data Integrity	Mutual Authentication	Man-in-the-Middle	Sender-Forward Secrecy	Replay Attacks	Brute-Force Attacks	PUF/Cloning Resistance	Formal Verification
Ranjan et al. (2024) [15]	✓	✓	✓	✓	∼	✓	∼	✗	✗
Chen et al. (2025) [12]	∼	✓	∼	✓	✓	✓	✓	✗	✓
Pu et al. (2026) [8]	∼	∼	✓	✓	✗	✓	✗	✗	✓
Peivandizadeh et al. (2024) [3]	✓	✓	✓	✓	✓	✓	✓	∼	✓
Gui et al. (2025) [19]	✓	✓	✓	✓	✓	✓	✓	∼	✓
Byun et al. (2026) [20]	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cherbal (2024) [6]	✓	✓	✓	✓	✓	✓	✓	∼	✓
Hang et al. (2025) [26]	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sudhakar et al. (2025) [25]	∼	✓	✓	✓	✓	✓	✓	∼	✗
Al-Hejri et al. (2024) [1]	✓	∼	✓	✓	✗	∼	✗	✓	✗
FRACE-IoT	✓	✓	✓	✓	✓	✓	✓	✓	✓

Notation: ✓ denotes that the property is achieved, ✗ denotes that it is not achieved, and ∼ denotes that it is partially achieved. *Confidentiality:* protection against unauthorized disclosure of transmitted information. *Data integrity:* protection against unauthorized modification of messages. *Mutual authentication:* verification of the identities of both communicating entities. *Man-in-the-middle resistance:* protection against an adversary intercepting or manipulating communication between legitimate parties. *Sender-Forward Secrecy:* Past session keys remain secure if the sender's long-term key is compromised. *Replay resistance:* prevention of the reuse of previously captured messages to impersonate a legitimate entity. *Brute-force resistance:* resistance to exhaustive guessing of secret credentials. *PUF/cloning resistance:* protection against device impersonation or physical cloning, including through the use of physical unclonable functions (PUFs). *Formal verification:* application of a formal verification method to assess protocol security.

8. Performance Analysis

This section presents a comprehensive performance evaluation of the proposed FRACE-IoT scheme compared to the scheme of Al-Hejri et al. [1] and other related approaches [3,6,25,26]. The assessment considers three main performance metrics: computational cost, communication overhead, and energy consumption during the mutual authentication, session key establishment, and secure data transmission phases. The results demonstrate that FRACE-IoT provides improved efficiency by reducing computational and communication overheads while achieving lower energy consumption than existing schemes.

8.1. Computational Cost

Since the gateway is considered a powerful device with sufficient computational capabilities, the computational analysis focuses on the overhead introduced in the constrained IoT (CIoT) and controller devices. The execution cost of the proposed scheme is evaluated and compared with that of the existing selected approaches considering the cryptographic operations performed during the phases of mutual authentication, establishing session keys and ensuring data transmission, as summarized in Table 3. The experimental measurements were performed on the FIT IoT-LAB Open Experimental IoT Testbed [47], which offers a realistic large-scale environment for evaluating low-power wireless IoT systems using various embedded devices and mobile robotic platforms. To ensure a fair comparison, the execution times of the cryptographic primitives involved in FRACE-IoT and the considered schemes were measured on two hardware platforms corresponding to the CIoT and controller nodes.

1. CIoT device: A FIT IoT-LAB M3 node [39] was selected to represent the constrained IoT device. This platform is equipped with an ARM Cortex-M3 microcontroller, 64 KB of RAM, and 256 KB of Flash memory, reflecting the limited computational resources typically available in IoT environments.
2. Controller device: A Nordic nRF52840DK board [40] was used to represent the controller device. Integrates an ARM Cortex-M4 processor, 256 KB of RAM, and 1 MB of Flash memory, providing higher computational capabilities than the CIoT device.

Table 4 summarizes the execution times of the cryptographic primitives required by FRACE-IoT and the corresponding operations used in relevant existing schemes. The results, expressed in milliseconds, were used to calculate the overall computational overhead of each scheme during the considered security phases.

