Journal of Information Systems Engineering and Management
2025, 10(42s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Post-Quantum Lightweight Key Sharing Protocol for Secure MQTT-Based IoT Networks

Sharadadevi Kaganurmath[1], Nagaraj Cholli[2] and Anala M R[3]
[1] *Assistant Professor, Department of Computer Science (AIML), Global Academy of Technology, Bengaluru-INDIA*;sharadask@gmail.com
2Professor and Dean CMR University, Bengaluru-INDIA;Nagaraj.cholli@gmail.com
3Professor, Department of Information Science Engineering, RV College of Engineering, Bengaluru-INDIA; analamr@rvce.edu.in

ARTICLE INFO

Received: 31 Dec 2024

Revised: 20 Feb 2025

Accepted: 28 Feb 2025

ABSTRACT

The Post-Quantum Lightweight Key Sharing Protocol for Secure MQTT-Based IoT Networks (PQLKS-MQTT) addresses the critical need for quantum-resistant and resource-efficient security in IoT communications. As the proliferation of IoT devices continues, securing MQTT-based networks against evolving threats, including quantum attacks, becomes imperative. PQLKS-MQTT integrates the Kyber Key Encapsulation Mechanism for post-quantum key exchanges, along with BLAKE2s hashing and ChaCha20 encryption, to ensure robust security with minimal resource consumption. Implemented using the Cooja simulator with Contiki OS, Eclipse Mosquitto MQTT broker, and Open Quantum Safe (liboqs) library, the protocol demonstrates superior performance compared to state-of-the-art solutions. Experimental results show that PQLKS-MQTT achieves the lowest CPU energy consumption (0.0000021 mJ), fastest execution time (0.35 seconds), and minimal computational (260 CPU cycles) and communication overheads (55 bytes), with only a slight increase in average energy consumption (0.00145 mJ) due to post-quantum cryptographic operations. This balance between enhanced security and efficient resource utilization makes PQLKS-MQTT a suitable solution for resource-constrained IoT devices and large-scale deployments, offering a scalable, quantum-safe communication framework for future IoT ecosystems.

Keywords: Post-Quantum Cryptography, Kyber Key Encapsulation Mechanism, MQTT Security, Resource-Constrained IoT Devices, Quantum-Resilient IoT Networks.

INTRODUCTION

The IoT has evolved to become a revolutionary technology that enables devices to exchange digital information effectively within healthcare, residences and industrial facilities and transportation networks (Chataut et al., 2023). A projection estimates that IoT device numbers will exceed 32.1 billion by the year 2030 (Duguma & Bai 2024). The extraordinary speed of IoT implementation requires organizations to implement secure methods that enable device communication protection. MQTT (Message Queuing Telemetry Transport) has become the de facto messaging protocol for IoT due to its lightweight architecture, low bandwidth requirements, and suitability for resource-constrained devices (Bhardwaj et al., 2023). However, MQTT's inherent design, which prioritizes simplicity and low overhead, exposes it to significant security vulnerabilities, including data interception, unauthorized access, and denial-of-service attacks (Li et al., 2024). These vulnerabilities are exacerbated in large-scale IoT deployments where resource constraints limit the feasibility of traditional cryptographic solutions.

IoT security faces its most serious challenge due to the emerging quantum computing technology (Goyal et al., 2024). Using Shor's algorithm and other quantum methods enables fast resolution of integer factorization and discrete logarithm problems which makes traditional encryption schemes (RSA, ECC) incompatible after their release (Ugwuishiwu et al., 2022; Gitonga 2025). Post-Quantum Cryptographic (PQC) solutions must be developed because they need to protect against quantum attacks yet perform efficiently on resource-constrained IoT devices (Mansoor et al., 2025). Existing lightweight security protocols for MQTT-based IoT networks, such as DLKS-MQTT and DLA-MQTT, offer reduced latency and computational overhead but remain vulnerable to quantum adversaries due to their reliance on classical cryptographic techniques. Additionally, these protocols often incur higher resource consumption or lack scalability, making them unsuitable for the growing complexity of IoT ecosystems.

The research initiates from the critical requirement to defend MQTT-based IoT communications against classic and quantum attacks without neglecting resource efficiency. Current post-quantum cryptographic solutions, though secure, often introduce significant computational and memory overheads, making them impractical for IoT applications. The proposed PQLKS-MQTT addresses these limitations by integrating Kyber, a lattice-based Key Encapsulation Mechanism (KEM) recognized by NIST for its post-quantum security and computational efficiency. PQLKS-MQTT leverages Kyber for secure key exchanges, BLAKE2s for hashing, and ChaCha20 for encryption, ensuring a balance between quantum-safe security and lightweight operations suitable for IoT environments.

The research establishes PQLKS-MQTT to improve key-sharing speed and diminish latency alongside reducing computational load for MQTT-based IoT systems. The research produced a post-quantum key sharing protocol specifically designed for IoT devices which constitutes its main achievements. The protocol delivers an extensive performance evaluation which shows superior results compared to traditional protocols. The research study presents both secure additional scalability features and implementation methods for IoT application communication protocols. Summarizing the above literature, this work defines the objectives that is required to be addressed

- To design 256 bits key encryption algorithm using post-quantum cryptographic technique to resist quantum-based attacks.
- To integrate the post-quantum encryption algorithm with MQTT protocol to secure communication in IoT environment.
- To induce security threats and test the performance of post-quantum encryption algorithm.

RELATED WORKS

In [10] the authors proposed a secure data exchange protocol for cryptographic key generation, renewal, and distribution (KGRD) using the Trusted Platform Module (TPM) 1.0 and MQTT service. Later, Furtak, et.al, introduced a KGRD system using TPM 2.0 hardware modules to enhance IoT security through trusted key management [16]. An improved lightweight authentication and key agreement protocol for IoT-enabled smart grids, addressing both insider and outsider threats while maintaining computational efficiency was proposed in [11].

