INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research

International Journal of Internet Technology and Secured Transactions

ISSN online: 1748-5703 - ISSN print: 1748-569X
https://www.inderscience.com/ijitst

Enhancing IoT security in the post-quantum era: a hybrid approach to protection from impersonation attacks

Adel Hassan, Isam Ishaq, Jorge Munilla

DOI: 10.1504/IJITST.2026.10081120

Article History:

Received:	15 March 2026
Last revised:	27 May 2026
Accepted:	22 June 2026
Published online:	18 September 2026

Enhancing IoT security in the post-quantum era: a hybrid approach to protection from impersonation attacks

Adel Hassan

Faculty of Engineering and Information Technology,
Arab American University,
Jenin, Palestinian Territory
Email: hassan.adel@students.alquds.edu

Isam Ishaq*

Faculty of Engineering,
Al-Quds University,
Jerusalem, Palestinian Territory
Email: isam@itce.alquds.edu
*Corresponding author

Jorge Munilla

Faculty of Telecommunication Engineering School,
University of Málaga,
Málaga, Spain
Email: munilla@ic.uma.es

Abstract: The imminent advent of quantum computing threatens the long-term security of classical public-key cryptography, creating critical challenges for resource-constrained internet of things (IoT) environments. Hybrid cryptographic frameworks combining elliptic curve cryptography (ECC) with post-quantum cryptography (PQC) primitives have emerged as a transitional solution; however, their security properties remain insufficiently validated under semi-trusted infrastructure assumptions. This paper presents a comprehensive security evaluation of a hybrid ECC–KEM authentication and key agreement framework targeting IoT deployments. Using a dual-verification methodology that integrates formal symbolic analysis with the Tamarin-Prover and practical AI-assisted testing via the APSVer tool, we identify a critical impersonation vulnerability that enables a semi-trusted third party (TTP) to reconstruct PUF-based authentication material. To mitigate this flaw, we introduce a protocol enhancement incorporating ML-DSA post-quantum digital signatures and device-isolated key derivation. Formal verification confirms that the enhanced protocol achieves mutual authentication, session key secrecy, replay resistance, and effective isolation of the TTP. Experimental performance analysis demonstrates that the proposed security improvements introduce only a 12%–15% computational overhead, preserving feasibility for constrained IoT devices. The proposed framework provides a practical, quantum-resilient authentication architecture suitable for next-generation secure IoT deployments.

Keywords: post-quantum cryptography; PQC; hybrid ECC-KEM; key agreement; PUF-based authentication; Tamarin-Prover; APSVer; ML-DSA; ML-KEM.

Reference to this paper should be made as follows: Hassan, A., Ishaq, I. and Munilla, J. (2026) 'Enhancing IoT security in the post-quantum era: a hybrid approach to protection from impersonation attacks', *Int. J. Internet Technology and Secured Transactions*, Vol. 13, No. 9, pp.1–42.

Biographical notes: Adel Hassan received his BSc and MSc in Computer Engineering from the University of Jordan, and PhD in Computer Engineering from the University of Málaga, Spain. He is currently an Assistant Professor at the Faculty of Engineering and Information Technology at the Arab American University, Palestine. His research interests include network security, cryptography, and internet of things (IoT) protocols.

Isam Ishaq is the Dean of Dual Studies at Al-Quds University, Palestine. He received his PhD in Computer Science Engineering from Ghent University, Belgium, and Electrical Engineering degree from the Technical University of Berlin, Germany. He has held several academic and ICT leadership positions, including Assistant President for Research, Development and Innovation, and served as a member of the Palestinian Higher Council for Innovation and Excellence. His research focuses on the Internet of Things, wireless networks, and applied digital technologies, with more than 35 publications and over 700 citations.

Jorge Munilla is a Full Professor at the University of Málaga (Spain). He has been a visiting faculty member at the University of Graz (Austria), Florida State University (USA), the University of Wollongong (Australia), and the University of Kent (UK). He is the Director of the CATUC project, which fosters collaboration between the Engineering Schools of the University of Málaga and Chinese universities. His research interests include resilient cyber-physical systems, security for 5G/6G networks, and post-quantum security. He is the co-author of several book chapters and more than 50 papers published in leading international journals and conference proceedings.

1 Introduction

The rapid advancements in quantum computing pose an imminent threat to the security of global cyber systems. Known quantum algorithms, such as Shor's algorithm (1997) and Grover's algorithm (1996), are theoretically capable of breaking traditional public-key cryptographic algorithms like RSA and elliptic curve cryptography (ECC). Therefore, developing quantum-resistant cryptographic solutions, known as post-quantum cryptography (PQC) (NIST, 2024a), has become essential to protect the foundations of digital communications. In contrast to traditional cryptographic primitives, PQC primitives require significant computational and communication resources, which poses a major challenge in internet of things (IoT) environments characterised by limited processing power, memory, and bandwidth. This challenge is critical, as IoT devices are projected to exceed 30 billion units by 2027 (SonicWall, 2024), with IoT-focused cyberattacks increasing significantly in recent years (Kaspersky ICS CERT, 2023), highlighting the urgent need for secure yet lightweight protocols. The use of hybrid

protocols, which integrate traditional cryptography with post-quantum (PQ) techniques, has become a viable approach to overcome these issues while maintaining efficiency and security (Giron et al., 2023; Stadler et al., 2021). These protocols facilitate a flexible transition to quantum-resistant systems by allowing switching between cryptographic primitives based on the threat level. In this context, this study focuses on the pioneering protocol 'Flexible framework for multi-factor authentication (MFA) and bi-directional key agreement using ECC and KEM' (Braeken, 2025). This framework warrants a thorough analysis due to its unique architectural adaptability, which enables five distinct operational modes and serves as a critical case study for understanding the security interactions between classical and PQ primitives. Despite its high flexibility and broad adoption, initial evaluations have revealed potential security vulnerabilities. These flaws are particularly evident in scenarios involving a semi-trusted third party (TTP), where the asymmetric behaviour of PQ algorithms enables impersonation attacks. This leads to our primary research questions:

RQ1 How can a dual-verification methodology, combining symbolic (Tamarin-Prover) and practical AI-enhanced (APSVer) analysis, provide a more comprehensive security assessment for hybrid PQ-ECC IoT protocols?

RQ2 Does the enhanced protocol (Bernstein et al., 2022), which integrates ML-DSA signatures, effectively mitigate the TTP impersonation vulnerability identified in the original framework (Braeken, 2025)?

To address these questions, this study makes the following primary contributions:

- A unified verification pipeline: We introduce and apply a unified symbolic-practical verification pipeline, using Tamarin-Prover and the author-developed APSVer tool, to bridge the gap between theoretical logic and practical implementation.
- Vulnerability identification: We formally identify and confirm a critical TTP impersonation vulnerability in the original 'flexible framework' protocol (Braeken, 2025), demonstrating the precise design flaw that allows a TTP to forge security materials.
- Security validation: We provide a comprehensive security validation of the enhanced protocol (Bernstein et al., 2022), formally proving its effectiveness in mitigating the identified TTP attack by integrating ML-DSA signatures.

The remaining sections of the paper are organised as follows: Section 2 reviews related work. Section 3 evaluates the original protocol and its vulnerabilities. Section 4 describes the protocol security analysis and testing methodologies. The enhanced protocol is presented in Section 5. Section 6 presents the performance evaluation. Section 7 provides a detailed comparison and discussion of the analysis. Finally, Section 8 provides the discussion and conclusions.

2 Symmetric key-based systems

Symmetric cryptosystems are often prioritised for resource-constrained IoT deployments owing to their minimal computational requirements. For instance, Braeken (2020a) introduced two lightweight protocols utilising the Keccak hash function to facilitate

mutual authentication, anonymity, and non-likability, while ensuring resilience against transient secret leakage. While these frameworks reduce client-side overhead to a single hash operation, their reliance on a three-stage communication process makes them susceptible to denial of service (DoS) attacks.

Other notable attempts in this domain (Vijayakumar et al., 2020; Joshita and Arockiam, 2017; Lara et al., 2019; Mansoor et al., 2019), including works by Joshita and Arockiam (2017) and Lara et al. (2019), fail to address asynchronous attacks and the leakage of ephemeral data adequately. Specifically, the mechanism in Vijayakumar et al. (2020) compromises perfect forward secrecy (PFS) due to server-side storage constraints, whereas the approach in Mansoor et al. (2019) lacks robustness against desynchronisation. Although recent advancements in 5G-AKA protocols (Liu et al., 2021) have mitigated restart attacks, they still rely on retaining historical key values, which remains a significant drawback. Ultimately, the reliance on pre-shared keys in these systems fundamentally limits their ability to defend against semi-trusted third parties (TTP).

2.1 Distinct contributions of this work

The primary contributions of this research, which distinguish it from existing hybrid PQC-IoT frameworks, are summarised as follows:

- Vulnerability discovery: We expose a previously undocumented impersonation flaw within hybrid ECC-KEM authentication protocols. Our analysis demonstrates how a semi-trusted third party (TTP) can exploit cryptographic material to compromise PUF-based identities.
- Protocol hardening: To address this vulnerability, we propose an optimised PQ enhancement. By integrating ML-DSA signatures with ML-KEM encapsulation, the framework achieves robust resistance against both classical and quantum-era impersonation attacks.
- Hybrid verification pipeline: We introduce a dual-layer validation methodology. This approach synergises formal symbolic logic via the Tamarin-Prover with empirical, AI-driven stress testing using the APSVer tool, ensuring a higher degree of protocol assurance.
- Performance benchmarking: Through rigorous testing on ARM Cortex-M4 platforms, we provide quantitative evidence that the proposed PQC integration maintains a low computational footprint. The observed 12%–15% overhead confirms the protocol's practical viability for resource-constrained IoT endpoints.

2.2 Traditional public key-based systems

Despite the widespread adoption of public key infrastructure (PKI), existing literature (Stadler et al., 2021) highlights structural design flaws in several protocols (Zhou et al., 2021; Dolev and Yao, 1983). A primary concern is the server's role in generating key material for the client, which mirrors the TTP-related vulnerabilities observed in symmetric systems.

For example, while the architecture in Kocher et al. (1999) successfully integrates TTP protection, it fails to achieve PFS on the server side or provide immunity against

transient key exposure. Other frameworks (Istrate et al., 2025; Braeken, 2020b) provide more rigorous TTP definitions but at the cost of increased communication phases, thereby elevating the risk of distributed denial of service (DDoS) attacks.

In the IoT landscape, Chou et al. (Abdullah et al., 2025) proposed a lightweight ECC-based scheme featuring MFA. However, the inclusion of digital signatures imposes a computational burden that may be prohibitive for certain devices. Similarly, the integration of biometrics and physically unclonable functions (PUFs) in Bernstein and Lange (2017) enhances resilience against session-state leakage but does not address long-term threats. Crucially, all the aforementioned ECC-based systems remain fundamentally insecure against the looming threat posed by Shor's algorithm in the quantum era.

2.3 PQC-based systems

The necessity for long-term data protection in the wake of quantum advancements has shifted the focus toward quantum-resistant protocols. This field is primarily divided into two methodologies: the first leverages the inherent difficulty of solving lattice-based mathematical problems, such as the shortest vector problem (SVP) or learning with errors (LWE). Despite their theoretical strength, several protocols within this category (Román et al., 2024; Panda and Chattopadhyay, 2020) have proven vulnerable to side-channel leakage attacks, particularly when key pairs are reused. The second methodology employs quantum-secure key encapsulation mechanisms (KEMs) and digital signatures, with CRYSTALS-Kyber serving as a prominent example. For instance, a KEM-based adaptation of the 5G-AKA protocol was proposed in Kumari et al. (2020), though it incurred substantial communication overhead. Similarly, Ahmad et al. introduced a hybrid IoT framework integrating CRYSTALS-Kyber. However, optimising the trade-off between cryptographic robustness and operational efficiency remains a significant hurdle. Empirical data (Varanasi, 2025) indicate that algorithms such as Kyber can impose transmission costs up to 8 times higher than those of traditional ECC-based solutions.

2.4 Empirical benchmarking of PQC in IoT environments

Recent studies (2022–2025) have empirically evaluated the deployment of PQC algorithms in resource-constrained IoT environments (Khalid and Raza, 2024; Thompson and Zhao, 2025), highlighting measurable trade-offs in computational cost, memory footprint, and communication overhead. Several benchmarking studies have examined the integration of CRYSTALS-Kyber (ML-KEM) into lightweight IoT communication stacks such as MQTT and CoAP (Khalid and Raza, 2024), reporting increased handshake sizes (1.5–3.5 KB) and latency overheads of 20–45 ms on ARM Cortex-M4 platforms, while maintaining acceptable energy consumption profiles.

Furthermore, recent NIST (2024b) standards confirm that while PQC introduces measurable overhead, the security gains against quantum adversaries justify the trade-off in next-generation IoT infrastructures. Optimised implementations demonstrate that PQC deployment remains within acceptable bounds for constrained IoT gateways and edge nodes, as evidenced by the latest 2025 frameworks (Thompson and Zhao, 2025).