According to Table 3, the proposed FRACE-IoT scheme demonstrates a computational time of 187.92 ms, which is the lowest compared to the scheme of Al-Hejri et al. [1] and other existing relevant schemes [3,6,25,26]. It is important to emphasize that the computational efficiency of FRACE-IoT is not merely an artifact of selecting modern primitives but stems directly from its architectural protocol design. Specifically, conventional IoT schemes [1,3,6,25,26] require multi-pass interactive handshakes that oblige resource-constrained nodes to execute multiple real-time scalar multiplications and maintain active radio-listening states across message rounds. In contrast, FRACE-IoT architecturally decouples static identity verification from session dynamism: by precomputing the static DH share Z^{static} during offline enrollment, the sensor node is required to perform only a single scalar multiplication ($K_{\text{IoT}}^{\text{EPr}} \cdot G$) and one point multiplication ($K_{\text{IoT}}^{\text{EPr}} \cdot K_{\text{CD}}^{\text{SPu}}$) during online execution. Furthermore, replacing decoupled two-pass encryption and hashing routines with the integrated ChaCha20-Poly1305 AEAD construction reduces payload-processing passes from two to one across all transmitted frames. In addition, two optimization factors were adopted in the FRACE-IoT scheme. The first optimization factor is the reduction in the cost of elliptic curve scalar multiplication (EC-SM), which is achieved by adopting

Curve25519. Compared with elliptic curves commonly used in existing related schemes, such as NIST P-256, NIST P-384, and BN254 [3,6,25,26], Curve25519 provides more efficient scalar multiplication, particularly on resource-constrained IoT platforms, while maintaining an equivalent 128-bit security level. This efficiency results from its optimized curve structure, use of the Montgomery form, and availability of fast and secure implementations based on the Montgomery ladder algorithm [48]. The second optimization factor was the adoption of BLAKE3 instead of SHA-256 for all cryptographic hashing operations, including PUF token generation and session key derivation through HKDF-BLAKE3. Because of its highly parallel and optimized design, BLAKE3 provides approximately twice the hashing throughput of SHA-256 on ARM Cortex-M3 / M4 platforms (as shown in Table 4), while maintaining a comparable 128-bit security level. Consequently, the computational overhead associated with hash-based operations is further reduced without compromising the security of the proposed scheme.

Regarding the cost of hardware-enforced private key security, regenerating K_d^{SPr} from the PUF response at runtime requires a single execution of HKDF-BLAKE3 (T_k), which consumes only 0.75 ms on the nRF52840DK controller node. This sub-millisecond cost accounts for less than 0.8% of the controller's active operational budget, representing an advantageous trade-off that entirely eliminates the attack vector of long-term credential extraction from physical flash storage.

Table 3. Comparison of computation costs (one session).

Scheme	CIoT	CD	Total	Time (ms)
Peivandizadeh et al. (2024) [3]	$3T_n + 10T_h + 2T_{sym}$	$3T_n + 10T_h + 2T_{sym}$	$6T_n + 20T_h + 4T_{sym}$	1349.92
Sudhakar et al. (2025) [25]	$T_n + 5T_h$	$T_n + 9T_h$	$2T_n + 14T_h$	462.21
Shang et al. (2025) [26]	$2T_r + 4T_h$	$T_r + 6T_h$	$3T_r + 10T_h$	3027.4
Cherbal (2024) [6]	$2T_h$	$2T_n + 4T_h$	$2T_n + 6T_h$	368.34
Al-Hejri et al. (2024) [1]	$2T_m + T_h$	$2T_m + 3T_h$	$4T_m + 4T_h$	692.81
FRACE-IoT	$2T_c + T_s + T_k + T_a$	$3T_c + 2T_s + 2T_k + T_a$	$5T_c + 3T_s + 3T_k + 2T_a$	187.92

Table 4. Time required to compute different cryptographic primitives (ms).

Symbol	Operation	CIoT	CD
T_c	Curve25519 scalar mult	44	31
T_m	P-256 scalar mult	183	160
T_n	BN254 scalar mult	259	179
T_r	P-384 scalar mult	1167	676
T_h	SHA-256 hash	1.89	1.64
T_s	BLAKE3 hash	0.76	0.68
T_k	HKDF-BLAKE3	0.84	0.75
T_a	ChaCha20-Poly1305 AEAD	1.25	1.21
T_{sym}	Symmetric encryption/decryption	0.17	0.14

8.2. Communication Cost

During the key establishment and secure data transmission phases, the proposed FRACE-IoT scheme exchanges four messages, denoted M_1, M_2, M_3, and M_4, across the three communicating tiers, including CIoT, CD, and GW. To provide full transparency, Table 5 details the exact numerical values, cryptographic primitives, and bit/byte sizes of all parameters constituting these transmissions.