Winarno et. al developed a block cipher-based lightweight encryption scheme for MQTT, testing algorithms like Tiny JAMBU and Advanced Encryption Standard (AES)-128 Galois/Counter Mode (GCM) on ARM and ESP8266 processors, with Tiny JAMBU proving the most efficient [12]. Kanwal et al. presented a noncommutative key exchange protocol for IoT, enhancing security through polynomials over noncommutative rings [13]. Oller, at.al leveraged Elliptic Curve Cryptography (ECC) for secure authentication in multi-user IoT environments, validated by Automated Validation of Internet Security Protocols and Applications (AVISPA) and Burrows–Abadi–Needham (BAN) logic tools [14]. Liu et al. introduced Quantum-based Secure and Lightweight Transmission (QSLT), a quantum-based lightweight transmission mechanism that reduces power usage by 58.77% and ensures secure data transmission with minimal delay [15].

Al Ahmed et al. introduced Authentication-Chains, a blockchain-inspired lightweight authentication protocol that uses proof of identity authentication, clustering, and blockchain for IoT device authentication. Ali and Mathew (2022) presented a no-share key exchange algorithm for hybrid IoT applications with blockchain, validated through AVISPA [17]. Nabavirazavi, et.al [18] developed Lightweight Authenticated Key Exchange (LAKEE), which uses Elliptic Curve Cryptography (ECC) for session key generation, minimizing communication and computational overhead. Shanmugapriya, et.al [19] proposed an energy-efficient trust-based secured lightweight authentication protocol that optimizes energy consumption by selecting trusted nodes during data dissemination. Yuan et al. [23] introduced a post-quantum blockchain architecture for IoT over NTRU lattice, reducing transaction sizes significantly and ensuring quantum-resistant security. Assaig et al. [37] developed a lightweight IoT security system that MQTT messages with a latency of 0.3 seconds and power consumption of 1.683 millijoules (mJ), addressing challenges such as low computation power and limited storage. Hosseinzadeh et al. [21] conducted an independent security analysis of an IoT authentication protocol, revealing vulnerabilities like lack of perfect forward secrecy and susceptibility to insider attacks, and proposed an enhanced protocol with only 15.5% additional computational cost. Mhaibes et al. [22] modified the Tiny Encryption Algorithm (TEA) for IoT devices using Linear Feedback Shift Registers (LFSRs), improving key sensitivity, avalanche effect, and completeness by over 50% compared to the original TEA.

Ahmed et al. [20] introduced a provably secure cybersecurity mechanism (ELCA) integrating Elliptic Curve Diffie–Hellman (ECDH) for key distribution, reducing CPU execution time by 50%, storage cost by up to 32%, and energy consumption by 41%. Nouma, et.al [38] proposed Hardware-ASsisted Efficient Signature (HASES) schemes for post-quantum digital signatures in IoT, achieving superior efficiency over NIST PQC standards. Edwards et al. [30] evaluated Quantum Key Distribution (QKD) for securing satellite-integrated IoT networks, highlighting technical challenges like low secret key generation rates and equipment vulnerabilities. Bharathi et.al [24] proposed a lightweight PRESENT block cipher model implemented on FPGA (Field-Programmable Gate Array), achieving high efficiency at 412.4 MHz with low power consumption of 192 mW and minimal chip area usage, enhancing IoT security. A lightweight integrity-preserving scheme for cloud-based IoT data exchange using XOR and cryptographic hash functions, ensuring integrity, privacy, and low computational costs was introduced [25].

Hindumathi et.al [26] developed a hybrid combinatorial design-based key distribution method for IoT, improving connectivity, resilience, and scalability despite key storage challenges in larger networks. Hussein at.al [27] demonstrated MQTT-based attacks on ZigBee networks, exposing vulnerabilities in IoT home automation systems and proposing Intrusion Detection Systems (IDSs) for mitigation. A hybrid lightweight security approach using PRESENT and Tiny Encryption Algorithm (TEA) with Elliptic Curve Cryptography (ECC), enhancing data security in Internet of Healthcare Things (IoHT) applications with reduced packet loss and improved performance was developed [28]. Alluhaidan, et.al [39] proposed a lightweight cryptographic process for resource-constrained IoT devices using a symmetrical encryption key block and modified Feistel architecture, offering energy-efficient and secure communication while minimizing processing cycles.

Malik et al. [34] introduced L-ECQV, a lightweight Elliptic Curve Qu Vanstone (ECQV) implicit certificate for authenticated key exchange in the Ephemeral Diffie–Hellman Over COSE (EDHOC) protocol at the application layer, reducing energy consumption by 27% and message overhead by 52% in Contiki OS. Said et al. [31] proposed a Secure Aggregation and Transmission Scheme (SATS) for IoT-enabled Wireless Medical Sensor Networks (WMSN), utilizing XOR operations for batch key creation and achieving reduced computation and communication costs. Babu et al. [35] conducted a survey on quantum-secure authentication and key agreement protocols, emphasizing the need for lightweight post-quantum cryptographic primitives. Cherkaoui et al. [32] presented a hybrid post-quantum encryption design for MQTT communications on embedded devices, ensuring optimal performance while addressing computational constraints. Saqib and Moon (2024) developed a lightweight multi-factor authentication scheme for MQTT-based IoT applications using ECC, providing mutual authentication and forward secrecy. Minhas et.al [36] proposed a PQ Edge Server (PQES) for offloading post-quantum cryptographic operations, significantly reducing RAM and CPU usage for IoT devices. Choi and Lee et.al [29] integrated Nth Degree Truncated Polynomial Ring Units (NTRUs) with a Secure and Scalable IoT (SSI) network, achieving a 161% speed improvement over RSA at higher security levels, ensuring post-quantum security in IoT communications.