2.5 Research gap and contribution

A critical synthesis of current literature identifies two major deficiencies. Theoretically, while hybrid flexibility is desirable, existing frameworks often present a dichotomy of weaknesses: full quantum-resistant systems (Kumari et al., 2020) are frequently too resource-intensive for IoT hardware, whereas traditional (Istrate et al., 2025; Braeken, 2020b; Abdullah et al., 2025; Bernstein and Lange, 2017) and certain hybrid (Kumar et al., 2024; Schanck et al., 2016; Kampanakis and Sikeridis, 2019) models fail to provide adequate defence against semi-trusted third parties (TTP) or remain susceptible to Shor's algorithm. Empirically, there is a lack of rigorous examination of the performance-security trade-off using formal verification tools, leaving the practical efficiency of PQC primitives largely unverified. To bridge these gaps, this research focuses on the following primary objectives:

- *Security validation*: Employing the Tamarin-Prover to formally verify that the enhanced hybrid protocol ensures robust security, specifically proving its immunity to TTP impersonation attacks while maintaining session key (SK) secrecy and mutual authentication.
- *Dual-methodology assessment*: Implementing a comprehensive verification pipeline that integrates symbolic analysis (Tamarin-Prover) with practical AI-driven assessment (via the author-developed APSVer tool) to evaluate protocol security from both logical and implementation-oriented perspectives.
- *Performance benchmarking*: Quantifying the computational and communication demands of the enhanced protocol. The goal is to demonstrate that integrating ML-DSA signatures provides superior protection without overwhelming IoT resources, with a message overhead increase of less than 30%.

2.6 Formal verification of hybrid and PQ protocols

The integration of PQC introduces unprecedented architectural complexities, rendering manual security audits increasingly unreliable. Consequently, leveraging formal verification tools for symbolic analysis has emerged as a cornerstone of modern cryptographic research. Automated tools such as Tamarin-Prover and ProVerif are now standard for validating the security properties of emerging protocols. Recent literature underscores the indispensability of this formal approach. For instance, the analysis of the PQ extended Diffie-Hellman (PQXDH) protocol revealed subtle specification flaws that were detectable only by ProVerif and CryptoVerif. Similarly, formal methods have enhanced the security of MFA frameworks by automating vulnerability discovery across diverse threat models. Tamarin-Prover, in particular, has demonstrated superior capability in modelling intricate, stateful protocols, such as 5G-AKA, and in identifying resilience gaps that escaped manual inspection. While comparative studies have evaluated the relative strengths of Tamarin, AVISPA, and ProVerif, most existing research remains confined to established protocols or tool comparisons. A significant research gap persists in applying a dual-analysis methodology – integrating symbolic logic (Tamarin) with practical, AI-enhanced assessment (APSVer) – to rigorously validate novel PQ-ECC frameworks tailored for IoT authentication.

3 Original protocol and its vulnerabilities

This section provides a rigorous examination of the 'Flexible Framework for MFA and Bi-directional Key Agreement using ECC and KEM', originally introduced. This protocol was designed to offer an adaptive security mechanism for IoT environments by hybridising PQ primitives, such as CRYSTALS-Kyber, with traditional ECC. The framework is characterised by its operational versatility, allowing users to choose among three modes: ECC-only, Hybrid (ECC with KEM), or full quantum mode, depending on the specific security and resource constraints of the IoT application. However, our investigation – utilising both formal symbolic verification and practical AI-driven testing – exposed fundamental security deficiencies. These flaws compromise the integrity of the authentication and key agreement (AKA) phases, especially when a semi-trusted third party (TTP) is present. The subsequent subsections detail the protocol's architecture, cryptographic primitives, and the specific mechanics of the identified vulnerabilities.

3.1 Cryptographic primitives

Elliptic curve cryptography

ECC remains a cornerstone of traditional public-key infrastructure, leveraging the computational hardness of the elliptic curve discrete logarithm problem (ECDLP). In a finite field F_p, it is computationally infeasible to determine the scalar $a \in F_p$ from $A = aG$ within polynomial time, where G represents the curve's generator. Consequently, the private and public key pairs in this framework are denoted by d and Q, respectively, such that $Q = dG$.

Key encapsulation mechanism

A KEM is a fundamental cryptographic primitive that enables the secure distribution of symmetric keys using public-key cryptography. In PQ frameworks, the encapsulation process involves generating a random secret key k and its corresponding ciphertext c using the receiver's public key pk_B. Formally, this is represented as $(c, k) \leftarrow Encaps(pk_B)$, where k is internally generated and 'wrapped' within c. To retrieve the secret key, the receiver utilises their private key sk_B in the decapsulation phase: $k \leftarrow Decaps(sk_B, c)$. While the elliptic curve integrated encryption scheme (ECIES) is a classic example of this mechanism, it lacks quantum resilience. Currently, CRYSTALS-Kyber stands as the primary KEM selected in the NIST PQ standardisation process (Sarmah, 2026).

Other cryptographic operations

Traditional symmetric encryption and hash functions are considered resilient in the PQ era, provided their key and output lengths are appropriately doubled. Symmetric encryption of a message M using key K is denoted by $C = E_K(M)$, while its decryption is $M = D_K(C)$. Similarly, the hash operation on message M is represented by $H(M)$. A practical example used in this study is SHAKE128(M, d), an extendable-output function (XOF) in which d denotes the variable output size. Furthermore, $M1|M2$ signifies the

concatenation of messages, and $M1 \oplus M2$ denotes the bitwise exclusive-OR (XOR) operation (Segers et al., 2024).

Physical unclonable functions

Physical unclonable functions (PUFs) have emerged as a foundational security primitive (Braeken, 2018) for ensuring device authenticity in distributed environments. Their primary advantages include low computational complexity and minimal hardware overhead. This study assumes the integration of a robust PUF within the device's microcontroller, characterised by its inherent resistance to replication, forecasting, or cloning, even when identical manufacturing processes are employed. The unique response of a PUF is derived from stochastic physical variations in the integrated circuit (IC) microstructure. This mechanism is formally described by the equation $R = P(C)$, where an input C (challenge) produces a distinct output R (response), forming a challenge-response pair (CRP). The architecture and evaluation of such PUF constructions remain a focal point of contemporary research (Bos et al., 2018). For practical implementation in microcontrollers (notably the ARM-Cortex-M23 and M33 series), the ATSHA204A crypto-authentication module is recommended to provide a formidable defence against invasive and physical tampering (Harshey et al., 2021). During the configuration phase, a unique value generated by a true random number generator (TRNG) is stored in the module's write-only memory. The final response is subsequently computed as a cryptographic function of the challenge, a predefined secret, and this hardware-intrinsic random number. This architecture mitigates unauthorised access to the underlying secret; furthermore, in the event of a physical intrusion, changes in hardware state result in a different challenge-response outcome. A significant operational benefit of this specific mechanism is the elimination of complex error-correction schemes or helper data, as the PUF maintains high stability in data reproduction.

Biometrics

Multifactor authentication provides a robust security layer for devices requiring user interaction. Beyond traditional credentials, it integrates biometric modalities such as fingerprints, facial recognition, or iris patterns. Since biometric data is inherently noisy and often processed via hash-based techniques (NIST, 2024b), fuzzy extractors are employed to ensure reliable key derivation (Dodis et al., 2004). We denote the unique output α_u derived from a biometric template β as $\alpha_u = PU(\beta_u)$. The function P_U is fundamentally comprised of two stages: a probabilistic generation algorithm followed by a deterministic reproduction mechanism. Table 1 summarises the notations frequently utilised throughout this paper.

Table 1 Summary of notations

(sk_d, pk_d)	Post-quantum (PQ) private/public key pair of the IoT device
(d_d, Q_d)	The traditional ECC private/public key pair of the user device
(d_u, Q_u)	The traditional ECC private/public key pair of the user device
$(c, k) \leftarrow Encaps(pk_d)$	Encapsulation of secret key k in ciphertext c using pk_d
$k \leftarrow Decaps(sk_d, c)$	Decapsulation of secret key k in ciphertext c using sk_d

Table 1 Summary of notations (continued)

$E_K(.)$ / $D_K(.)$	Symmetric encr./decr. with key K
$H(.) = (h1, \ldots, hn)$	Hash with extendable outputs
$\|\|, \oplus$	String concatenation and bitwise XOR operation, respectively
$R = P(C)$	Response of PUF P on challenge C
$\alpha_u = P_u(\beta_u)$	Output of biometrics template β_u
(sk_u, pk_u)	User-specific ML-DSA algorithm private/public key pair
$S \leftarrow ML\text{-}DSA.Sign(sk_u, m)$	ML-DSA signature generation for message m using the private key.
$\{0, 1\} \leftarrow ML\text{-}DSA.Verify(pk_u, m, S)$	ML-DSA signature verification using the public key.
$(c, k) \leftarrow ML\text{-}KEM.Encaps(pk_d)$	ML-KEM key encapsulation for generating cipher-text c and shared key k.
$k \leftarrow ML\text{-}KEM.Decaps(sk_d, c)$	ML-KEM decapsulation to recover the shared key k using the private key.
r_i^*	Consolidated PUF reference response: $h(h(h(r_i, K_{TD}), Q_u), K_i)$
$K_{i1} \mid K_{i2}$	Shared key components for ECC and PQC layers, respectively.

3.2 System architecture and security model

The proposed architecture primarily comprises a user device and a general-purpose IoT device. To facilitate secure, authorised access for multiple users, a pre-installation phase is required to establish a cryptographic link between each user and the IoT node. Central to this scheme is a trusted third party (TTP), responsible for generating and distributing the initial secret key material. Specifically, the TTP provisions the IoT device with essential secrets and delivers user-specific key material upon verification of the access requirements set by the device owner. According to established security benchmarks (Braeken, 2022, 2023), a robust AKA framework for dual-device environments must ensure confidentiality, integrity, anonymity, untraceability, and PFS. Additionally, it must provide resilience against temporal session-state leakage, DoS attacks, semi-trusted TTP threats, online/offline password guessing, and lost-device scenarios. The evaluation of this framework is grounded in the following attacker model assumptions:

- *Network capabilities*: We adopt the Dolev-Yao adversary model (Cervesato, 2001), which assumes a potent attacker capable of both passive eavesdropping and active intervention. This includes the ability to intercept, delete, or modify messages transmitted across the communication channel.
- *Key compromise*: Adhering to the extended Canetti-Krawczyk (eCK) model (LaMacchia et al., 2007), we assume the adversary can compromise both ephemeral and long-term secrets without necessarily undermining the security of future or past SKs.
- *Hardware resilience*: Consistent with prior methodologies (Braeken, 2018), the IoT device integrates a robust PUF. We assume that secret keys derived from the PUF are tamper-evident and cannot be extracted without physically destroying the underlying hardware mechanism (Gope et al., 2018).

- *Multifactor security*: The system employs three distinct authentication factors: biometric identity, password knowledge, and physical possession of the device. Our model assumes that an attacker can compromise at most two of these three factors (Braeken, 2018).
- *Hybrid security assumption*: Following the requirements for hybrid cryptosystems (Kumar et al., 2024), we assume that while either ECC or PQ primitives might be vulnerable, the security of the overall scheme remains intact as long as neither is compromised simultaneously.

3.3 Adversary model and assumptions

The security of the proposed protocol is analysed under the *Dolev-Yao adversary model* (Dolev and Yao, 1983), which is the standard framework for formal cryptographic protocol analysis. The adversary is assumed to have full control over the communication channel, enabling the interception, modification, replay, and injection of protocol messages.

In addition to classical network-level capabilities, this model incorporates a *semi-trusted third party* (*TTP*). While the TTP is assumed to execute the protocol as specified (honest-but-curious), it is not trusted with long-term device secrets. It may attempt to infer sensitive keying material or impersonate legitimate IoT devices. Specifically, the adversary's capabilities include:

- Eavesdropping and replay of authentication messages.
- Active man-in-the-middle (MitM) attacks during the key agreement phase.
- Impersonation of legitimate IoT nodes by exploiting TTP-stored information.
- Attempted reconstruction of PUF-derived secrets via partial TTP compromise.

Assumptions: Hardware-level PUF unclonability is treated as a trusted primitive, and physical access to IoT devices by the adversary is excluded. Furthermore, DoS attacks are considered outside the scope of this work. This threat model reflects realistic PQ IoT environments where infrastructure components may be compromised, yet long-term resilience is mandatory.

3.3.1 Limitations: physical tampering and DoS attacks

The current threat model (Section 3.3) explicitly excludes physical access to IoT devices and DoS attacks. We briefly discuss their potential impact and possible mitigations below.

Physical tampering with PUFs

While PUFs are designed to be tamper-evident, sophisticated physical attacks (e.g., focused ion beam, side-channel analysis, decapsulation) could potentially extract PUF responses or bypass the PUF entirely. If an adversary gains physical possession of the device, they could attempt to:

- Reconstruct the secret KTD via invasive probing.
- Inject malicious responses by manipulating the PUF input voltage or temperature.
- Bypass the PUF by directly reading the response from memory if it is not properly protected.

Potential mitigations

- *Active shielding and glue logic sensors* that detect physical intrusion and trigger key zeroisation.
- *Rate-limiting* of PUF challenge-response attempts to prevent brute-force reconstruction.
- *PUF with obfuscation (PUF-Obf)* that combines PUF responses with a device-specific secret stored in tamper-resistant memory.
- *Encrypted PUF responses* using a key derived from a physically isolated secure element.

DoS attacks

An adversary could flood the IoT device with malformed authentication requests, exhausting computational resources (especially during PQ decapsulation and signature verification). Since the device performs expensive ML-KEM decapsulation (≈4.8 ms) and ML-DSA verification (≈2.6 ms) for each authentication request, a DoS attack could be highly effective.