Table 5. Cryptographic parameters and communication overhead breakdown in FRACE-IoT.

Parameter	Description	Primitive / Standard	Size
ID_x	Entity Unique Identifier	Fixed-length bitstring	32 bits [4 B]
K_x^{EPu}	Ephemeral Public Key	Curve25519 (X25519 point)	256 bits [32 B]
τ_x	PUF Authentication Token	BLAKE3 truncated digest	64 bits [8 B]
n_x	Handshake Fresh Nonce	TRNG / Cryptographic PRNG	64 bits [8 B]
n_{tx}, n'_{tx}	AEAD Transmission Nonce	RFC 8439 standard	96 bits [12 B]
T_1, T_2	AEAD Authentication Tag	Poly1305 MAC tag	128 bits [16 B]
seq, seq'	Monotonic Sequence Counter	Unsigned integer counter	32 bits [4 B]
Scheme Message Bit-Length Breakdown			
Message	Transmission Link	Constituent Fields	Message Size
M_1	CIoT $\rightarrow$ CD	$\{ID_{IoT}, K_{IoT}^{EPu}, \tau_{IoT}, n_{IoT}\}$	$32 + 256 + 64 + 64 = 416$ bits (52 B)
M_2	CD $\rightarrow$ GW	$\{ID_{CD}, K_{CD}^{EPu}, \tau_{CD}, n_{CD}\}$	$32 + 256 + 64 + 64 = 416$ bits (52 B)
M_3	CIoT $\rightarrow$ CD	$\{n_{tx}, T_1, seq_{tx}\}$	$96 + 128 + 32 = 256$ bits (32 B)
M_4	CD $\rightarrow$ GW	$\{n'_{tx}, T_2, seq'_{tx}\}$	$96 + 128 + 32 = 256$ bits (32 B)
Total	Complete Scheme Execution ($M_1 + M_2 + M_3 + M_4$)		1344 bits (168 B)

As summarized in Table 5, the total communication overhead introduced by FRACE-IoT across a complete session is $416 + 416 + 256 + 256 = 1344$ bits (168 bytes). In the data transmission phase (M_3 and M_4), the ciphertext payload C is excluded from the overhead accounting because ChaCha20 is a stream cipher that guarantees length preservation ($|C| = |m|$), incurring zero ciphertext expansion. Because the physical sensing payload m must be transmitted regardless of the security mechanism in place, the reported 256 bits per link strictly reflects the additional cryptographic metadata (nonce, Poly1305 tag, and sequence number) required to guarantee confidentiality, authenticity, and replay immunity.

Table 6 presents a comparative evaluation of the communication overhead of FRACE-IoT against existing schemes. The results show that FRACE-IoT achieves a total communication cost of 1344 bits per session using four messages. This cost is equal to that of Sudhakar et al. [25] (1344 bits) and lower than those of Peivandizadeh et al. [3] (1600 bits), Shang et al. [26] (2496 bits), Cherbal [6] (2208 bits), and Al-Hejri et al. [1] (1440 bits). In particular, FRACE-IoT reduces the communication overhead by 6.67% compared with Al-Hejri et al. [1] and by 46.15% compared with Shang et al. [26]. These results demonstrate that FRACE-IoT achieves a competitive communication cost while requiring fewer messages than several of the compared schemes.

Table 6. Comparison of communication costs (one session).

Scheme	Number of Messages	Total (Bits)
Peivandizadeh et al. (2024) [3]	2	1600
Sudhakar et al. (2025) [25]	2	1344
Shang et al. (2025) [26]	4	2496
Cherbal (2024) [6]	8	2208
Al-Hejri et al. (2024) [1]	6	1440
FRACE-IoT	4	1344

8.3. Energy Consumption

The total energy consumption of each entity during a complete communication session is expressed as the sum of the computation and communication energy components: $E = E_\alpha + E_\beta$, where the computation energy is given by $E_\alpha = V_c \cdot I_c \cdot t$, and the communication energy is computed as $E_\beta = V_r \cdot I_{\text{tx/rx}} \cdot \frac{U}{dr}$, where V_c and V_r denote the supply voltages of the micro-controller and radio, respectively; I_c is the micro-controller current in active mode; $I_{\text{tx/rx}}$ is the radio current during transmission or reception; t is the cryptographic computation time (second); U is the size of the message (bits); and dr is the rate of communication data (bits/s). Table 7 presents the values of I_c, $I_{tx/rx}$, V_c, V_r and dr for the FIT IoT-LAB M3 and Nordic nRF52840DK boards used in the evaluation of energy consumption.