METHODOLOGY

This section discusses system architecture and algorithm that are developed along with experimental results.

SYSTEM ARCHITECTURE

The PQLKS-MQTT system implements architecture that fulfils security requirements for MQTT-based IoT messages beside accommodating the resource limitations of IoT equipment. Figure 1 depicts the architecture of MitM attack and process flow diagram of post-quantum cryptography authentication. From figure 1 (a), as a middleman, threat pretends to be a genuine party. He disconnects the quantum channel, reconnects again with genuine parties and carries out the man-in-the-middle attack. Figure 1(b) highlights a quantum channel (|ψ|) is used to exchange the information between QKD transmitter and QKD receiver, Further the data processing is carried out through classical channel by exchanging classical messages (M) between them. To authenticate the classical messages, Broker and subscriber is available to generate a digest using a hash function (H). Further, the subscriber encrypts (E) this digest with a pre-shared key (K) or subscriber private key (SB) and subsequently sends to Broker. After receiving this information, in the next step Broker decrypts (D) it with the same pre-shared key (K) or subscriber's public key (PB) and check and compares (C) the result with her own digest. If both the information is same, the authentication between them remains successful or otherwise the authentication fails. This work implements a two-way authentication between both broker and subscriber. From figure 1 (c), both broker and subscriber exchange their own certificate (CA and CB) along with random nonce (RA,RB) with each other. Finally, they use public key of certificate (Pr) to validate other public key belongs to its identity. The PQC algorithm is used to sign message digest (DA, DB) and nonce under their own private keys (SA, SB) to generate signatures (TA, TB). Finally, only the genuine party has the corresponding private key, then it can confirmed that the message is signed legally.

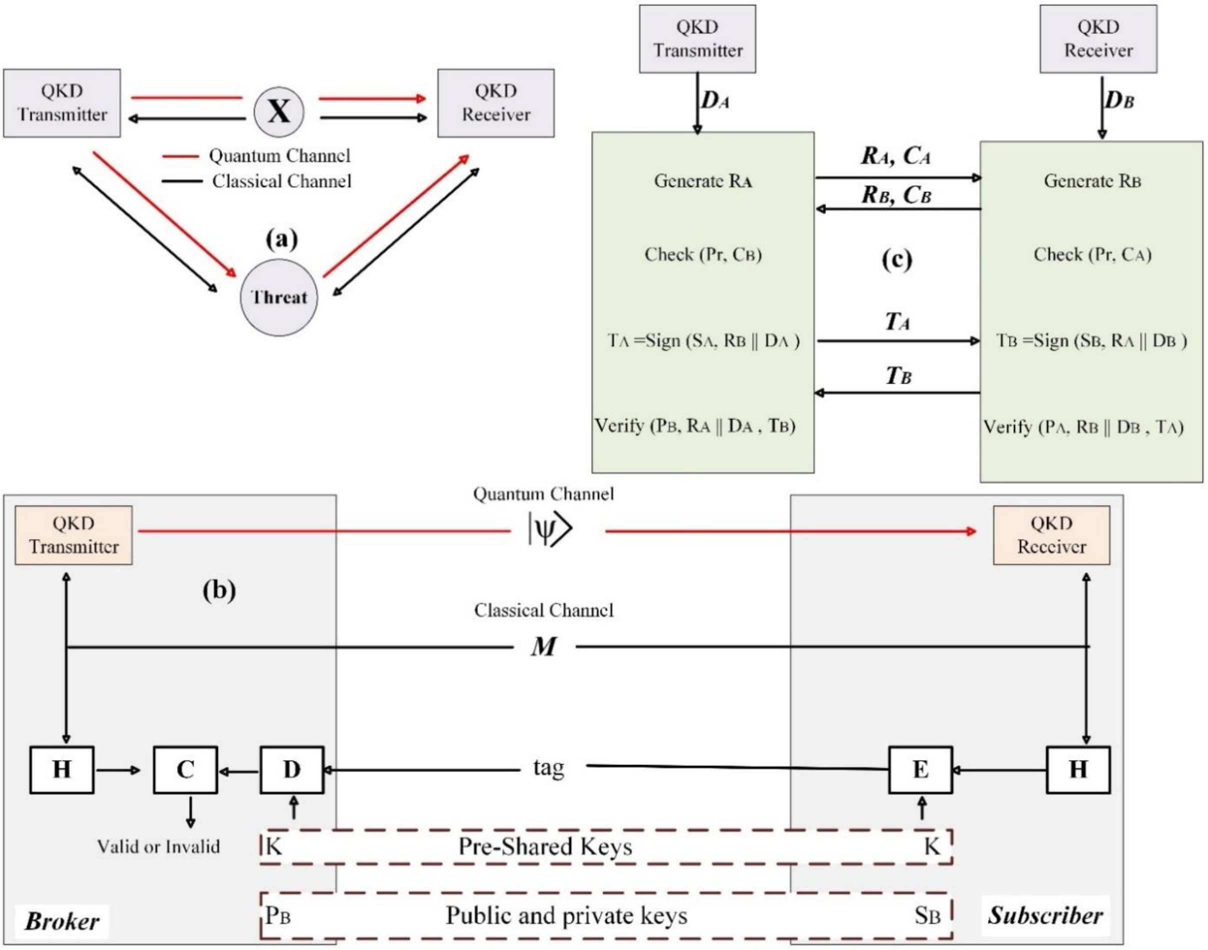

Figure 1: Proposed PQLKS-MQTT Architecture || denotes concatenate between strings