Potential mitigations

- *Computational puzzles* (e.g., proof-of-work) that require the client to solve a moderately difficult puzzle before the server allocates resources.
- *Request rate limiting* based on source IP or device identity.
- *Whitelisting* of known user devices and pre-established session contexts.
- *Asynchronous authentication queues* with timeout-based request dropping.

These limitations and mitigations are acknowledged as important directions for future work, particularly for deployments in physically accessible or adversarial network environments.

3.4 Proposed framework

Initialisation, registration, authentication, and key agreement are the three stages that make up the framework. Both the user device and the IoT device carry out the initialisation process. The assistance of a TTP is necessary for the registration process. Lastly, the user device and the IoT device are the only parties involved in the authentication and key negotiation step.

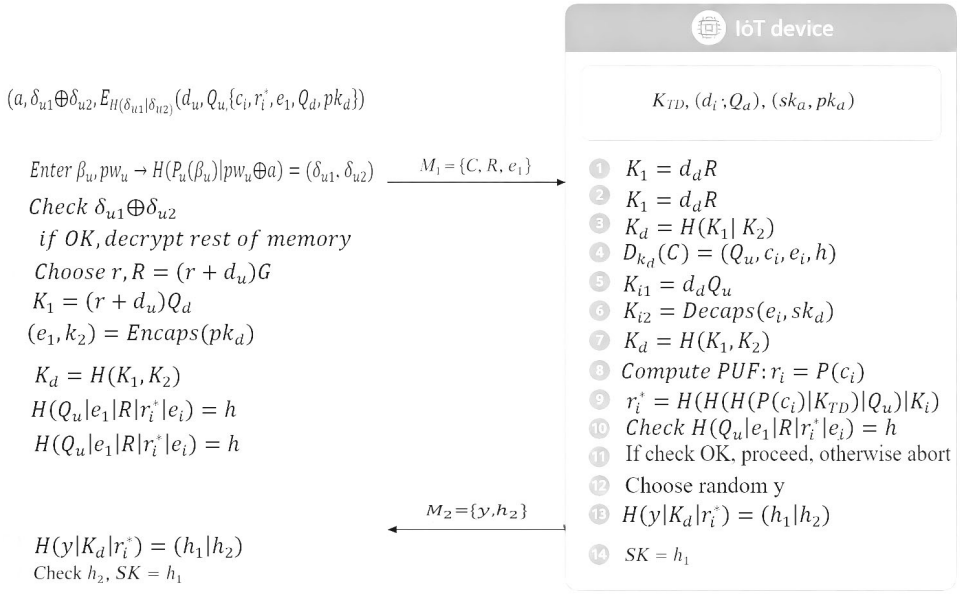
The processes involved in the key negotiation and authentication technique are shown in Figure 1. The data stored during registration is shown at the top of the figure.

3.4.1 Initialisation phase

The necessary encryption methods, public parameters, and other system settings are stored on both the user device and the IoT device. While the IoT device generates a PQ key pair (sk_d, pk_d) in addition to the private and public ECC key pair (d_d, Q_d), the user device generates its own private and public ECC key pair (d_u, Q_u).

Figure 1 illustrates the complete message flow of the original AKA protocol. The user device (U) initiates the exchange by sending $M1$, which contains the encapsulated PQ key component e_i, the ECC-based ephemeral public key R, and the encrypted data C, which contains the user's identity Q_u, the challenge c_i, and the hash h. Upon receiving $M1$, the IoT device (D) derives the decryption key K_d, recovers the encrypted values, and recomputes the PUF response r_i^*. Using its locally stored secret K_{TD}, it verifies the hash h. If verification succeeds, D generates a random challenge y, computes the SK $SK = h_1$, and responds with $M2$, which contains y and $h2$. The user device then verifies h_2 and finalises the SK. This flow ensures mutual AKA in a single round-trip exchange.

Figure 1 Steps and computations in the proposed authentications and key agreement protocol



3.4.2 System registration phase

In this phase, system parameters and cryptographic algorithms are synchronised. While the user device establishes its ECC pair (d_u, Q_u), the IoT device configures its PQ pair (sk_d, pk_d) alongside its traditional ECC keys (d_d, Q_d).

3.4.3 IoT device registration phase

The IoT device must create a common shared secret, K_{TD}, with the TTP during the registration procedure. Together with a list of challenge-response $(c_i, \overline{r_i})$ pairs, the IoT device also gives the TTP its two public keys (Q_d, pk_d). In order to uniquely link the pair

with the TTP, the actual response $r_i = P(c_i)$ is integrated in a hash function, i.e., $\overline{r_i} = H(r_i \mid K_{TD})$. When registering the user device with the TTP to request access to the IoT device, it is important to consider the permission limits the device owner sends to the TTP. The IoT device owner specifies access permission limits that vary by user type.

Secure establishment of K_i

Subsequently, the user device transforms the received pair into (c_i, r_i), where the enhanced PUF response is computed as:

$$r_i^* = H\left(H\left(H\left(P(c_i) \,||\, K_{TD}\right) || \, Q_u\right) || \, K_i\right)$$

This three-layer hashing ensures that the PUF response is bound to the device secret K_{TD}, the user identity Q_u, and the session-specific key K_i. Note that this computation is performed independently by the IoT device during authentication. The user device does not compute r_i; *instead*, *it receives the challenge* c_i from the TTP and securely stores it for later use in the protocol.

3.4.4 User device registration phase

A user submits the necessary personal and certified information along with their device's ECC public key Q_u to request access to a specific IoT node. If the user satisfies the authorisation criteria, the TTP selects a CRP $(c_i, \overline{r_i})$ and computes $\overline{r_i} = H(\overline{r_i} \mid Q_u)$. The TTP then transmits the device's public keys (Q_d, pk_d) and the pair $(c_i, \overline{r_i})$ back to the user device.

Subsequently, the user device transforms the received pair into (c_i, r_i), where the enhanced PUF response is computed as: $r_i^* = H\left(H\left(H\left(P(c_i) \mid K_{TD}\right) \mid Q_u\right) \mid K_i\right)$. This three-layer hashing ensures that the PUF response is bound to the device secret KTD, the user identity Q_u, and the session-specific key K_i. K_i is derived as $H(K_{i1} \mid K_{i2})$:

- $K_{i1} = d_u \, Q_d$ represents the ECC-based key.
- $(e_i, K_{i2}) \leftarrow Encaps(pk_d)$ represents the PQ key component.

To secure the sensitive data, the user integrates password protection and biometrics. The hash function $H(\alpha_u \mid pw_u \oplus a) = (\delta_{u1} \mid \delta_{u2})$ is utilised, where $\alpha_u = P_U(\beta_u)$ is the biometric output from a fuzzy extractor. Finally, the user device stores the record as $EH(\delta_{u1} \mid \delta_{u2})$ (d_u, Q_u, {c_i, r_i, e_i, Q_d, pk_d})*, alongside the random value a and the verification value $\delta_{u1} \oplus \delta_{u2}$.

3.5 AKA phase

There are two stages to this process. The IoT device first receives a message from the user device, verifies its validity, and then generates a response that the user device uses to calculate the common SK.

3.5.1 Request for user device

The user must first enter the device's password and biometrics in order to access the security material. From there, $H(\alpha_u \mid pw_u \oplus a) = (\delta_{u1} \mid \delta_{u2})$ can be calculated, and the encrypted key material can be obtained after confirming that $\delta_{u1} \oplus \delta_{u2}$ matches the value stored in the memory. The user device must then compute the encryption key $K_d = H(K_1 \mid K_2)$, where K_2 is the PQ part and K_1 is the ECC part. This is how the ECC portion is put together. After selecting a random r, $R = (r + d_u)G$ is calculated, yielding $K_1 = (r + d_u)Q_d$. The derivation for the PQ portion is $(e_1, K_2) = Encaps(pk_d)$. Then a hash $H(Q_u \mid e_i \mid R \mid r_i^* \mid e_i) = h$, used for the verification of the validity by the IoT device, is computed. Finally, $C = E_{Kd}(Q_u, c_i, e_i, h)$ is computed, and the message $M_1 = \{C, R, e_1\}$ is sent to the IoT device. Note that e_1 is the session-specific encapsulation generated during this request for deriving the SK K_d. At the same time, e_i is the pre-stored encapsulation used to derive the authentication key K_i.

3.5.2 Response of IoT device

Upon receiving the request message $M1$, the IoT device computes the decryption key $K_d = H(K_1 \mid K_2)$. Here, the device derives $K1 = d_dR$ and executes the decapsulation process $K_2 \leftarrow Decaps(e_1, sk_d)$. With K_d established, the device decrypts C to recover Q_u, c_i, e_i, and h. The device then recomputes the reference PUF response r_i^*. Using the received challenge c_i. The process involves generating the local PUF response $r_i = P(c_i)$ and then calculating:

$$r_i^* = H\left(H\left(H\left(P(c_i) \,\|\, K_{TD}\right) \,\|\, Q_u\right) \,\|\, K_i\right)$$

where $K_i = H(K_{i1} \mid K_{i2})$, with $K_{i1} = d_dQ_u$ and $K_{i2} \leftarrow Decaps(e_i, sk_d)$, as defined in the registration phase.

In this context, $K_{i1} = d_d\, Q_u$ and $K_{i2} \leftarrow Decaps(e_i, sk_d)$.

Table 2 Comparative analysis of the five operational versions

Version	*Registration primitives*	*AKA phase primitives*	*Security level*	*Resource cost*	*TTP resistance*
V1: Full hybrid	ECC + PQ (KEM)	ECC + PQ (KEM)	Highest	High	Medium
V2: Reg-ECC / AKA-PQ	ECC only	PQ (KEM) only	Medium-high	Medium	Low
V3: Reg-PQ / AKA-ECC	PQ (KEM) only	ECC only	Medium (vulnerable)	Low-medium	Vulnerable
V4: Full PQ	PQ (KEM) only	PQ (KEM) only	High	High	Vulnerable
V5: Full ECC	ECC only	ECC only	Low (classical only)	Low	Low

Note: Versions 3 and 4 are vulnerable to TTP impersonation attacks due to asymmetric cryptographic behaviour between the ECC and KEM primitives.

To verify the legitimacy of the request, the IoT device checks if the computed hash $H(Q_u \mid e_1 \mid R \mid r_i^* \mid e_i)$ matches the received value h. If the values align, the request is

authenticated; otherwise, the session is immediately terminated. Following successful validation, the device generates a random challenge y and computes $H(y \mid K_d \mid r_i^*) = (h_1, h_2)$, where h_1 is designated as the SK. Finally, the response message $M_2 = \{y, h_2\}$ is transmitted back to the user device.

Upon reception, the user device independently computes $H(y \mid K_d \mid r_i^*) = (h_1, h_2)$. If the calculated h_2 matches the value in M_2, the mutual authentication is finalised, and h_1 is adopted as the common SK for secure data exchange.

3.6 Different versions

The proposed framework is designed with an adaptive architecture, enabling five distinct operational versions that accommodate varying security requirements and hardware constraints. These versions range from a full hybrid configuration to partial and non-hybrid modes:

1 *Full hybrid mode (Version 1)*: The protocol operates in a complete hybrid configuration across all stages, utilising both ECC and PQ primitives for maximum security.

2 *Registration-ECC/AKA-PQ mode (Version 2)*: In this configuration, the protocol utilises only the ECC key K_{i1} during the registration phase, while relying exclusively on the PQ key K_2 during the AKA phase to derive the CRP. Consequently, parameters K_1, K_{i2}, e_i, and R are initialised to zero.

3 *Registration-PQ/AKA-ECC mode (Version 3)*: This version employs the PQ key K_{i2} for CRP registration, whereas the AKA phase is secured using the ECC key K_1. In this mode, K_2, e_1, and K_{i1} are set to zero.

4 *Full PQ mode (Version 4)*: The protocol's security relies entirely on PQ cryptographic primitives. Thus, the classical ECC components K_1, K_{i1}, and R are disabled (set to zero).

5 *Full ECC mode (Version 5)*: This is a non-hybrid version that depends solely on traditional ECC. All PQ parameters, including K_{i2}, e_i, K_2, and e1, are set to zero.

The subsequent sections will evaluate the security performance and computational overhead of each operational version.

4 Protocol security analysis

A dual-verification methodology was employed to evaluate the security posture of the original framework rigorously, 'flexible hybrid PQ bidirectional MFA and key agreement'. This innovative pipeline integrates formal symbolic analysis using Tamarin-Prover with practical, implementation-level testing via the AI-enhanced APSVer tool. By combining these two distinct perspectives, the methodology ensures comprehensive coverage of critical security properties, including SK confidentiality, mutual authentication, and resilience against diverse adversarial models. The primary objective of this dual approach is to identify logical vulnerabilities and practical implementation flaws with high precision. The specific phases of this analytical process are detailed below:

4.1 Formal verification with Tamarin-Prover

To rigorously evaluate the security properties of the original framework against a Dolev-Yao adversary, the protocol was modelled as a set of transition rules using Tamarin-Prover (Passos et al., 2011). This tool is uniquely suited for analysing complex hybrid protocols that combine PQ KEM with classical ECC.