Table 7. Electrical parameters of the experimental hardware boards.

Parameter	Notation	IoT-LAB M3 [39]	nRF52840DK [40]
Supply voltage, MCU	V_c	3.3 V	3.0 V
Supply voltage, radio	V_r	3.3 V	3.0 V
Active current, MCU (72/64 MHz)	I_c	30.5 mA	3.3 mA
Radio transmit current	I_{tx}	14.0 mA (+3 dBm)	4.8 mA (0 dBm)
Radio receive current	I_{rx}	12.3 mA	4.6 mA
Data rate	dr	250 kbps (IEEE 802.15.4)	250 kbps (IEEE 802.15.4)

The proposed model considers one complete communication session, including key establishment (Phases 2 and 3) and data transmission (Phase 4), consistent with the assumptions described in Section 8.1.

Figures 10 and 11 compare the energy consumption of the proposed FRACE-IoT scheme with five recent schemes, considering both cryptographic computation and communication energy. For each scheme, the total energy was obtained by summing the energy consumed by the CIoT and the corresponding CD. As shown in Figure 10, FRACE-IoT achieved the lowest computational energy, requiring only 10.11 mJ in total (9.14 mJ for the CIoT device and 0.96 mJ for the CD). In comparison, the relevant schemes consume between 13.83 and 242.47 mJ, with the scheme [26] exhibiting the highest computational cost. This is because the computational energy is directly derived from the computation time. Since FRACE-IoT requires less computation time than the existing schemes, it consequently exhibits lower computational energy consumption.

As shown in Figure 11, FRACE-IoT achieves a total communication energy consumption of 199.99 mJ, comprising 124.19 mJ for the CIoT device and 75.80 mJ for the CD. Although this value is slightly higher than that of Al-Hejri et al. [1] (196.09 mJ), it remains substantially lower than those of the other schemes, whose total communication energy consumption ranges from 305.29 to 377.59 mJ. These results indicate that FRACE-IoT reduces communication energy consumption by approximately 34.4% to 47.0% compared with the three schemes exhibiting the highest energy consumption, while remaining competitive with Al-Hejri et al. [1]. Communication energy consumption is primarily influenced by the volume of transmitted and received data, as well as the energy characteristics of the underlying hardware.

Considering both cryptographic computation and communication, the total energy consumption of FRACE-IoT is obtained by summing the energy required for these two operations. Specifically, the total energy consumption is 10.11 mJ for cryptographic computation and 199.99 mJ for communication, resulting in an overall energy consumption of 210.10 mJ. This result highlights the importance of evaluating both components when assessing the energy efficiency of lightweight security schemes for resource-constrained IoT environments.

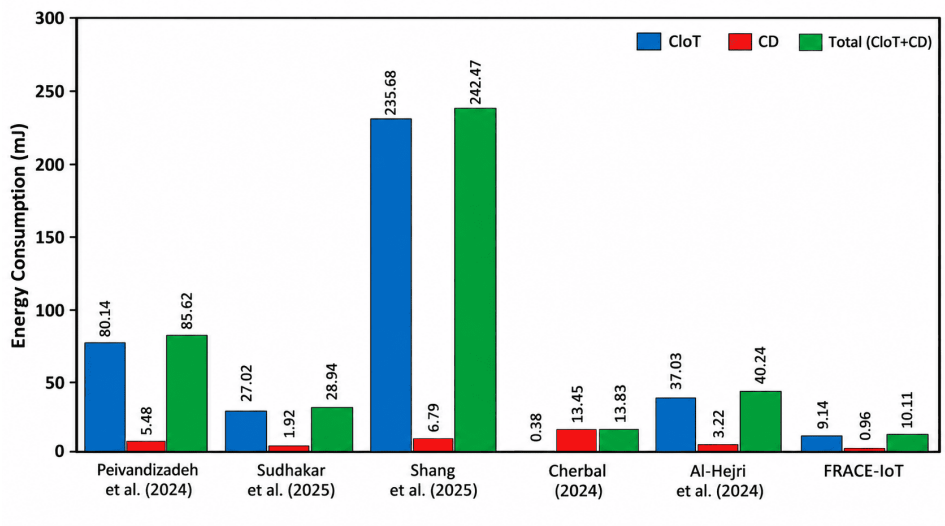


Figure 10. Energy required for cryptographic computation, comparing FRACE-IoT with Peivandizadeh et al. [3], Sudhakar et al. [25], Shang et al. [26], Cherbal [6], and Al-Hejri et al. [1].