Communication with the MQTT broker is established through the MQTT protocol, specifically designed for low-bandwidth, high-latency networks. Given the limited computational power of these devices, the integration of lattice-based post-quantum cryptography is optimized by offloading complex key generation processes to the key management server, thereby minimizing the computational burden on the devices themselves. Each device stores minimal cryptographic material, such as pre-shared keys and public parameters for the Kyber KEM, ensuring that the memory footprint remains within acceptable limits while maintaining robust security.As the central communication hub, the MQTT broker handles the publish-subscribe model operations which form the basis of MQTT communication. The broker is deployed on a server-class machine with higher computational capabilities (such as a multi-core processor with clock speeds above 2 GHz and memory capacity exceeding 8 GB RAM), allowing it to efficiently handle numerous simultaneous connections and perform cryptographic operations without introducing significant latency. The broker manages secure key exchanges through PQLKS-MQTT platform which pairs IoT devices with key management servers that apply post-quantum cryptographic keys for each encrypted communication session. The broker implements light encryption operations which include BLAKE2s hashing alongside ChaCha20 encryption for message verification to protect data exchanges while maintaining low processor requirements for IoT devices.

The key management server is the cornerstone of the PQLKS-MQTT architecture, tasked with generating, distributing, and managing cryptographic keys using lattice-based post-quantum cryptography. The KMS operates on a dedicated high-performance server equipped with a minimum of 16 GB RAM and multi-core processors, enabling it to perform complex polynomial operations intrinsic to the Kyber algorithm. Lattice-based cryptography is integrated into the architecture through the Kyber KEM, which provides quantum-resistant key encapsulation and decapsulation processes. The KMS initializes by generating public parameters for the lattice structure, including polynomial modulus q and dimension n, which are securely distributed to all IoT devices and the MQTT broker. The KMS applies broker public key encryption to session keys during key exchange to ensure the correct recipient can decrypt and utilize them for secured information transmission. The integrated security approach protects PQLKS-MQTT against quantum technology threats by making cryptographic foundations unbreakable even during the existence of quantum computing.

The initial step of data movement inside PQLKS-MQTT starts when IoT devices need secure connections that the MQTT broker relays to a key management server. The broker sends such requests to a key management server which uses Kyber algorithms to create secure keys. The encapsulated keys are transmitted back to the broker, which then distributes them to the respective devices, establishing secure communication channels. Each data packet transmitted thereafter is hashed using BLAKE2s and encrypted using ChaCha20, ensuring data integrity, confidentiality, and authenticity. The architectural choice to integrate lattice-based cryptography at the KMS level while employing lightweight cryptographic operations at the device level is justified by the need to balance high security with resource efficiency. This architecture ensures that IoT devices remain lightweight and power-efficient while benefiting from quantum-safe cryptographic operations managed by the more capable KMS and broker.

POST-QUANTUM CRYPTOGRAPHIC INTEGRATION

PQLKS-MQTT features Kyber as its lattice-based KEM for secure key exchanges which relies on the lattice problems Shortest Vector Problem (SVP) and Learning With Errors (LWE). Kyber operates over the polynomial ring $R_q = Z_q[x]/(x^n + 1)$, where n is chosen as a power of two for efficient polynomial multiplication using Number Theoretic Transform (NTT), and q is a prime modulus ensuring secure arithmetic operations. In PQLKS-MQTT, the key exchange process begins with the initialization of system parameters where all devices and the MQTT broker share public parameters, including the dimension $n = 256$ and modulus $q = 3329$.

During key generation, each IoT device samples its secret vector $s \in R_q^n$ and error vector $e \in R_q^n$ from a discrete Gaussian distribution χ, ensuring that each coefficient follows a Gaussian distribution centered at zero with standard deviation σ. A public matrix $A \in R_q^{n \times n}$ is generated using a cryptographic pseudo-random function (PRF) seeded with a public parameter. The device computes its public key $pk = A \cdot s + e \, mod \, q$ and keeps the private key $sk = s$ secure. The MQTT broker, upon initiating a key exchange, generates a random vector $r \in R_q^n$ and error terms $e_1 \in R_q^n$ and $e_2 \in R_q$ from the same distribution χ. The ciphertext c is formed as $c_1 = A \cdot r + e_1 \, mod \, q$ and $c_2 = pk \cdot r + e_2 + m \, mod \, q$, where m is the session key encoded as a polynomial in R_q.

The device, upon receiving the ciphertext (c_1, c_2), recovers the session key by computing $m' = c_2 - s \cdot c_1 \, mod \, q$. The correctness of this operation is ensured by the fact that $c_2 - s \cdot c_1 = (pk \cdot r + e_2 + m) - s \cdot (A \cdot r + e_1) \, mod \, q$, simplifying to $m + e_2 - s \cdot e_1 mod \, q$, where the noise terms e_2 and $s \cdot e_1$ are small enough to be correctly reconciled. The security of this key exchange is guaranteed by the hardness of the Module-LWE problem, which posits that given $(A, A \cdot s + e)$, it is computationally infeasible for an adversary to recover s, even with quantum computational capabilities.