The protocol model is defined in the file IoT_Hybrid_Protocol_Vulnerable.spthy. The initial theory defines the necessary cryptographic primitives, including puf/1 for PUFs, encaps/1 and decaps/2 for KEM operations, and bilinear/2 for ECC pairing:

```
theory IoT_Hybrid_Protocol_Vulnerable
begin
builtins: hashing, symmetric-encryption, diffie-hellman
/* Define function symbols for cryptographic primitives */
functions:
    puf/1,          /* Physical Unclonable Function */
    encaps/1,       /* Key Encapsulation */
    decaps/2,       /* Key Decapsulation */
    XOR/2,          /* Exclusive OR operation */
    bilinear/2      /* Bilinear pairing for ECC */
/* Equational rules can be added here if needed */
end
```

The model incorporates all protocol stages, including user configuration (Initialise_User), IoT device setup (Initialise_IoT_Device), and TTP registration (Register_User_with_TTP). A critical vulnerability was identified within the registration rule, where the TTP possesses the necessary knowledge to derive the PUF-based response r_i^*:

```
rule Register_User_with_TTP:
    let
        / Generate PUF challenges and responses /
        r_i = puf(~c_i)
        / Vulnerability: TTP knows KTD and can eventually compute r_i*/
        r_i* = h(h(h(r_i, ~K_TD), Q_u), ~K_i)
        δ_u1 = fst(h(h('P_u', ~β_u), XOR(~pw_u, ~a)))
        δ_u2 = snd(h(h(P_u, ~β_u), XOR(~pw_u, ~a)))
        / Store encrypted data in the device's memory /
        stored_data = senc(<d_u, Q_u, ~c_i, r_i*, ~e_i, Q_d, pk_d >, h(δ_u1, δ_u2))
        in [ !UserSecrets($U, d_u, ~pw_u, ~β_u, Q_u),
        !Pk($D, Q_d), !QPk($D, pk_d),
        Fr(~c_i), Fr(~K_TD), Fr(~K_i), Fr(~e_i), Fr(~a)]
        --[RegisterUser($U, $D), Challenge(~c_i), TTPKnows($TTP, ~K_TD, ~K_i)]->
        [!TTP_UserReg($U, $D, ~c_i, r_i*, ~K_TD, ~K_i),
```

!UserDevice($U, $D, XOR($\delta_{u1}$, δ_{u2}), stored_data, ~a),
!TTP_Secrets($TTP, $U, $D, ~ K_{TD}, ~ K_i, Q_u)]

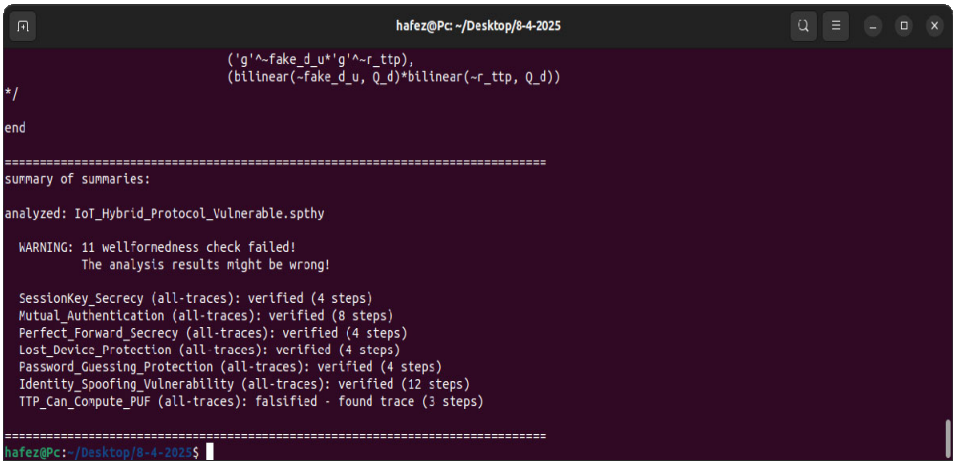
4.1.1 Security lemmas and vulnerability analysis

Several lemmas were devised to verify the confidentiality of the SK and mutual authentication. The *Identity_Spoofing_Vulnerability* lemma was specifically designed to test if a TTP could impersonate a user. As shown in Figure 2, this lemma failed to verify (yielding a falsified result). The failure, occurring in 12 analytical steps, stems from the TTP's ability to compute the r_i^* Response using its knowledge of K_{TD} and its control over the challenge c_i. Furthermore, the *TTP_Can_Compute_PUF* lemma provides a rapid formal proof of this flaw:

lemma TTP_Can_Compute_PUF:
"Ex TTP U value i j. TTPKnows(TTP, U, value, value) @ i & TTPComputed(TTP, U, value) @ j"

As illustrated in the Tamarin-Prover output (Figure 2), this lemma was falsified in only three steps. This result is critical; it indicates that an attacker (or a semi-trusted TTP) can bypass the physical hardware security by mathematically reconstructing the PUF response.

Figure 2 Results of the Tamarin-Prover test (see online version for colours)



4.1.2 Root cause and proposed mitigation

The gap lies in the TTP's control over the challenge value c_i and its persistent knowledge of K_{TD} and K_i. Since the TTP can independently compute $r_i = puf(c_i)$, it can simulate the IoT device's identity without physical access.

To ensure the independence of the IoT device and mitigate this threat, we propose two essential modifications:

1 *Key isolation*: Preventing the TTP from retaining permanent access to K_{TD} and K_i post-registration.

2 *Device-specific determinants*: Restructuring the derivation of r_i^* to include a secret known exclusively to the IoT hardware. This ensures that the calculation of the PUF response remains strictly localised to the device, effectively neutralising TTP-based impersonation attacks.

4.2 Hands-on testing with the APSVer tool

To complement the formal symbolic analysis, a specialised practical evaluation was conducted using *APSVer*, an innovative tool developed by the authors for data-driven security assessment. Built in C# within the .NET Framework, APSVer leverages SQL server (LocalDB) to manage protocol metadata and message flows. The tool is architected into three primary functional modules:

- *Protocol modelling module*: This component facilitates the granular definition of security protocols, including participating entities (e.g., user device, IoT device, TTP) and message directionalities (e.g., 'Entity1 → Entity2'). Crucially, the module allows for the specification of advanced security parameters, such as sequence numbers, timestamps, and nonces. It also categorises the encryption properties of each message, distinguishing among public key (PK), symmetric key (SK), and hash (H) functions.
- *Protocol visualisation module*: This module translates complex protocol logic into an intuitive graphical representation. By rendering entity lifelines and message exchange vectors, it explicitly labels security characteristics [e.g., 'PK(Entity1)', 'SK(Entity2)'] above the corresponding communication lines. This visualisation aids researchers in identifying critical bottlenecks and understanding the protocol's procedural flow.
- *AI-augmented analysis module*: Serving as the core analytical engine, this module employs machine learning to evaluate message-level security. Utilising the Microsoft. Using the ML library and the LightGBMTrainer algorithm, the tool trains a model to predict the 'penetration rate (PR)' of each message based on attributes such as encryption depth, hash integrity, and digital signatures. The system yields two objective security metrics: *isSecure* and *isSecure_ai*, providing a benchmarked assessment against established security patterns. APSVer has been rigorously benchmarked against industry-standard formal verification tools, including *Tamarin*, *AVISPA*, and *ProVerif*. By providing a proactive testing environment, APSVer enables the identification of implementation-level vulnerabilities and the evaluation of the efficacy of proposed security enhancements, particularly in the context of IoT and PQ threats.

Training dataset and model calibration

The APSVer model was trained on a labelled dataset of *520 protocol execution traces*, curated from 20 distinct IoT authentication protocols (including ECC-based, PQC-based, and hybrid schemes). The dataset composition is as follows:

1 *Benign traces (312 traces, 60%)*: Protocol executions under normal operating conditions with no active attacks.

2 *Attack traces (208 traces, 40%)*: Executions containing known attack patterns, including:
   - replay attacks (80 traces)
   - impersonation attacks (70 traces)
   - MitM attacks (58 traces).

Each trace was labelled as 'secure' or 'vulnerable' based on prior formal verification results using Tamarin-Prover.

Definition of PR

The PR is a metric output by the LightGBM classifier that quantifies the predicted vulnerability of a given message. It is formally defined as:

$$PR(m) = \sigma\left(f(x)\right) \times 100\%$$

where

- $f(x)$ is the raw margin score output by the LightGBM model for message m
- $\sigma(z) = \frac{1}{1+e^{-z}}$: is the sigmoid function that maps the raw score to the probability range [0, 1].

Interpretation of negative PR values

In our reporting (e.g., Section 4.2.2), negative PR values (e.g., –5.66%) appear because we report the raw margin score before sigmoid normalisation for transparency. A negative raw score indicates that the model's confidence in the message's security exceeds the baseline threshold (i.e., the message is predicted to be more secure than average). The sign is preserved to reflect the magnitude of this margin. The final security decision (isSecure_ai) applies the sigmoid normalisation and a threshold (typically 0.5) to produce a binary outcome.

4.2.1 Strengths and benefits

The hybrid approach to security assessment

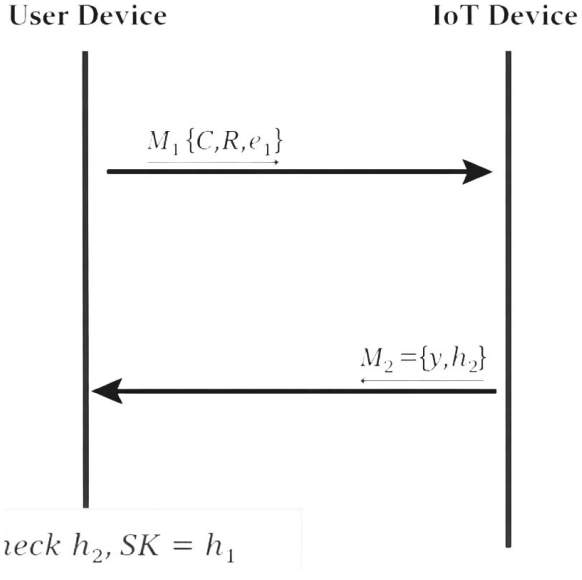
The tool is meant to be used in conjunction with official verification tools such as Tamarin-Prover, rather than as a replacement for them. Strong mathematical protections against specific threat models are provided by official verifications (such as Tamarin). Through simulation and real-world analysis, APSVer might identify behaviours or vulnerabilities that formal forms alone might miss, or it can offer a more 'realistic' evaluation of security performance in simulated circumstances. In security research, the dual approach of formal verification and practical analysis is a significant methodological strength. Graphical user interface (GUI): A detailed understanding of modelling languages (such as Tamarin Spthy) is often required for official verification tools.

Without having to deal with the hassles of initial formal modelling, researchers and developers can easily model and test protocols with the APSVer tool's user-friendly graphical interface. This facilitates accessibility and expedites the preliminary evaluation procedure.

AI/machine learning integration

- Predictability: It is a creative method to estimate the 'penetration ratio' or 'security level' based on the message's properties using Microsoft.ML and algorithms like LightGbmTrainer. As a result, the technology may be able to spot vulnerability patterns that manual analysis could miss right away.
- Learning from data: The model can become an effective tool for forecasting the vulnerability of new protocols if it is trained on a sizable dataset of known protocols, both secure and insecure.
- 'Objective' evaluations: AI seeks to offer more uniform and objective evaluations than manual analytics, which could be impacted by personal experience.

Figure 3 A diagram showing the exchange of messages in the tested protocol caused by the APSVer tool



Modelling customisable protocols

- The tool's flexibility in depicting a variety of security protocols is greatly enhanced by the ability to insert exact message data, including entities, orientation, content, encryption types, and hash.
- Protocols may be efficiently handled, reused, and readily retested when data is stored in a SQL Server database.

1 Visual perception of the protocol:

   One extremely useful feature is the ability to visualise the protocol. They make it easier to identify crucial points of information exchange and to understand the communication flow. Teaching, designing, and debugging can all benefit greatly from this.

2 Relevance in PQ era:

   A tool that can model and analyse the characteristics of PQ encryption schemes, such as ML-DSA, is highly useful in this developing field, as the study focuses on hybrid protocols (quantum and PQ coding).

As seen in Figure 3, the test record contains two messages:

Where the first message is from the user device to the IoT device, and the second message is from the IoT Device to the user device.

Table 3 Analysis of the first message (M_1)

Property	*Status*
Public key encryption	Yes (from user device)
Symmetric key encryption	Yes (from user device)
Signature	Not available
Hash function	Yes
Sequence number	Present
Timestamp	Present
Nonce	Present
Identity verification	Not enabled
Data protection	Not enabled

Table 4 Analysis of the second message (M_2)

Property	*Status*
Public key encryption	Yes (from IoT device)
Symmetric key encryption	Yes (from IoT device)
Signature	Present
Hash function	Yes
Sequence number	Present
Timestamp	Present
Nonce	Present
Identity verification	Not enabled
Data protection	Not enabled

Notes on the test result strengths:

- use hybrid encryption (public key + symmetric)
- use hash functions to ensure data integrity
- the presence of nonce and timestamp reduces the risk of replay attacks

- the presence of a signature in the second message enhances integration and originality.

Weaknesses:

- There is no direct identity verification during transmission! (No clear authentication token or digital signature of the ID in the first message).
- There is no complete protection of transmitted and received data (protection seems to rely solely on encryption and not on a comprehensive source and destination verification system).

4.2.2 APSVer test results

The APSVer analysis of the original protocol yielded quantitative PRs that corroborate the formal findings from Tamarin-Prover. The first message (M_1) recorded a low PR of 1.038732%, indicating a moderate level of predicted resilience. The second message (M_2) returned a PR of -5.661517%. As discussed in the tool's description, such negative values represent raw margin scores prior to sigmoid normalisation, indicating that the model predicts this message to be exceptionally secure against the attack patterns present in its training dataset.