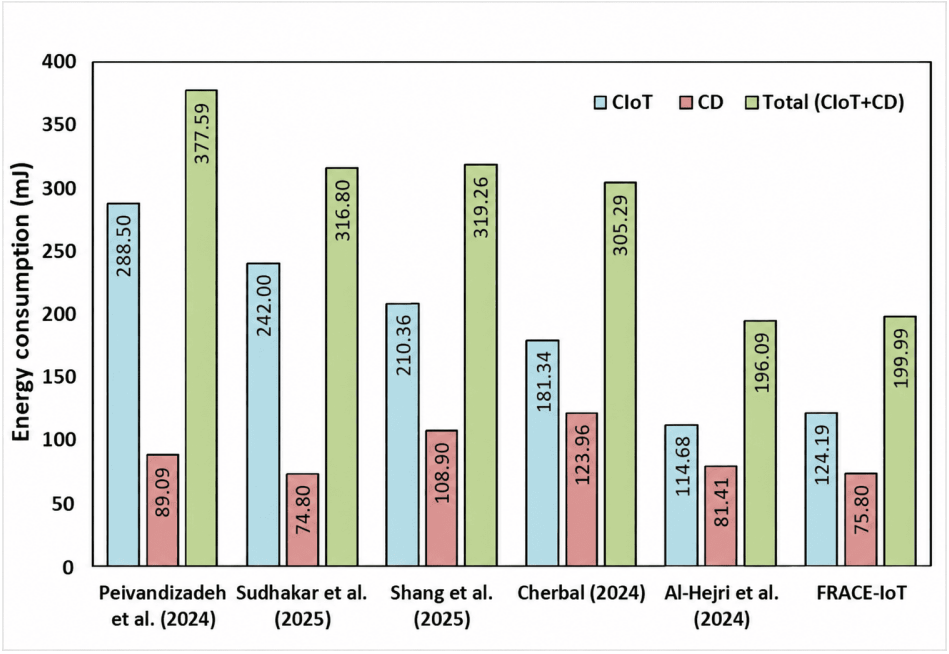


Figure 11. Energy consumption for communication, comparing FRACE-IoT with Peivandizadeh et al. [3], Sudhakar et al. [25], Shang et al. [26], Cherbal [6], and Al-Hejri et al. [1].

9. Conclusions and Future Work

This paper presented FRACE-IoT, a lightweight, authenticated encryption scheme for secure data transmission in heterogeneous IoT networks. The proposed scheme was motivated by a comprehensive cryptanalysis of the recent scheme by Al-Hejri et al., which revealed several important security shortcomings, including vulnerability to state desynchronization, the absence of sender-forward secrecy, reliance on non-atomic authenticated encryption, susceptibility to replay and device-cloning attacks, and the lack of formal security verification.

To address these limitations, FRACE-IoT integrates Curve25519-based key establishment, SRAM-based PUFs, HKDF-BLAKE3 key derivation, and ChaCha20-Poly1305 authenticated encryption within an integrated, desynchronization-resistant 0-RTT architecture. By eliminating state-dependent key updates and binding cryptographic identities

to hardware, the proposed design simultaneously achieves mutual authentication, confidentiality, integrity, sender-forward secrecy, resistance to replay, man-in-the-middle, state desynchronization, and hardware-level cloning attacks, while remaining suitable for resource-constrained IoT devices. Security analysis and formal verification of FRACE-IoT using the BAN logic and the AVISPA tool under the Dolev–Yao model confirm session key security, mutual belief, and robust resistance to major cyber-attacks. Experimental evaluation on embedded platforms FIT IoT-LAB M3 and Nordic nRF52840DK further showed that the proposed scheme achieves lower computational cost, reduced communication overhead, and improved energy efficiency compared to several recent lightweight IoT security schemes, demonstrating that stronger security guarantees can be achieved without sacrificing deployment efficiency.