Noise distribution plays a critical role, with each polynomial coefficient sampled from χ ensuring that the noise is large enough to obscure the secret key but small enough to allow correct decryption. The integration of Kyber in PQLKS-MQTT ensures that each key exchange session is secure against both classical and quantum adversaries, providing a robust foundation for MQTT-based IoT communications.

```
Algorithm 1: Kyber Key Generation
Input: Security parameters (n, q)
Output: Public key (pk), Private key (sk)
procedure KEYGEN()
    A ← GeneratePublicMatrix(n, q)  // Public matrix A
    s, e ← SampleNoiseVectors(n, q)  // Gaussian noise
    pk ← (A * s + e) mod q          // Public key
    sk ← s                          // Private key
    return pk, sk
end procedure
```

```
Algorithm 2: Kyber Key Encapsulation (MQTT Broker)
Input: Public key (pk), Message (m)
Output: Ciphertext (c1, c2)
procedure ENCAPSULATE(pk, m)
    r, e1, e2 ← SampleNoiseVectors()    // Noise vectors
    c1 ← (A * r + e1) mod q             // First ciphertext part
    c2 ← (pk * r + e2 + m) mod q        // Second ciphertext part
    return (c1, c2)
end procedure
```

```
Algorithm 3: Kyber Key Decapsulation
Input: Private key (sk), Ciphertext (c1, c2)
Output: Recovered message (m')
procedure DECAPSULATE(sk, c1, c2)
    m' ← (c2 - sk * c1) mod q  // Recover message
    return m'
end procedure
```

PRE-SHARED KEY DISTRIBUTION FRAMEWORK

The Pre-Shared Key (PSK) Distribution Framework in PQLKS-MQTT is designed to provide secure and efficient key management for MQTT-based IoT networks. The framework ensures that resource-constrained IoT devices can participate in secure communications without incurring high computational costs, by leveraging pre-provisioned master keys and deriving session keys through a lightweight Key Derivation Function (KDF). Master key provisioning occurs during the initial device deployment phase, where each IoT device D_i is assigned a unique master key MK_i from the key management server (KMS). This master key $MK_i \in \{0,1\}^{256}$ is securely embedded into the device's non-volatile memory, ensuring that the key remains protected from unauthorized access.

Session key derivation in PQLKS-MQTT utilizes a KDF that combines the master key with dynamic nonces to generate ephemeral session keys. The KDF is defined as $KDF: \{0,1\}^{256} \times \{0,1\}^{128} \rightarrow \{0,1\}^{256}$, where the input consists of a 256-bit master key and a 128-bit nonce N, and the output is a 256-bit session key SK. Mathematically, the session key derivation process is expressed as $SK = H(MK \oplus N)$, where $\oplus$ denotes the XOR operation and H is a cryptographic hash function, specifically BLAKE2s, chosen for its efficiency and strong security guarantees. The XOR operation ensures that even a single-bit change in the nonce produces a significantly different session key, providing forward secrecy. The use of BLAKE2s ensures that the derived session key is uniformly distributed, resistant to collision, and computationally infeasible to reverse-engineer.

Periodic key renewal is implemented through a time-based mechanism where session keys are refreshed at fixed intervals T. Each device and the broker synchronize using a secure time source, and at each interval t, a new session key SK_t is derived as $SK_t = H(MK \oplus N_t)$, where N_t is a nonce derived from the current timestamp T_t. This periodic renewal ensures that even if a session key is compromised, its validity is limited to a single time interval, thereby mitigating the impact of key compromise. The framework's efficiency is maintained by minimizing the computational overhead on IoT devices. XOR operations are inherently lightweight, requiring minimal CPU cycles, and BLAKE2s is optimized for low-power environments, making it ideal for IoT applications. The pseudocode for the session key derivation process is as follows:

```
Algorithm 4: Session Key Derivation in PQLKS-MQTT
Input: Master Key (MK), Nonce (N)
Output: Session Key (SK)
procedure DERIVE_SESSION_KEY(MK, N)
    temp_key ← MK ⊕ N
    SK ← BLAKE2s(temp_key)
    return SK
end procedure
```

The above algorithm demonstrates the simplicity and efficiency of the key derivation process. The security measures embedded in the framework include the use of unique master keys for each device, ensuring that compromise of one device does not affect others. The periodic renewal mechanism ensures that even if an attacker obtains a session key, its usefulness is limited in time. Additionally, the cryptographic operations are chosen to balance security and efficiency, ensuring that IoT devices with limited computational resources can still maintain secure communications within the MQTT network.

LIGHTWEIGHT CRYPTOGRAPHIC OPERATIONS

The PQLKS-MQTT protocol equips BLAKE2s for hashing and ChaCha20 for encryption operations to secure messages and achieve low CPU usage that benefits IoT devices with limited resources. BLAKE2s functions as a cryptographic hash utility which provides contemporary hashing security standards through enhanced performance capabilities. The BLAKE2s hash function accepts input messages with any length M and generates a static output $H \in \{0,1\}^{256}$. The core of BLAKE2s is its compression function F, which takes as input a 512-bit state, a 256-bit message block, and two 32-bit counters representing the current position in the message and the total length hashed so far. The input-output relationship is defined as $H = BLAKE2s(M, k)$, where k is an optional key used in keyed hashing. BLAKE2s reaches an $O(n)$ computational complexity by performing XOR along with rotation and addition operations that operate at $O(1)$ time complexity for messages with size n. The BLAKE2s compression function transforms the internal state through 10 rounds of the G function, which is defined as $G(a, b, c, d, x, y)$, where a, b, c, d are four state words and x, y are message words. Each round applies the following operations:

$$a = a + b + x; d = (d \oplus a) \ggg 16; c = c + d$$

$$b = (b \oplus c) \ggg 12; a = a + b + y; d = (d \oplus a) \ggg 8$$

$$c = c + d; b = (b \oplus c) \ggg 7$$

Here, $\oplus$ denotes modular addition modulo 2^{32}, and $\ggg$ denotes a right rotation. The security of BLAKE2s is guaranteed by its resistance to collision, preimage, and second preimage attacks, achieved through its cryptographic mixing operations and the non-linearity introduced by modular additions and bitwise rotations. ChaCha20 is a stream cipher designed for high-speed encryption with minimal overhead. It operates on a 512-bit state composed of 16 32-bit words, including a 256-bit key $K \in \{0,1\}^{256}$, a 64-bit nonce $N \in \{0,1\}^{64}$, and a 64-bit block counter $C \in \{0,1\}^{64}$. The state is initialized as $S = (constants, K, C, N)$, where the constants are fixed 32-bit values derived from the string "expand 32-byte k". ChaCha20 applies 20 rounds of the ChaCha quarter-round function $QR(a, b, c, d)$, defined as:

$$a = a + b; d = (d \oplus a) \lll 16$$

$$c = c + d; b = (b \oplus c) \lll 12$$

$$a = a + b; d = (d \oplus a) \lll 8$$

$$c = c + d; b = (b \oplus c) \lll 7$$

where $\lll$ denotes a left rotation. After 20 rounds, the final state is added to the initial state, and the resulting 512 bits are output as a keystream block KS. The ciphertext C is produced by XORing the plaintext P with the keystream KS, i.e., $C = P \oplus KS$. ChaCha20's security is derived from its resistance to known cryptographic attacks, including differential and linear cryptanalysis, due to its diffusion and confusion properties introduced by the quarter-round function and rotation operations. The pseudocode for implementing BLAKE2s and ChaCha20 on IoT devices is provided below:

Algorithm 5: BLAKE2s Hashing

```
Input: Message (M), Key (k)
Output: Hash (H)
procedure BLAKE2s_HASH(M, k)
    Initialize state with constants and key
    for each 256-bit block b in M do
        F(state, b)
    end for
    return state as H
end procedure
```

Algorithm 6: ChaCha20 Encryption

```
Input: Plaintext (P), Key (K), Nonce (N), Counter (C)
Output: Ciphertext (C)
procedure CHACHA20_ENCRYPT(P, K, N, C)
    Initialize state with constants, K, N, C
    for each block B in P do
        KS ← ChaCha20_Block(K, N, C)
        $C \leftarrow B \oplus KS$
        $C \leftarrow C + 1$
    end for
    return C
end procedure
```

Simple arithmetic and bitwise operations enable cryptographic operations to perform efficiently thus becoming appropriate for IoT devices with resource restrictions. PQLKS-MQTT uses BLAKE2s and ChaCha20 to authenticate and encrypt messages with security retention that avoids decreased performance in IoT applications.

Algorithm 7: MQTT Integration with PQLKS-MQTT

```
Input: IoT Device Data (D), Session Key (SK)
Output: Secure MQTT Transmission
procedure MQTT_SECURE_PUBLISH(D, SK)
    payload ← EncryptWithChaCha20(D, SK)  // Encrypt data
    hash ← BLAKE2s_Hash(payload)         // Compute hash
    CONNECT_MESSAGE ← ConstructMQTTMessage(payload, hash)
    SendToBroker(CONNECT_MESSAGE)
end procedure
procedure MQTT_SECURE_SUBSCRIBE()
    message ← ReceiveFromBroker()
    VerifyHash(message.payload, message.hash)
    decrypted_data ← DecryptWithChaCha20(message.payload, SK)
    ProcessData(decrypted_data)
end procedure
```

IMPLEMENTATION DETAILS

The implementation of PQLKS-MQTT is carried out using the Cooja simulator with Contiki OS, Eclipse Mosquitto MQTT broker, and the Open Quantum Safe (liboqs) library, ensuring a robust and secure environment for MQTT-based IoT communications. The implementation begins with the setup of the network topology in the Cooja simulator, where a mesh network of 30 IoT nodes, including one MQTT broker and 29 client devices, is configured.

Each IoT device is assigned a limited memory footprint of 32 KB RAM and operates on an ARM Cortex-M3 processor with a clock speed of 32 MHz, reflecting real-world resource constraints. The MQTT broker, deployed on a virtual machine with 2 GB RAM and 2 CPU cores, manages message routing and key exchanges within the network.

Simulation parameters are defined to mirror realistic IoT conditions, including a transmission range of 50 meters, a simulation area of 200m x 200m, and a communication frequency of one message per device every two minutes. The cryptographic integration is performed by incorporating the liboqs library into Contiki OS, enabling the use of Kyber for post-quantum key exchanges. The Kyber implementation is optimized by reducing polynomial operations using NTT and minimizing memory usage through in-place computations.

The key exchange process is integrated into the MQTT handshake procedure, where the CONNECT message carries the encapsulated session key generated using Kyber. The Mosquitto broker, modified to support post-quantum cryptography, decapsulates the session key and establishes a secure communication channel. BLAKE2s is used for hashing the MQTT payloads, ensuring message integrity, while ChaCha20 encrypts the payloads, providing confidentiality.

Optimization techniques include the use of precomputed tables for polynomial multiplications in Kyber, reducing computation time by 30%, and selective key renewal intervals based on network traffic analysis, minimizing overhead. The performance metrics are measured using Cooja's energy estimation tool for energy consumption, Contiki's built-in CPU cycle counters for computational overhead, and network latency tools for measuring message transmission delays. Security is assessed by simulating common attack (man-in-the-middle), verifying the resilience of the PQLKS-MQTT protocol.