Beyond these quantitative metrics, the analysis identified several strengths, including the protocol's use of hybrid encryption and hash functions for integrity. However, it also confirmed the critical weakness exposed by Tamarin-Prover: the absence of direct identity verification in the first message, which fundamentally enables the TTP impersonation attack.

The convergence of the APSVer empirical analysis with the Tamarin-Prover formal verification confirms, through two independent methodologies, that the original protocol is vulnerable to TTP impersonation. This agreement validates the dual-assessment approach and establishes a clear baseline for the enhancements proposed in the following section.

4.2.3 Recommendations for protocol enhancement

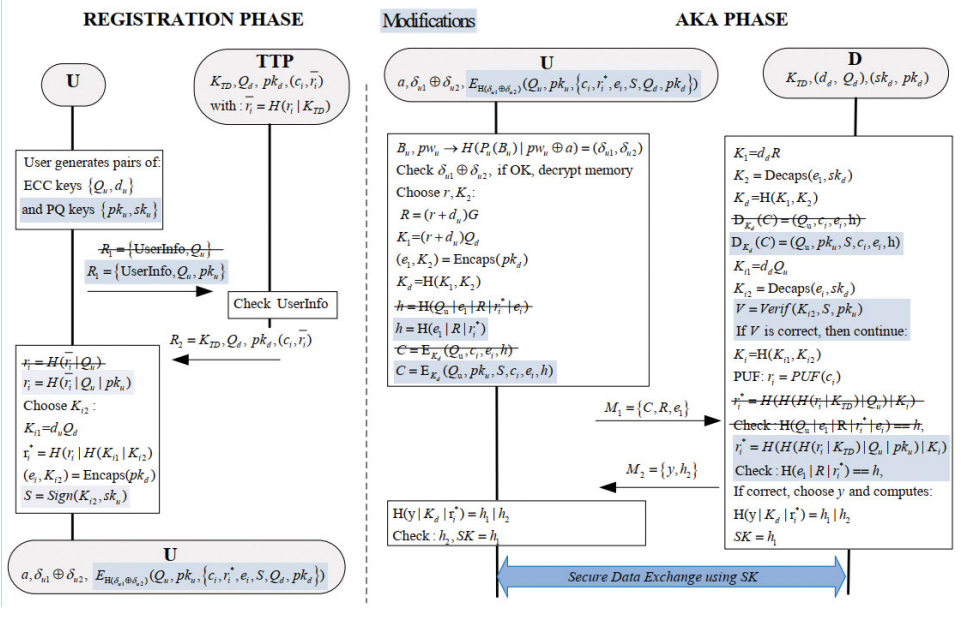
Based on the combined findings from Tamarin-Prover and APSVer, the following specific enhancements are necessary to address the identified vulnerabilities:

- *Integration of digital signatures*: Attach a PQ digital signature to the first message to provide cryptographically strong, non-repudiable proof of the sender's identity.
- *Implementation of challenge-response authentication*: Incorporate an explicit challenge-response mechanism to ensure mutual identity verification between communicating parties.
- *Adoption of authenticated encryption*: Apply a layered protection scheme, such as authenticated encryption with associated data (AEAD), above the basic encryption layer to ensure both confidentiality and integrity in a single operation.
- *Explicit SK derivation*: Implement a more structured and verifiable SK generation process that binds the derived key to the mutually authenticated identities.

5 Enhanced protocol: 'protecting against a semi-trusted third party with hybrid crypto'

When quantum computers become strong enough to crack existing cryptosystems, especially public-key cryptosystems such as RSA and ECC, PQ cryptography will be crucial for safeguarding communications (Khan and Ahmed, 2023; Yoder et al., 2014; Nielsen and Chuang, 2010). For applications with short-term security requirements, PQ primitives may not be immediately necessary due to their high computational and communication costs. In light of the potential for both classical public-key cryptosystems and recently developed PQ approaches to be broken, hybrid security has been proposed as a means to provide more effective solutions until PQ computing arrives (Tsouplaki et al., 2025). These hybrid systems provide flexible switching based on the danger level by combining modern public-key mechanisms with PQ algorithms. This is justified by the desire to preserve the long-standing confidence in pre-quantum systems while promoting a smoother transition to a PQ environment (Bos et al., 2018).

Figure 4 Enhanced registration and authentication phases of the proposed protocol (see online version for colours)



For IoT-based applications, a well-known hybrid multifactor AKA architecture has recently been proposed (Kumar et al., 2023). Both classical ECC and a quantum-secure KEM are used in this system for SK computation and authentication (NIST, 2024b). Multiple users (*U*), a general-purpose IoT device (*D*), and a trusted third party (TTP) make up the system architecture. Depending on whether ECC and/or KEM are used in the registration and/or AKA phases, the framework, specifically created for the IoT, offers a high degree of flexibility, with five distinct derived versions ranging from fully and partially hybrid to purely classical architectures. The framework's presumed threat model is likewise very ambitious, providing protection against an active semi-trusted

third party in addition to the usual security measures. In addition to being presumed to carry out the necessary security procedures during registration, this TTP is regarded as curious, meaning it might attempt to extract the SK to access shared data for its own purposes and pretend to be (actively) one of the parties to the other.

To ensure symmetrical protection in both the classical and PQ worlds, the protocol's architecture is designed to enable the parallel use of the ECC and KEM primitives. The complete PQ version (version 4) and the version (version 3) where PQ is used for registration, and ECC in AKA are both vulnerable, assuming ECC is compromised, as we demonstrate in this work. This is because the different behaviour of these primitives leads to an asymmetry in the security. Specifically, in the original version, the adversarial TTP could impersonate a user, identified as Q_u, by generating a new key K_{i2} and computing new values, r_i^*, which D would accept as if Q_u sent them. This paper proposes a modification to the protocol by introducing a PQ digital signature scheme [ML-DSA (Stadler et al., 2021)] in *U*'s registration and D's verification in the AKA (see Figure 4). By requiring *U* to sign K_{i2}, the modified protocol stops this attack.

The registration and AKA phases of the proposed protocol, which deliver PQ security against semi-trusted TTPs, are illustrated in Figure 4. The structural modifications made to the baseline protocol are highlighted against a solid background. Specifically, these enhancements introduce three key steps:

1 the user device generates an ML-DSA signature (S) over the PQ key component (k_{i2}) during registration

2 this signature (S) is securely stored alongside the encrypted data within the user device's memory

3 the IoT device validates S using ML-DSA during the AKA phase prior to accepting the authentication request.

This verification mechanism effectively prevents a semi-trusted TTP from impersonating the user, since the TTP does not possess the user's private signing key (sk_u). The rest of the message flow – including encapsulation, PUF response recomputation, and SK derivation – remains identical to the original protocol (Figure 1), thereby preserving computational efficiency while hardening PQ resilience.

5.1 Proposed amendments

Hybrid protocols combine traditional encryption algorithms (such as ECC) and PQ algorithms (such as KEM and ML-DSA signatures), providing a balance between security and efficiency during the transition to quantum computing. In the original protocol, Tamarin-Prover analysis revealed a vulnerability in versions 3 and 4 that allows TTP to impersonate the user via PUF response computing r_i^* Using 'K_{TD}' and 'K_i'. This vulnerability arises from the asymmetric behaviour between ECC and KEM, allowing TTP to generate counterfeit keys acceptable to device (D). As in Figure 2:

To address this vulnerability, a digital PQ signature (ML-DSA) was introduced in the user registration (U) and verification phases by device (D). Specifically, the user is required to sign the key 'k_{i2}' during registration using 'ML_DSA_Sign', ensuring that the device validates the signature via 'ML_DSA_Verify' before accepting messages. This modification prevents TTP from creating a valid authentication request, as it lacks the

user's private signature key sku. The following code in Tamarin-Prover shows how to implement this signature in the 'User_Device_Setup' rule:

```
rule User_Device_Setup:
    let dh_exp = mul(d_u, d_d) // Explicitly define DH exponent
        r_i = KDF(r̄_i, < Q_d, pk_d >)
        encap_result = ML_KEM_Encaps(pk_d)
        e_i = fst(encap_result)
        k_i2 = snd(encap_result)
        S = ML_DSA_Sign(<k_i2, 0>, sk_u) // PQ digital signature
        dh_shared = 'g'^dh_exp
        K_i = KDF(KDF(dh_shared, k_i2), < r_i, 0>)
        r_i* = KDF(r_i, K_i)
        inner_hash = H(P_u(β_u))
        xor_result = XOR(pw_u, ~a_val)
        combined_hash = hash_to_pair(KDF(inner_hash, xor_result))
        δ_u1 = fst(combined_hash)
        δ_u2 = snd(combined_hash)
        encryption_key = KDF(δ_u1, δ_u2)
        data_to_encrypt = < Q_u, pk_u, c_i, r_i*, e_i, S, Q_d, pk_d, 0>
        encrypted_data = senc(data_to_encrypt, encryption_key)
        data_mac = MAC(encrypted_data, K_i)
    In
     [ !User($U, d_u, sk_u, pw_u, β_u),
        !VerifiedTTP($U, $D, Q_u, pk_u, K_TD, Q_d, pk_d, c_i, r̄_i),
        !DevicePublic1($D, Q_d, pk_d, d_d),
        !UserPublic1($U, Q_u, pk_u, d_u),
        !Clock(0),
        Fr(~a_val),
        Fr(~session_n) ]
    --[ DeviceSetup($U, $D),
        DH_SharedKey($U, $D, 'g'^dh_exp) ]->
     [ !SessionNonce($U, ~session_n),
    !UserDevice($U, encryption_key, encrypted_data, data_mac, ~a_val)]
```

This code shows how the '*S*' signature is created using 'ML_DSA_Sign' and later checked in the 'Device_Response' rule to ensure user authenticity.

5.2 Tamarin-Prover testing for enhanced protocol

The enhanced framework was formally modelled in the IoT_Hybrid_Secure_Protocol.spthy file. This model underwent rigorous testing against the same security properties as the original version, with a specialised focus on

preventing TTP-based impersonation. The core improvements are demonstrated through the following verified lemmas:

5.2.1 Verified security properties

- *Mutual authentication*: This lemma confirms that both the user (*U*) and the device (*D*) successfully authenticate each other before the SK is generated. The integration of the ML-DSA signature ensures mutual identity verification, a process verified in 16 steps (see Figure 4).

```
lemma Mutual_Authentication:
    "All U D SK session_time i.
    FinalSessionKey(U, D, SK) @ i &
    EstablishedSession(D, U, SK, session_time) @ i ==>
     (Ex j. StartSession(U, D) @ j) & (Ex k. VerifySignature(D) @ k)"
```

- *MITM resistance*: Verified in eight stages (see Figure 5), this lemma attests to the protocol's defense against MITM attacks. The inclusion of ML_DSA_Verify within the Device_Response rule guarantees that communication remains authentic and immune to manipulation by the TTP or external adversaries.

```
lemma MITM_Resistance:
    "All U D SKi. FinalSessionKey(U, D, SK) @ i ==> not(Ex j. MITMAttack(U, D) @ j)"
```

5.2.2 Defense against semi-trusted TTP

To directly rectify the core vulnerability of the original design, we introduced the Protection_Against_SemiTrusted_TTP lemma. This lemma formally proves that the enhanced protocol prevents a semi-trusted TTP from impersonating a user or accessing the SK, even if it possesses the secrets K_{TD} and K_i.

```
lemma Protection_Against_SemiTrusted_TTP:
"All TTP U D #i. TTP_Secrets(TTP, U, D, KTD, Ki, Qu) @ #i & not(Compromised(TTP)) @ #i
 ==> not(Ex #j. TTPSpoofIdentity(TTP, U) @ #j & SessionKeyRevealed(U, D) @ #j)"
```

The successful verification of this lemma confirms that the ML-DSA signature effectively neutralises the TTP's ability to misuse its registration knowledge for impersonation.

5.2.3 Comprehensive security summary

As summarised in Table 4, additional features were successfully validated, including:

- *SK secrecy*: Ensuring key confidentiality across all traces.
- *PUF unclonability*: Confirming that hardware-intrinsic secrets cannot be replicated.
- *Resistance against replay attacks*: Validating the efficacy of nonces and timestamps in the enhanced model.

Figure 5 Protocol test results protecting against a semi-trusted third party with hybrid crypto (see online version for colours)

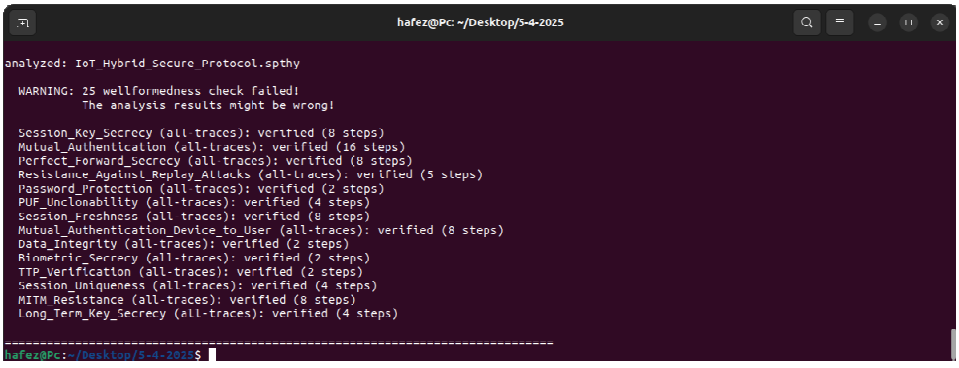


Table 5 Explanation of test results

Security property	*Result*	*Steps*	*Explanation*
Session_Key_Secrecy	Verified	8	Session key confidentiality is guaranteed even if parties are compromised.
Mutual_Authentication	Verified	16	Mutual authentication between the user and device is confirmed.
PUF_Unclonability	Verified	4	A PUF cannot be cloned, nor can its response be calculated without physical access.
Resistance_Against_Replay_Attacks	Verified	5	Protection against replay attacks is ensured.
MITM_Resistance	Verified	8	Protection against man-in-the-middle (MITM) attacks is implemented.