Future work will focus on extending FRACE-IoT in several directions. First, we plan to develop a formal computational security proof in the random oracle model to complement the symbolic verification presented in this study. Second, we intend to investigate post-quantum variants of the protocol by replacing the elliptic-curve key establishment mechanism with lightweight quantum-resistant primitives while preserving the protocol's efficiency. Finally, we plan to evaluate FRACE–IoT in large–scale heterogeneous IoT deployments and emerging edge–enabled environments to further assess its scalability, resilience, and real–world performance under diverse network conditions.

Author Contributions: Conceptualization, F.M.; methodology, F.M.; software, F.M.; validation, F.M., A.N.B., A.Z. and P.L.; formal analysis, F.M.; investigation, F.M., A.N.B., A.Z. and P.L.; resources, F.M.; data curation, F.M.; writing—original draft preparation, F.M.; writing—review and editing, F.M., A.N.B., A.Z. and P.L.; visualization, F.M.; supervision, F.M.; project administration, F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Al-Hejri, I.; Azzedin, F.; Almuhammadi, S.; Eltoweissy, M. Lightweight secure and scalable scheme for data transmission in the internet of things. *Arab. J. Sci. Eng.* **2024**, *49*, 12919–12934. [CrossRef]
2. Ju, S.; Park, Y. Provably secure lightweight mutual authentication and key agreement scheme for cloud-based iot environments. *Sensors* **2023**, *23*, 9766. [CrossRef] [PubMed]
3. Peivandizadeh, A.; Y. Adarbah, H.; Molavi, B.; Mohajerzadeh, A.; H. Al-Badi, A. A secure key exchange and authentication scheme for securing communications in the Internet of Things environment. *Future Internet* **2024**, *16*, 357. [CrossRef]
4. Gangadharaiah, S.; Bhajantri, L.B. Secure data dissemination and routing in Internet of Things. *Int. J. Inf. Technol.* **2024**, 1–18. [CrossRef]
5. Attkan, A.; Ahlawat, P.; Ranga, V. A Novel Pyra-Cryptosystem Relying Secure Light-Weight Data Exchange, Peer-to-Peer Authentication, and Secure Communication Session Key Management Approach-Based Model for Internet of Things Network Security. *SN Comput. Sci.* **2025**, *6*, 84. [CrossRef]
6. Cherbal, S. 3s-alddt: A triple security scheme for authenticating lightweight devices and securing data transmission in internet of things. *Wirel. Pers. Commun.* **2024**, *135*, 727–762. [CrossRef]
7. Saif, S.; Halder, S.; Biswas, S. DLSEA-64: Dynamic lightweight symmetric encryption algorithm for secure data transmission in IoT based pervasive sensing applications. *J. Cyber Secur. Technol.* **2026**, *10*, 1–27. [CrossRef]
8. Pu, C.; Seol, J.; Park, N.; Korać, D. Authenticated Key Agreement Protocol for Device-to-Gateway Communication in IoT. In *Proceedings of the 2026 IEEE 23rd Consumer Communications & Networking Conference (CCNC)*; IEEE: New York, NY, USA, 2026; pp. 1–6. [CrossRef]
9. Zeng, W.; Xu, X.; Zhang, Q.; Shi, J.; Guan, Z.; Li, S.; Qin, Z. A secure and efficient distributed semantic communication system for heterogeneous Internet of Things. *IEEE Trans. Mob. Comput.* **2026**, *25*, 10370–10384. [CrossRef]