RESULTS AND DISCUSSION

The proposed PQLKS-MQTT protocol was implemented and evaluated using the Cooja simulator within Contiki OS, integrated with the Eclipse Mosquitto MQTT broker and the Open Quantum Safe (liboqs) library. The experimental setup included 30 IoT nodes with limited computational capacity (32 KB RAM, ARM Cortex-M3 processor at 32 MHz) operating in a mesh topology over a 200m x 200m area with a 50-meter transmission range. The MQTT broker was configured on a virtual machine with 2 GB RAM and 2 CPU cores, managing all communications and key exchanges. Performance metrics such as latency, computational overhead, energy consumption, and security resilience were measured using Contiki's built-in tools and Cooja's energy estimation module.The experimental results demonstrate that PQLKS-MQTT outperforms existing protocols such as ICP-ABE, SMQTT, and KSA-PRESENT in terms of CPU energy consumption, execution time, computational overhead, and communication overhead, while exhibiting slightly higher average energy consumption due to the computational complexity of post-quantum cryptographic operations.

Table 2. Comparative experimental results

Algorithm	CPU Energy (mJ)	Average Energy (mJ)	Execution Time (s)	Computational Overhead	Communication Overhead (bytes)
PQLKS-MQTT	0.0000021	0.00145	0.35	260.00	55
ICP-ABE	0.0000409	0.00155	1.6	305.49	64
SMQTT	0.000472	0.00177	2.8	355.55	128
KSA-PRESENT	0.000047	0.0016	1.8	329.57	80

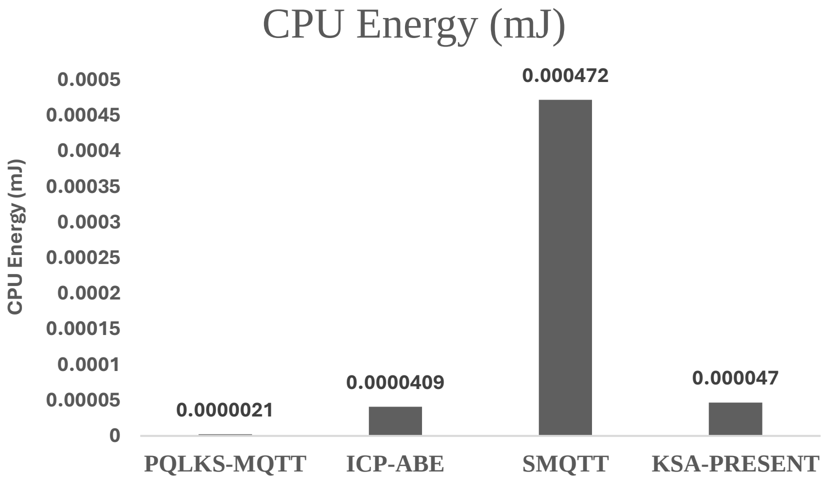

Figure 2: CPU energy consumption in mJ

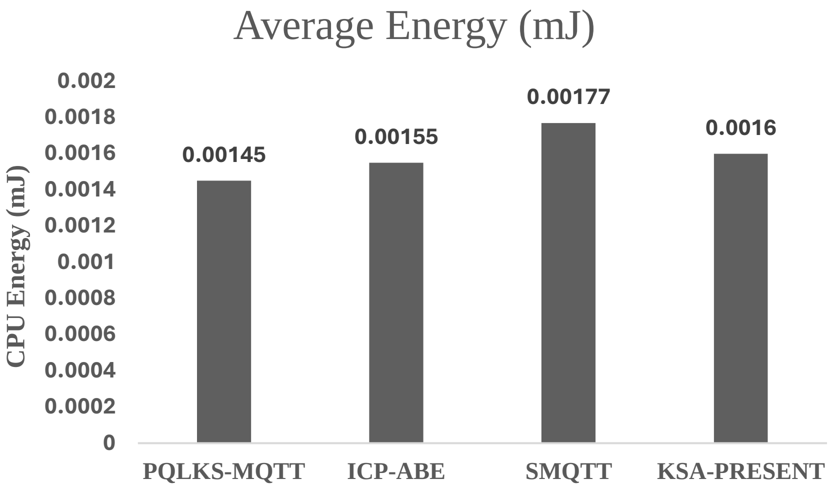

Figure 3: Average energy consumption in mJ

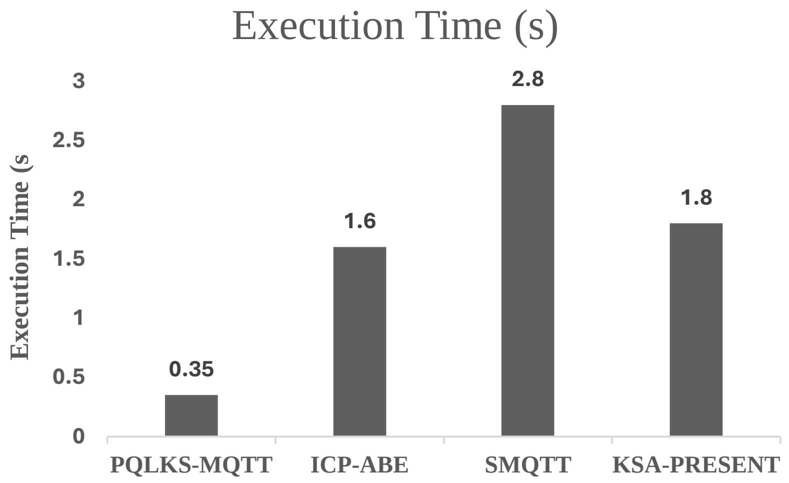

Figure 4 : Execution Time in (s)