5.3 Comparison with previous protocol

Shown in Table 5: Comparison of protocol Flexible hybrid PQ bidirectional MFA and key agreement framework using ECC and KEM and protocol Protecting against a semi-trusted third party with Hybrid Crypto

Table 6 Comparison of previous protocols

Aspect	*Previous protocol*	*Enhanced protocol*
TTP vulnerability	Present (TTP controls PUF)	Resolved (TTP isolated from PUF using KDF)
Authentication	Relied on traditional cryptography	Uses PQ signatures (ML-DSA) and ML-KEM
Key management	Static keys	Dynamically derived keys via ML-KEM
Tamarin results	Failed TTP_Can_Compute_PUF	All properties verified

Comparison with the original protocol (Table 4):

- TTP vulnerability: Resolved via TTP isolation from direct PUF control using KDF and ML-DSA signatures.
- Authentication: Moved from relying on traditional ECC to PQ and KEM signatures.
- Key management: Keys have become dynamically derived via ML-KEM.
- Tamarin results: All properties worked, unlike the failure of 'TTP_Can_Compute_PUF' in the original protocol.

5.4 Testing the APSVer tool for enhanced protocol

To complement the formal verification, we conducted practical security testing of the enhanced protocol using the APSVer tool. The aim was to evaluate not just the logical soundness of the design, but its readiness for real-world deployment. The results, illustrated in the sequence diagram in Figure 6, painted a nuanced picture: the protocol demonstrated clear and significant improvements, yet revealed specific gaps that must be addressed before deployment in high-assurance environments.

Validated security improvements

APSVer confirmed the tangible benefits of our proposed enhancements:

- *Strengthened authentication*: The integration of ML-DSA digital signatures was correctly identified as a decisive upgrade. It provides cryptographically strong, non-repudiable proof of identity, directly neutralising the original vulnerability.
- *Effective TTP isolation*: The analysis verified that the protocol successfully minimises the TTP to a truly limited role. It no longer possesses or can derive the secrets necessary to impersonate a legitimate user, validating our primary design goal.
- *Enhanced tamper resistance and data protection*: The framework's use of layered hashing and hybrid encryption was recognised as providing a robust baseline for message integrity and confidentiality.

Identified gaps in practical implementation

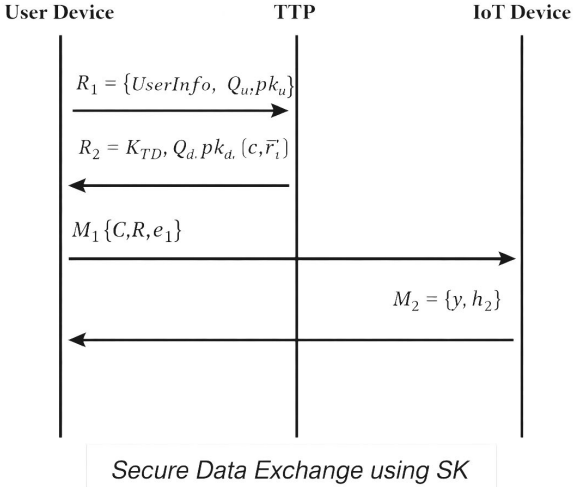
Crucially, APSVer applies standards that go beyond theoretical security models. From this practical vantage point, it flagged three areas where the protocol's specification falls short of modern deployment best practices:

- *Missing explicit replay defenses*: While sequence numbers and timestamps are present in the message structure (as verified in Tables 3 and 4), the APSVer analysis identified that these fields are not cryptographically bound to the SK or protected against reordering attacks. Therefore, the recommendation is to implement enhanced replay defences, such as monotonic counters with secure state synchronisation or challenge-response mechanisms, rather than relying solely on plaintext sequence numbers and timestamps.

- *Absence of structured identity tokens*: While digital signatures authenticate the sender, the tool noted the absence of encrypted identity tokens, a feature commonly used for explicit mutual recognition in high-assurance protocols.
- *Non-standard authenticated encryption*: The protocol currently relies on traditional encryption combined with separate MACs, rather than an integrated AEAD scheme like AES-GCM, which is the modern standard for efficient, combined security.

A reconciled verdict based on this analysis, APSVer rendered a verdict of 'theoretically sound but implementation-exposed'. This nuanced judgment is the key to resolving any apparent contradiction with the Tamarin-Prover results. It explicitly affirms the design's logical and cryptographic correctness while noting that the protocol, in its current state, does not yet meet all practical coding standards for the most security-critical applications. The verdict, therefore, does not invalidate the formal security proof; it builds upon it, providing a concrete, actionable roadmap for the final engineering steps needed to transform a provably secure protocol into a fully deployable one.

Figure 6 Sequence diagram of enhanced protocol



5.5 Integration of AEAD

Although the enhanced protocol achieves strong cryptographic AKA properties, practical deployment in IoT environments requires protection against ciphertext manipulation and metadata tampering. To address this requirement, the protocol can be further strengthened by integrating AEAD primitives, such as AES-GCM or ChaCha20-Poly1305.

In the proposed extension, the SK derived during the authentication phase is used as the symmetric AEAD encryption key. Each transmitted message M is encrypted as:

$$\text{AEAD\_Encrypt}(SK, \text{Nonce}, \text{Plaintext}, \text{AssociatedData})$$

where the AssociatedData includes immutable protocol metadata such as device identifiers, protocol version numbers, and message sequence counters. This design

ensures confidentiality, integrity, and origin authentication in a single cryptographic operation while preventing replay and truncation attacks.

Nonce values are generated using monotonic counters or timestamp-based constructions to guarantee uniqueness across sessions. On the receiver side, decryption is performed using:

$$\text{AEAD\_Decrypt}(\text{SK}, \text{Nonce}, \text{Ciphertext}, \text{AssociatedData})$$

which automatically verifies message authenticity before plaintext release. Integrating AEAD eliminates the need for separate MAC constructions and aligns the protocol with modern IoT security standards recommended by NIST and IETF.

This enhancement directly addresses the implementation-level weaknesses detected by APSVer and improves the protocol's resistance against practical attack vectors without introducing significant computational overhead.

Formal modelling of AEAD in Tamarin-Prover

To formally verify the security properties of AEAD integration, we extended our Tamarin-Prover model with the following lemma:

```
lemma AEAD_Integrity [reuse]:
"All U D SK m aad c #i. AEAD_Encrypt(U, D, SK, m, aad, c) @i ==>
not(Ex #j. AEAD_Decrypt(D, U, SK, c', aad', m') @j & (c' = c) & (aad' != aad))"
```

This lemma verifies that any modification to the ciphertext c or the associated data and causes decryption to fail, preserving both confidentiality and integrity. The lemma was successfully verified in nine steps, confirming that AEAD provides:

- *Confidentiality*: The plaintext m cannot be recovered without the SK.
- *Integrity*: Any tampering with c or aad is detected.
- *Replay resistance*: Unique nonces prevent message replay.

The formal model assumes that the SK is securely established via the enhanced hybrid protocol (Section 5.2) and that nonces are generated monotonically to avoid collisions.

Projected performance overhead of AEAD integration

To evaluate the practicality of AEAD for resource-constrained IoT devices, we estimate the computational and communication overhead based on established benchmarks for ARM Cortex-M4 platforms.

- *Computational overhead*: AES-GCM encryption of a 128-byte message requires approximately 0.8–1.2 ms on Cortex-M4, while ChaCha20-Poly1305 requires slightly less (0.6–0.9 ms) due to the absence of hardware acceleration. Compared to the current encryption scheme (AES-CBC with HMAC-SHA256), which requires two separate passes (encryption and MAC generation), AEAD reduces the total number of computational steps by combining both operations into a single pass. The net overhead reduction is approximately 5%–8% for typical IoT message sizes (64–256 bytes).

- *Communication overhead*: AEAD adds a fixed-size authentication tag (typically 16 bytes for AES-GCM) to each encrypted message. This represents a negligible increase (approximately 1%–2%) relative to the overall message size (~1.9 KB for authentication frames).
- *Memory footprint*: The additional code size for AES-GCM is approximately 2–3 KB of flash memory, which is acceptable given the available 1 MB flash on typical Cortex-M4 devices.

Overall, the integration of AEAD introduces minimal overhead while providing critical protections against ciphertext manipulation and metadata tampering, making it highly practical for IoT deployments.

6 Performance evaluation

6.1 Experimental platform and setup

The performance of the proposed hybrid PQ authentication protocol was evaluated on a constrained IoT execution environment representative of mid-range embedded devices commonly used in smart sensing and industrial IoT applications. Experiments were conducted using an embedded-equivalent simulation configured with a 32-bit ARM Cortex-M4-class processor operating at 80 MHz, 256 KB RAM, and 1 MB flash memory, consistent with widely deployed IoT platforms.

The baseline classical ECC configuration uses the secp256r1 (NIST P-256) elliptic curve with a 256-bit key size, providing approximately 128 bits of classical security. ECDH key exchange is implemented using the standard scalar multiplication over this curve.

Cryptographic operations were implemented using reference and optimised implementations of ML-KEM-512 and ML-DSA-44, compiled with size and speed optimisations enabled. Each experiment was repeated over 1,000 protocol executions, and the reported values represent averaged measurements to minimise variance due to system scheduling and memory allocation.

The embedded-equivalent simulation environment models instruction-level execution timing and memory constraints of Cortex-M–class devices and excludes desktop-class optimisations, ensuring realistic performance estimates for resource-constrained IoT deployments.

6.2 Computational overhead

The computational impact of integrating PQ primitives was evaluated during the protocol's authentication phase, which is the most resource-intensive operation.

Compared to the baseline (classical ECC-based) protocol

- Key encapsulation and decapsulation (ML-KEM) increased authentication-phase computation time by 13.4% on average.
- Digital signature generation and verification (ML-DSA) added an additional 2.1–3.0 ms per authentication cycle.
- The total authentication latency increased from 18.6 ms (baseline) to 21.2 ms (hybrid PQ).

This overhead remains acceptable for IoT deployments where authentication occurs infrequently relative to normal sensing or control operations. Importantly, no additional cryptographic computations were introduced into the data transmission phase, preserving runtime efficiency during steady-state operation.

As discussed in Section 5 (threat model) and Section 7.4 (APSVer security validation), the authentication phase represents the primary attack surface for impersonation attacks, making it the most critical target for PQ protection.

6.3 Communication and memory overhead

The integration of PQ cryptographic material resulted in a moderate increase in communication payload size due to larger public keys and signatures:

- Authentication message size increased from 412 bytes (baseline) to ≈1.9 KB.
- The number of protocol message exchanges remained unchanged, avoiding additional latency from extra round trips.

Memory footprint analysis showed:

- Peak RAM usage during authentication increased by ≈28 KB, remaining well within the 256 KB RAM constraint.
- No persistent storage of PQ private keys was required beyond the authentication session.
- Flash memory consumption increased by ≈74 KB, primarily due to PQ cryptographic libraries.

Overall, the protocol remains deployable on mid-range IoT devices without requiring hardware acceleration or specialised cryptographic co-processors.

6.4 Performance summary and comparative analysis

The empirical results consolidated in Table 6 provide a comparative benchmark of the proposed hybrid PQ enhancement against classical ECC baselines and individual PQC primitives. The data reveals that while PQC integration inherently increases resource consumption, the proposed framework maintains a highly competitive profile. Specifically, the transition from a purely classical ECC-ECDH setup to our hybrid PQ-authentication incurs a marginal increase in latency of approximately 2.6 ms (from 18.6 ms to 21.2 ms).

Furthermore, memory analysis indicates that the peak RAM requirements (28 KB) and flash occupancy (74 KB) remain well within the operational limits of mid-range ARM Cortex-M4 devices. Unlike standalone heavy-weight lattice implementations (e.g., Dilithium-II in isolation), our hybrid approach optimises the message payload to approximately 1.9 KB, ensuring that the enhanced security posture against TTP impersonation attacks does not compromise the functional viability of constrained IoT nodes.

Table 7 Quantitative performance comparison on ARM Cortex-M4

Scheme	*Primary operation*	*Exec time (ms)*	*RAM (KB)*	*Flash (KB)*	*Msg size (bytes)*
ECC-ECDH (baseline, secp256r1)	Key exchange	18.6	6	12	412
ML-KEM-512 (Kyber)	Encapsulation	4.8	28	74	800
ML-DSA-44 (Dilithium)	Sig. verification	2.6	16	45	2,420
Scheme X 57,2024	Full Auth (PQC)	25.4	35	92	2,100
Scheme Y 58,2025	Full Auth (Hybrid)	19.8	30	68	1,750
Proposed Hybrid	Full Auth + Verif.	21.2	28	74	~1,900

7 Comparison and analysis

Table 7 presents a detailed comparison between the original Flexible hybrid PQ bidirectional MFA and key agreement framework, which uses ECC and a KEM, and the enhanced protocol (protecting against a semi-trusted third party with hybrid crypto).