10. Chen, X.; Cheng, Q.; Yang, W.; Luo, X. An anonymous authentication and secure data transmission scheme for the Internet of Things based on blockchain. *Front. Comput. Sci.* **2024**, *18*, 183807. [CrossRef]
11. Chai, B.; Yu, J.; Yan, B.; Yu, Y.; Wang, S. BSCDA: Blockchain-based secure cross-domain data access scheme for Internet of Things. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 4006–4023. [CrossRef]
12. Chen, X.; Cheng, Q.; Chen, X.; Luo, X. A Blockchain-Based Cross-Domain Data Transmission Scheme for Industrial Internet of Things with Edge-Cloud Computing. *IEEE Trans. Serv. Comput.* **2025**, *18*, 2489–2502. [CrossRef]
13. Balraj, L.; Prasanth, A.; Sowndarya, K.D.; Kuntavai, T. A lightweight blockchain scheme for secure data communication in internet of things-enabled wireless sensor network. In *Proceedings of the 2024 International Conference on Smart Systems for Applications in Electrical Sciences (ICSSES)*; IEEE: New York, NY, USA, 2024, pp. 1–6. [CrossRef]
14. Fu, T.; Hao, S.; Chen, Q.; Yan, Z.; Liu, H.; Rezaeipanah, A. An energy-aware secure routing scheme in internet of things networks via two-way trust evaluation. *Pervasive Mob. Comput.* **2024**, *105*, 101995. [CrossRef]
15. Ranjan, A.K.; Kumar, P. APPS: Authentication-enabled privacy protection scheme for secure data transfer in Internet of Things. *Ad Hoc Netw.* **2024**, *164*, 103631. [CrossRef]
16. Lee, C.; Oh, M.; Kwon, D.; Park, Y.; Park, Y. PLAKA-MD: PUF-based lightweight authentication and key agreement scheme for medical devices in IoHT. *IEEE Internet Things J.* **2025**, *12*, 19338–19353. [CrossRef]
17. Lee, H.J.; Kook, S.; Kim, K.; Ryu, J.; Lee, Y.; Won, D. LAMT: Lightweight and Anonymous Authentication Scheme for Medical Internet of Things Services. *Sensors* **2025**, *25*, 821. [CrossRef] [PubMed]
18. Parameswarath, R.P.; Wang, C.; Sikdar, B. Quantum-Safe Authentication Protocol for IoT-Enabled Transportation Systems. *IEEE Trans. Ind. Appl.* **2025**, *62*, 729–741. [CrossRef]
19. Gui, S.; Zheng, M.; Liu, Y. Once: A unified secure and lightweight key agreement scheme for Internet of Things. *IEEE Internet Things J.* **2025**, *12*, 52143–52161. [CrossRef]
20. Byun, S.; Ryu, J.; Choi, Y.; Lee, H. Improved secure three-factor-based mutual authentication scheme for wireless sensor networks in internet of things environments. *Clust. Comput.* **2026**, *29*, 47. [CrossRef]
21. Kumar, V.; Ranjan, A.K.; Kumar, P. An EAP based Privacy Preserving Authentication and key agreement protocol for WLAN-IOT devices. *Peer-to-Peer Netw. Appl.* **2026**, *19*, 85. [CrossRef]
22. Narasimha Swamy, S.; Kumar, P.; Vijayalakshmi, M.; Dharani, A.; Raju, K.S. An Hybrid device authentication algorithm for edge-based IoT networks. *Automatika* **2026**, *67*, 22–36. [CrossRef]
23. Jhariya, M.K.; Dehalwar, V.; Bharti, J.; Pateriya, R.; Kumar, Y. Secure and efficient mutual authentication framework for IoT: A blockchain-integrated graph neural network approach. *Clust. Comput.* **2026**, *29*, 28. [CrossRef]
24. Verma, V.; Jha, V.K. Secure and energy-aware data transmission for IoT-WSNs with the help of cluster-based secure optimal routing. *Wirel. Pers. Commun.* **2024**, *134*, 1665–1686. [CrossRef]
25. Sudhakar, T.; Praveen, R.; Natarajan, V. An efficient ECC and fuzzy verifier based user authentication protocol for IoT enabled WSNs. *Sci. Rep.* **2025**, *15*, 9974. [CrossRef] [PubMed]
26. Shang, Y.; Chen, J.; Wang, S.; Zhang, Y.; Ma, K. A Secure and Lightweight ECC-Based Authentication Protocol for Wireless Medical Sensors Networks. *Sensors* **2025**, *25*, 6567. [CrossRef] [PubMed]
27. Du, H.; Wen, Q.; Zhang, S.; Gao, M. A new provably secure certificateless signature scheme for Internet of Things. *Ad Hoc Netw.* **2020**, *100*, 102074. [CrossRef]
28. Patil, H.K.; Szygenda, S.A. *Security for Wireless Sensor Networks using Identity-Based Cryptography*; Auerbach Publications: Boca Raton, FL, USA, 2012. [CrossRef]
29. Akhundov, H.; van der Sluis, E.; Hamdioui, S.; Taouil, M. Public-Key Based Authentication Architecture for IoT Devices Using PUF. In *Proceedings of the 6th International Conference on Computer Science, Engineering and Information Technology (CSEIT-2019)*; Aircc Publishing Corporation: Chennai, India, 2019; pp. 353–371. [CrossRef]
30. Gassend, B.; Clarke, D.; van Dijk, M.; Devadas, S. Silicon physical random functions. In Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA, 18–22 November 2002; CCS '02; pp. 148–160. [CrossRef]
31. Lounis, K.; Ding, S.H.; Zulkernine, M. D2D-MAP: A drone to drone authentication protocol using physical unclonable functions. *IEEE Trans. Veh. Technol.* **2022**, *72*, 5079–5093. [CrossRef]
32. Lounis, K.; Zulkernine, M. T2T-MAP: A PUF-based thing-to-thing mutual authentication protocol for IoT. *IEEE Access* **2021**, *9*, 137384–137405. [CrossRef]
33. Nir, Y.; Langley, A. ChaCha20 and Poly1305 for IETF Protocols. *RFC* **2018**, *8439*, 1–46. [CrossRef]
34. Serrano, R.; Duran, C.; Sarmiento, M.; Pham, C.; Hoang, T.T. ChaCha20-Poly1305 Authenticated Encryption with Additional Data for Transport Layer Security 1.3. *Cryptography* **2022**, *6*, 30. [CrossRef]
35. Ashur, T.; Dunkelman, O.; Luykx, A. Boosting authenticated encryption robustness with minimal modifications. In *Proceedings of the Annual International Cryptology Conference*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 3–33.