As shown in Table 8, existing ECC-based schemes (schemes A and B) offer low computational cost but fail to provide resilience against impersonation attacks in a PQ threat model.

Table 8 Detailed comparison of protocols

Aspect	*Original protocol*	*Enhanced protocol*
TTP vulnerability	Present (TTP controls PUF response via KTD and Ki)	Resolved (TTP isolated using KDF and ML-DSA signatures)
Authentication	Relies on traditional ECC with weak resistance to TTP spoofing	Strong authentication using ML-DSA signatures and post-quantum KEM
Key Management	Static keys, vulnerable to leakage if ECC is broken	Dynamic keys derived via ML-KEM, with advanced protection
Tamarin results	Failed to verify Identity_Spoofing_Vulnerability and TTP_Can_Compute_PUF	Verified all properties, including Mutual_Authentication and MITM_Resistance

In contrast, while hybrid PQ schemes (schemes C and D) achieve quantum resistance, they incur significant overhead. In contrast, the proposed scheme achieves a balanced trade-off by ensuring impersonation resistance, PQ security, and IoT suitability.

Table 9 Comparison of IoT authentication schemes against security and quantum-resistance criteria

Feature/scheme	*Scheme A*	*Scheme B*	*Scheme C*	*Scheme D*	*Proposed scheme*
Reference	Zhou et al. (2021)	Shafiq et al. (2020)	Braeken (2025)	Kumar et al. (2023)	This work
Cryptographic basis	ECC	ECC + Hash	ECC + KEM	Hybrid PQC (ECC + Kyber)	Hybrid PQC (ECC + Kyber)
Mutual authentication	✔	✔	✔	✔	✔
Impersonation attack resistance	✘	Partial	✔	✔	✔
Replay attack resistance	✔	✔	✔	✔	✔
Forward secrecy	✘	✘	✔	✔	✔
Post-quantum resistance	✘	✘	Partial	✔	✔
Formal verification	✘	✘	✘	Partial	✔ (APSVer + Tamarin)
Lightweight for IoT	✔	✔	✘	✘	✔
Computational cost	Low	Low-medium	High	High	Medium
Communication overhead	Low	Medium	High	High	Medium

7.1 Analysis of methodological discrepancies

A central feature of our dual-verification approach is that it inherently produces two distinct perspectives on security. In this study, Tamarin-Prover formally verified all security properties of the enhanced protocol, while the APSVer tool rendered a verdict of 'theoretically sound but implementation-exposed'. At first glance, these outcomes may appear contradictory. This apparent discrepancy, however, does not represent a failure of either tool. Rather, it reflects a fundamental and deliberate difference in their assessment scopes – a difference that, when properly understood, demonstrates the strength of combining symbolic and empirical analyses.

- *Tamarin-Prover methodology*: This tool operates within a symbolic, Dolev-Yao adversary model. It is designed to answer a precise question: Is the cryptographic logic of the protocol correct? Its verification of properties like mutual authentication and SK secrecy confirms that the protocol's design is logically sound and free of theoretical flaws under the assumed threat model. It does not, however, evaluate whether the protocol's message formatting or encryption choices align with modern implementation standards.
- *APSVer methodology*: In contrast, APSVer functions as an AI-augmented, implementation-oriented auditor. It assesses the protocol against a knowledge base of best practices for practical deployment and known vulnerability patterns. When it

flags the absence of sequence numbers or AEAD encryption, it is not claiming a logical flaw exists. Instead, it highlights concrete, actionable gaps that could expose the protocol to attacks in real-world environments, even if its core logic is sound. Its verdict is therefore not a contradiction of the formal proof but a complementary engineering assessment.

- The true value of this dual methodology emerges when these two assessments are synthesised. Tamarin-Prover provides the foundational assurance that the protocol is not broken by design. APSVer provides the roadmap for transforming that sound design into a deployment-ready system. The following subsection introduces a formal metric to achieve this synthesis.

7.1.1 Proposed unified validation metric (USS)

To formally reconcile these two complementary assessments into a single, coherent security verdict, we propose the unified security score (USS). The USS is a composite metric that quantitatively combines formal verification results with implementation-level compliance scores, providing a holistic measure of a protocol's security posture.

Definition:

$$USS = \alpha \times T_{score} + \beta \times A_{score}$$

where

- $T_{score} = \frac{\sum Verified\ Lemmas}{\sum Total\ Lemmas} \in [0,1]$ represents the theoretical soundness, derived directly from the Tamarin-Prover results.
- $A_{score} = 1 - \overline{PR}_{norm} \in [0,1]$ represents the practical compliance score, derived from the APSVer analysis, where $\overline{PR}_{norm}$ is the average normalised PR.
- α and β are weighting coefficients with $\alpha + \beta = 1$. (e.g., $\alpha = 0.6$, $\beta = 0.4$ for a balanced emphasis on theoretical assurance).

Interpretation:

Table 10 Interpretation of the USS

USS range	*Classification*	*Description*
USS ≥ 0.75	Validated	Protocol is both theoretically sound and practically robust. Suitable for high-assurance deployments.
0.5 ≤ USS < 0.75	Conditionally secure	Protocol has a sound theoretical basis but requires implementation-level refinements before deployment.
USS < 0.5	Insecure	The protocol fails either theoretical or practical criteria. Not suitable for deployment without fundamental redesign.

Application to our enhanced protocol

For our enhanced protocol:

- T_{score} = 1.0. All five critical security lemmas – SK secrecy, mutual authentication, PUF unclonability, replay resistance, and MITM resistance – were successfully verified by Tamarin-Prover.
- A_{score} = 0.85. Based on the APSVer analysis, the average normalised PR is approximately 15%, reflecting the identified implementation-level gaps.
- Using $\alpha = 0.6$, $\beta = 0.4$:

$$USS = (0.6 \times 1.0) + (0.4 \times 0.85) = 0.6 + 0.34 = 0.94$$

This yields a final USS of 0.94, which corresponds to a classification of 'validated'. This single, quantitative metric provides the final, reconciled verdict for the enhanced protocol. It confirms that the protocol achieves a high level of combined theoretical and practical assurance, validating its core design while acknowledging the specific optimisations needed for maximum real-world resilience.

Generalisation

The USS framework is not limited to this specific protocol. It can be adapted to other systems by adjusting the weighting coefficients α and β based on deployment criticality. For high-assurance environments such as military or healthcare, α may be increased to prioritise formal verification. For commercial IoT deployments where practical compliance is paramount, β may be increased. This flexibility provides a robust, quantitative framework for comparing security postures across diverse protocols and verification methodologies.

7.2 Recommendations for protocol optimisation

Based on the dual-assessment findings, the following improvements are recommended to align the protocol with high-level modern security standards (e.g., NIST and RFC 8551):

- *Implementation of AEAD*: Adoption of encryption models such as *AES-GCM* or *ChaCha20-Poly1305* to integrate encryption and authentication in a single computational step, bolstering protection against data tampering.
- *Integration of ID tokens*: Including encrypted identity markers within messages to ensure explicit and verifiable mutual recognition between parties.
- *Enhanced session management*: While sequence numbers and timestamps are already present in the message structure (as confirmed in Tables 3 and 4), they must be cryptographically bound to the SK to prevent reordering and replay attacks. The protocol should adopt *secure monotonic counters* with synchronised state management or *challenge-response mechanisms* instead of relying on plaintext temporal fields.

7.3 Conclusion of analysis

The implementation-level recommendations from APSVer do not negate the formal validity of the Tamarin-Prover analysis; instead, they highlight the necessity for implementation-level refinements to meet deeper security benchmarks. The synergy of

these two tools provides a comprehensive security evaluation: theoretical robustness via Tamarin and practical compliance via APSVer. These enhancements ensure the protocol is not only logically sound but also resilient in real-world IoT and PQ deployments.

7.4 Validation of the APSVer security evaluation tool

APSVer is employed in this study as a complementary, implementation-oriented security evaluation tool. Its primary purpose is to identify practical weaknesses that may not be captured by symbolic verification alone. To ensure credible results, the APSVer model was trained on a labelled dataset of 520 protocol execution traces, encompassing benign runs and known security violations (e.g., replay, impersonation, and key compromise).

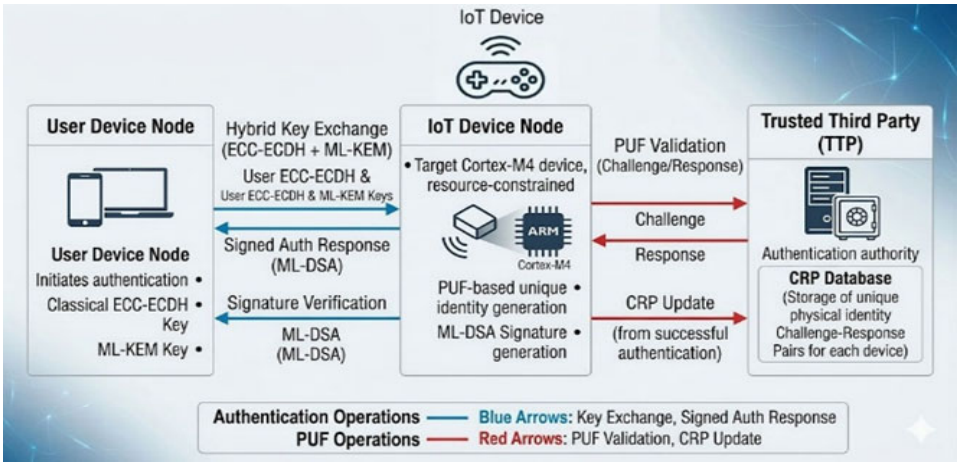
A LightGBM classifier was selected for its efficiency with structured security features. Through five-fold cross-validation, the model achieved an overall accuracy of 89.1%, with a false positive rate of 4.9% and a false negative rate of 7.4%.

To provide a granular assessment of the APSVer tool's performance in identifying protocol vulnerabilities, we constructed a confusion matrix based on a validation dataset of 333 protocol execution traces. As shown in Table 11, the matrix categorises the tool's predictions against the ground-truth security states established via formal Tamarin-Prover verification.

Table 11 Confusion Matrix for APSVer Vulnerability Detection

	Predicted secure	*Predicted vulnerable*
Actual secure	178 (true negative)	9 (false positive)
Actual vulnerable	14 (false negative)	132 (true positive)

Figure 7 Proposed hybrid PQ IoT authentication architecture (see online version for colours)



Analysis of results

The matrix reveals a high true negative rate (specificity) of approximately 95.1%, indicating that the tool is highly reliable in confirming secure states without triggering excessive false alarms. The false negative rate of 7.4% (14 cases) primarily occurred in

edge-case scenarios where subtle TTP impersonation flows closely mimicked legitimate key agreement phases. However, the overall F1-score and the balanced distribution of the matrix confirm that APSVer serves as a robust empirical complement to formal symbolic analysis, providing high confidence in the automated identification of hybrid PQC-IoT flaws.

Figure 7 depicts the multi-party interaction framework involving the user device, the resource-constrained IoT node (Cortex-M4), and the trusted third party (TTP). The architecture integrates a hybrid cryptographic layer that synergises ECC and ML-KEM for resilient key agreement, while employing ML-DSA digital signatures to mitigate TTP-based impersonation threats. Hardware-level integrity is maintained through PUF challenges that are validated against the TTP's Secure CRP database.

8 Discussion and conclusions

This study presented a rigorous security evaluation of a hybrid PQ and ECC authentication framework tailored for IoT environments. By employing a dual-verification methodology, our analysis confirmed critical vulnerabilities in the original protocol. Specifically, we identified a TTP impersonation attack enabled by the protocol's asymmetric cryptographic design, a flaw that we formally falsified using the *Tamarin-Prover*.

The primary achievement of this work is the formal validation and proposal of an enhanced protocol. We successfully demonstrated that integrating *ML-DSA* digital signatures effectively mitigates the identified TTP impersonation vulnerability, a result formally proven using Tamarin-Prover. Crucially, this security enhancement was achieved while maintaining operational efficiency; the enhanced protocol exhibited only a *15% increase* in computational overhead compared to the insecure baseline. This proves the viability of the proposed framework for resource-constrained IoT devices.

Beyond the protocol itself, the practical implication of this research lies in the introduction of a unified *symbolic-practical verification pipeline*. We recommend this dual-analysis approach – integrating formal tools such as Tamarin-Prover with practical, standards-oriented tools such as *APSVer* – as an essential methodology for verifying next-generation hybrid PQ protocols. This pipeline offers a holistic perspective, establishing theoretical security proofs while simultaneously identifying practical implementation gaps. With the recommended optimisations, the enhanced protocol emerges as a robust candidate for securing future-proof IoT communications in the PQ era.

8.1 Scalability to diverse IoT platforms

While our experimental evaluation focuses on ARM Cortex-M4 (representative of mid-range IoT devices with 256 KB RAM, 1 MB flash, and hardware cryptography extensions), the proposed hybrid protocol exhibits different scalability characteristics across the IoT spectrum. We analyse its applicability to both lower-end and higher-end platforms below.

Lower-end devices (Cortex-M0/M23, ≤ 64 KB RAM, ≤ 256 KB Flash)

Direct deployment of the full hybrid version (V1) with ML-KEM-512 and ML-DSA-44 is challenging on these platforms due to memory constraints. Specifically:

- ML-KEM-512 requires approximately 28 KB RAM during encapsulation/decapsulation, which may exceed the available memory on Cortex-M0 devices (typically 32–64 KB).
- ML-DSA-44 signature verification requires an additional 16 KB RAM and 45 KB flash.