36. De Santis, F.; Schauer, A.; Sigl, G. ChaCha20-Poly1305 authenticated encryption for high-speed embedded IoT applications. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*; IEEE: New York, NY, USA, 2017; pp. 692–697. [CrossRef]
37. Krawczyk, H.; Eronen, P. *HMAC-Based Extract-and-Expand Key Derivation Function (HKDF)*; RFC 5869; IETF: Fremont, CA, USA, 2010. [CrossRef]
38. O'Connor, J.; Aumasson, J.P.; Neves, S.; Wilcox-O'Hearn, Z. One Function, Fast Everywhere. 2021. Available online: https://github.com/BLAKE3-team/BLAKE3-specs/blob/master/blake3.pdf (accessed on 6 July 2026).
39. FIT IoT–LAB Testbed. IoT–LAB M3 Board Documentation. Available online: https://www.iot-lab.info/docs/boards/iot-lab-m3/ (accessed on 6 July 2026).
40. FIT IoT–LAB Testbed. Nordic nRF52840DK Board Documentation. Available online: https://www.iot-lab.info/docs/boards/nordic-nrf52840dk/ (accessed on 6 July 2026).
41. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
42. Blake-Wilson, S.; Johnson, D.; Menezes, A. Key agreement protocols and their security analysis. In *Proceedings of the IMA International Conference on Cryptography and Coding*; Springer: Berlin/Heidelberg, Germany, 1997; pp. 30–45. [CrossRef]
43. Priya, L.; Kapilamithran, S. Exploiting AES encryption vulnerabilities through padding oracle attacks and generative AI techniques. In *Proceedings of the 2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*; IEEE: New York, NY, USA, 2024; pp. 682–689. [CrossRef]
44. Armando, A.; Basin, D.; Boichut, Y.; Chevalier, Y.; Compagna, L.; Cuéllar, J.; Drielsma, P.H.; Héam, P.C.; Kouchnarenko, O.; Mantovani, J.; et al. The AVISPA tool for the automated validation of internet security protocols and applications. In *Proceedings of the International Conference on Computer Aided Verification*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 281–285. [CrossRef]
45. Vigano, L. Automated security protocol analysis with the AVISPA tool. *Electron. Notes Theor. Comput. Sci.* **2006**, *155*, 61–86. [CrossRef]
46. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst. (TOCS)* **1990**, *8*, 18–36. [CrossRef]
47. Adjih, C.; Baccelli, E.; Fleury, E.; Harter, G.; Mitton, N.; Noel, T.; Pissard-Gibollet, R.; Saint-Marcel, F.; Schreiner, G.; Vandaele, J.; et al. FIT IoT-LAB: A large scale open experimental IoT testbed. In *Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*; IEEE: New York, NY, USA, 2015; pp. 459–464. [CrossRef]
48. Muchtadi-Alamsyah, I.; Tama, Y.B.W. Implementation of Elliptic Curve25519 in Cryptography. In *Theorizing STEM Education in the 21st Century*; Fomunyam, K.G., Ed.; IntechOpen: London, UK, 2019; Chapter 13. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.