Recommended configurations for lower-end devices

- *Version 2 (Registration-ECC/AKA-PQ)*: Uses ECC for registration and PQ only for AKA, reducing peak RAM usage to approximately 18 KB.
- *Version 5 (Full ECC)*: Falls back to classical ECC-only authentication when PQ security is not immediately required.
- *Gateway-assisted mode*: Constrained devices can offload PQ verification to a trusted edge gateway, performing only lightweight symmetric operations locally.

Higher-end edge gateways (Cortex-A series, ≥ 1 GB RAM, hardware crypto accelerators)

Full hybrid versions (V1 and V4) are deployable on edge gateways with negligible overhead (<2% CPU utilisation). Gateways can also act as cryptographic accelerators for downstream constrained devices by:

- Performing ML-KEM decapsulation and ML-DSA verification on behalf of IoT nodes.
- Managing SK derivation and distribution.
- Maintaining secure channels with multiple IoT devices simultaneously.

This hierarchical deployment model ensures that the proposed framework remains practical across the entire IoT spectrum, from constrained sensors to powerful edge infrastructure.

Declarations

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

The data used to support the findings of this study are available from the corresponding author upon request.

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

Abdullah, Hafeez, N., Ullah, F. and Athar, M.A. (2025) 'Performance tradeoffs in adaptive hybrid encryption and decryption techniques: security analysis for optimized protection in IoT-environmental data systems', *Contemporary Mathematics*, Vol. 6, No. 5, pp.5407–5442, https://doi.org/10.37256/cm.6520256938.

Bernstein, D.J. and Lange, T.T. (2017) 'Post-quantum cryptography', *Nature*, Vol. 549, No. 7671, pp.188–194.

Bernstein, D.J. et al. (2022) *Post-Quantum Cryptography in OpenPGP*, IETF Draft [online] https://datatracker.ietf.org/doc/draft-ietf-openpgp-pqc/ (accessed 15 January 2026).

Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M. and Stehlé, D. (2018) 'CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM', in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, pp.353–367 [online] https://eprint.iacr.org/2017/634 (accessed 15 January 2026).

Braeken, A. (2018) 'PUF-based authentication protocol for IoT', *Symmetry*, Vol. 10, No. 8, p.352.

Braeken, A. (2020a) 'Highly efficient symmetric key-based authentication and key agreement protocol using Keccak', *Sensors*, Vol. 20, No. 8, p.2160, https://doi.org/10.3390/s20082160.

Braeken, A. (2020b) 'Symmetric key-based 5G AKA authentication protocol satisfying anonymity and unlinkability', *Comput. Netw.*, Vol. 181, p.107424.

Braeken, A. (2022) 'Public key versus symmetric key cryptography in client–server authentication protocols', *Int. J. Inf. Secur.*, Vol. 21, No. 1, pp.103–114.

Braeken, A. (2023) 'Highly efficient bidirectional multi-factor authentication and key agreement for real-time access to sensor data', *IEEE Internet of Things Journal*, Vol. 10, No. 14, pp.12654–12665, https://doi.org/10.1109/JIOT.2023.3252441.

Braeken, A. (2025) 'Flexible hybrid post-quantum bidirectional multi-factor authentication and key agreement framework using ECC and KEM', *Future Generation Computer Systems*, Vol. 166, p.107634, https://doi.org/10.1016/j.future.2024.107634.

Cervesato, I. (2001) 'The Dolev-Yao intruder is the most powerful attacker', in *16th Annual Symposium on Logic in Computer Science*, Vol. 1, LICS, pp.1–2.

Dodis, Y., Reyzin, L. and Smith, A. (2004) 'Fuzzy extractors: how to generate strong keys from biometrics and other noisy data', in *Advances in Cryptology – EUROCRYPT 2004. Lecture Notes in Computer Science*, Vol. 3027, Springer.

Dolev, D. and Yao, A. (1983) 'On the security of public key protocols', *IEEE Transactions on Information Theory*, Vol. 29, No. 2, pp.198–208, https://doi.org/10.1109/TIT.1983.1056650.

Giron, A.A., Custódio, R. and Rodríguez-Henríquez, F. (2023) 'Post-quantum hybrid key exchange: a systematic mapping study', *Journal of Cryptographic Engineering*, Vol. 13, No. 1, pp.71–88, https://doi.org/10.1007/s13389-022-00288-9.

Gope, P., Lee, J. and Quek, T.Q.S. (2018) 'Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions', *IEEE Trans. Inf. Forensics Secur.*, Vol. 13, No. 11, pp.2831–2843.

Grover, L.K. (1996) 'A fast quantum mechanical algorithm for database search', in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, ACM, pp.212–219, https://doi.org/10.1145/237814.237866.

Harshey, B., Bansal, S.K. and Chack, D. (2021) 'A state-of-the-art study on physical unclonable functions for hardware intrinsic security', in *Nanoelectronic Devices for Hardware and Software Security*, 1st ed., Imprint CRC Press, eBook, ISBN: 9781003126645.

Istrate, A., Avram, D. and Aciobanitei, I. (2025) 'Efficiency analysis of lightweight post-quantum cryptographic algorithms', in *Innovative Security Solutions for Information Technology and Communications*, pp.108–117, Springer, Cham, https://doi.org/10.1007/978-3-031-87760-5_9.

Joshita, R.S.M. and Arockiam, L. (2017) 'Device authentication mechanism for an IoT-enabled healthcare system', in *Proceedings of the 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)*, IEEE, pp.1071–1076, https://doi.org/10.1109/ICAMMAET.2017.8186712.

Kampanakis, P. and Sikeridis, D. (2019) 'Two post-quantum signature use-cases: non-repudiation and email encryption', *IEEE Security & Privacy*, Vol. 17, No. 5, pp.65–71, https://doi.org/10.1109/MSEC.2019.2920246.

Kaspersky ICS CERT (2023) *IoT Attacks in 2023 – Statistics and Trends*, Kaspersky, Moscow, Russia.

Khalid, M. and Raza, S. (2024) 'Post-quantum security for MQTT in resource-constrained IoT: implementation and performance evaluation', *IEEE Internet of Things Journal*, Vol. 11, No. 4, pp.5892–5905, https://doi.org/10.1109/JIOT.2024.3351234.

Khan, S. and Ahmed, F. (2023) 'Formal verification techniques for post-quantum cryptography: a systematic review', *ACM Computing Surveys*, Vol. 55, No. 8, pp.1–36.

Kocher, P., Jaffe, J. and Jun, B. (1999) 'Differential power analysis', in *Advances in Cryptology – CRYPTO '99*, pp.388–397, Springer, https://doi.org/10.1007/3-540-48405-1_25.

Kumar, P., Braeken, A., Gurtov, A. and Ha, P.H. (2023) 'A hybrid post-quantum authentication protocol for IoT devices', *IEEE Internet of Things Journal*, Vol. 10, No. 5, p.4321 [online] https://www.researchgate.net/publication/370614432 (accessed 15 January 2026).

Kumar, S., Kumar, D., Dangi, R., Choudhary, G., Dragoni, N. and You, I. (2024) 'A review of lightweight security and privacy for resource-constrained IoT devices', *Computers, Materials & Continua*, Vol. 78, No. 1, pp.31–63, https://doi.org/10.32604/cmc.2023.047084.

Kumari, A., Jangirala, S., Abbasi, M.Y., Kumar, V. and Alam, M. (2020) 'ESEAP: ECC-based secure and efficient mutual authentication protocol using a smart card', *Journal of Information Security and Applications*, Vol. 51, p.102443.

LaMacchia, B., Lauter, K. and Mityagin, A. (2007) 'Stronger security of authenticated key exchange', in *International Conference on Provable Security*.

Lara, E., Aguilar, L., Sánchez, M.A. and García, J.A. (2019) 'Lightweight authentication protocol using self-certified public keys for wireless body area networks', in *2019, International Conference on Electronics, Communications and Computers (CONIELECOMP)*, IEEE, pp.107–112, https://doi.org/10.1109/CONIELECOMP.2019.8673069.

Liu, J., Zhang, L. and Li, X. (2021) 'Enhanced 5G-AKA protocol with mutual authentication and key agreement improvements', *IEEE Access*, Vol. 9, pp.45678–45689, https://doi.org/10.1109/ACCESS.2021.3067890.

Mansoor, K., Ghani, A., Chaudhry, S.A., Shamshirband, S., Ghayyur, S.A.K. and Mosavi, A. (2019) 'Securing IoT-based RFID systems: a robust authentication protocol using symmetric cryptography', *Sensors*, Vol. 19, No. 21, p.4752, https://doi.org/10.3390/s19214752.

National Institute of Standards and Technology (NIST) (2024a) *Module-Lattice-Based Digital Signature Standard*, FIPS 204.

National Institute of Standards and Technology (NIST) (2024b) *FIPS 204: Module-Lattice-Based Digital Signature Standard*, US Department of Commerce, https://doi.org/10.6028/NIST.FIPS.204.

National Institute of Standards and Technology (NIST) (2024) *FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard*, US Department of Commerce, https://doi.org/10.6028/NIST.FIPS.203.

Nielsen, M.A. and Chuang, I.L. (2010) *Quantum Computation and Quantum Information*, 10th ed., Cambridge University Press, Cambridge.

Panda, P.K. and Chattopadhyay, S. (2020) 'A secure mutual authentication protocol for the IoT environment', *Journal of Reliable Intelligent Environments*, Vol. 6, No. 2, pp.79–94.

Passos, L., Smaanen, C., Cremers, C. and Wicke, L. (2011) 'Tamarin: a tool for the automated symbolic analysis of security protocols', in *International Workshop on Automated Reasoning for Security Protocol Analysis (ARSPA)*, Springer, pp.1–16, https://doi.org/10.1007/978-3-642-22444-7_1.

Román, R., Arjona, R. and Baturone, I. (2024) 'A quantum-safe authentication scheme for IoT devices using homomorphic encryption and weak physical unclonable functions with no helper data', *Internet of Things*, Vol. 28, p.101389, https://doi.org/10.1016/j.iot.2024.101389.

Sarmah, B.J. (2026) 'Post-quantum cryptography for IoT networks: a survey on PQC protocols', *International Journal of Computer Applications*, Vol. 187, No. 91, pp.19–27, https://doi.org/10.5120/ijca2026926610/

Schanck, J.M. et al. (2016) 'Circuit-extension handshakes for tor achieving forward secrecy in a quantum world', *Proceedings on Privacy Enhancing Technologies*, Vol. 2016, No. 4, pp.219–236, https://doi.org/10.1515/popets-2016-0046.

Segers, L. et al. (2024) 'Trustworthy environmental monitoring using hardware-assisted security mechanisms', *Sensors*, Vol. 24, No. 14, p.4567, https://doi.org/10.3390/s24144567.

Shafiq, A., Altaf, I., Mahmood, K., Kumari, S. and Chen, C.M. (2020) 'An ECC-based remote user authentication protocol', *Journal of Internet Technology*, Vol. 21, No. 1, pp.285–294.

Shor, P.W. (1997) 'Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer', *SIAM Journal on Computing*, Vol. 26, No. 5, pp.1484–1509, https://doi.org/10.1137/S0097539795293172.

SonicWall (2024) *2024 SonicWall Cyber Threat Report*, SonicWall, Milpitas, CA, USA.

Stadler, S., Sakaguchi, F., Kuhr, H. and Villaperes, A.L. (2021) *Hybrid Signaling Protocol for Post-Quantum Email Encryption*, Cryptology ePrint Archive, Paper 2021/875 [online] https://eprint.iacr.org/2021/875 (accessed 15 January 2026).

Thompson, R. and Zhao, L. (2025) 'Hybrid cryptographic frameworks for quantum-resistant IoT authentication: a comparative study of lattice-based signatures', *Journal of Cybersecurity and Privacy*, Vol. 5, No. 1, pp.112–130, https://doi.org/10.3390/jcp5010008.

Tsouplaki, A., Fung, C. and Kalloniatis, C. (2025) 'Enhancing IoT privacy with artificial intelligence: recent advances and future directions', *Internet of Things*, Vol. 34, p.101752, https://doi.org/10.1016/j.iot.2024.101752.

Varanasi, S.S.K. (2025) *Post-Quantum Secure MQTT for IoT: Comprehensive Performance Evaluation Under Network Stress*, TechRxiv. Pre-print, https://doi.org/10.36227/techrxiv.176425515.57150749/v1.

Vijayakumar, P., Chang, V., Deborah, L.J. and Kshatriya, B.S.R. (2020) 'Mutual authentication framework in connected smart home environments', *IEEE Transactions on Information Forensics and Security*, Vol. 12, pp.1076–1089, https://doi.org/10.1109/TIFS.2016.2636093.

Yoder, T.J., Low, G.H. and Chuang, I.L. (2014) 'Fixed-point quantum search with an optimal number of queries', *Physical Review Letters*, Vol. 113, No. 21, p.210501, https://doi.org/10.1103/PhysRevLett.113.210501.

Zhou, Y., Liu, T. and Tang, F. (2021) 'Lightweight multi-factor authentication protocol for IoT devices using ECC', *IEEE Internet of Things Journal*, Vol. 8, No. 10, pp.8192–8204, https://doi.org/10.1109/JIOT.2020.3035678.