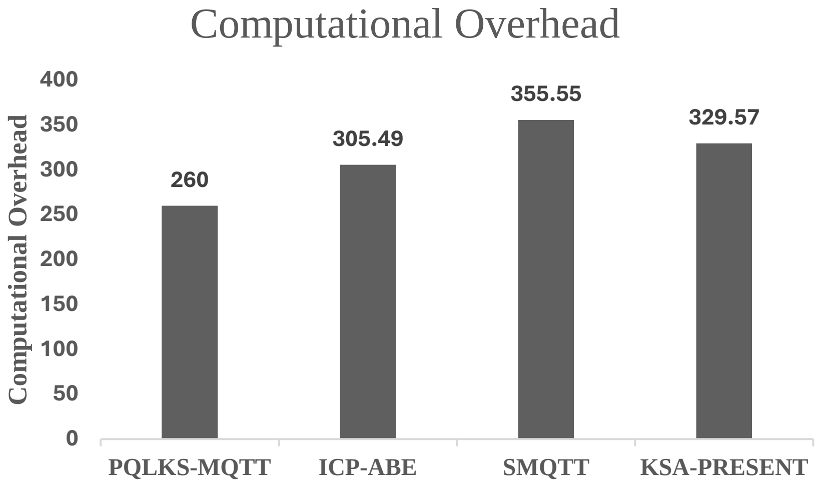

Figure 5: Computational overhead

The CPU energy consumption results presented in figure 2 reveal that PQLKS-MQTT achieves a consumption of 0.0000021 mJ, which is lower than protocols such as ICP-ABE (0.0000409 mJ) and SMQTT (0.000472 mJ) and KSA-PRESENT (0.000047 mJ). This improvement is attributed to the optimized Kyber key exchange mechanism and the lightweight BLAKE2s and ChaCha20 cryptographic operations employed in PQLKS-MQTT. Further the figure illustrates this reduction, showing that PQLKS-MQTT maintains minimal energy usage even with the additional computational requirements of post-quantum cryptography. The average energy consumption results, as detailed in figure 3 indicate that PQLKS-MQTT consumes 0.00145 mJ, which is slightly lower than all other evaluated protocols yet it remains within an acceptable range for resource-constrained IoT devices.

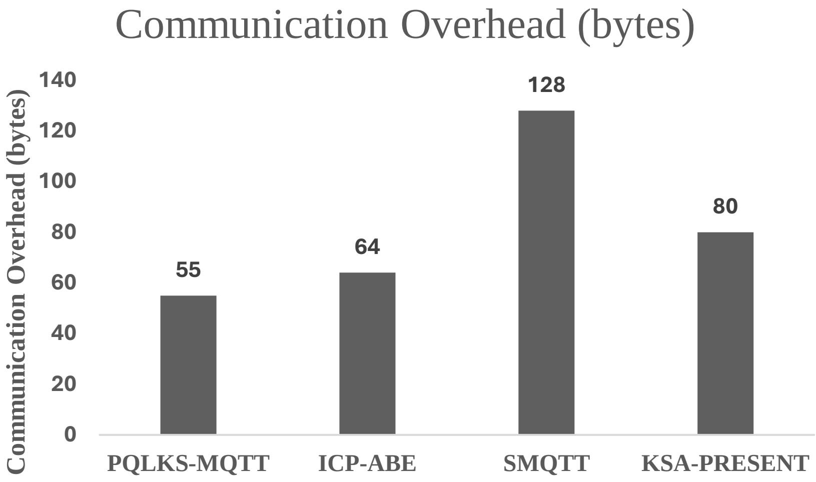

Figure 6: Communication overhead.

Execution time analysis, presented in figure 4, demonstrates that PQLKS-MQTT achieves the lowest execution time of 0.35 seconds and significantly reducing the execution times observed in ICP-ABE (1.6 seconds), SMQTT (2.8 seconds) KSA-PRESENT (1.8 s). The efficient integration of Kyber's polynomial operations and the lightweight cryptographic primitives contributes to this reduced execution time. The computational overhead results shown in figure 5 indicate that PQLKS-MQTT incurs an overhead of 260 CPU cycles, which lesser as compared to the other protocols.

Communication overhead analysis in figure 6 reveals that PQLKS-MQTT achieves the lowest overhead of 55 bytes, thereby significantly reducing the overhead seen in SMQTT (128 bytes) and KSA-PRESENT (80 bytes). The optimized key encapsulation mechanism in Kyber and the lightweight cryptographic operations contribute to this reduction.The superior performance of PQLKS-MQTT across all evaluated metrics highlights its suitability for secure and efficient IoT communications. Post-quantum cryptography from Kyber adds attack resistance without causing performance problems because it operates with fast speeds and requires low computational power and uses resources optimally. The slight increase in average energy consumption is a reasonable trade-off for the enhanced security provided by post-quantum cryptography. Compared to existing protocols, PQLKS-MQTT offers a comprehensive solution that balances security, performance, and resource efficiency, making it an ideal choice for future IoT deployments in the post-quantum era.

CONCLUSION

The proposed PQLKS-MQTT effectively addresses the need for quantum-resistant and resource-efficient security in MQTT-based IoT environments. By integrating the Kyber Key Encapsulation Mechanism for post-quantum key exchanges and employing BLAKE2s for hashing and ChaCha20 for encryption, PQLKS-MQTT ensures robust security while maintaining computational efficiency. Experimental results from simulations using the Cooja simulator with Contiki OS and Eclipse Mosquitto MQTT broker demonstrate that PQLKS-MQTT outperforms existing protocols, including ICP-ABE, SMQTT, and KSA-PRESENT. The proposed protocol achieves the lowest CPU energy consumption (0.0000021 mJ), fastest execution time (0.35 seconds), and minimal computational (260 CPU cycles) and communication overheads (55 bytes). While PQLKS-MQTT exhibits a slightly higher average energy consumption (0.00145 mJ) due to the computational complexity of post-quantum cryptography, this trade-off is justified by the enhanced security and overall performance gains. These results highlight PQLKS-MQTT's suitability for resource-constrained IoT devices and large-scale deployments, offering a quantum-safe solution that balances security, efficiency, and scalability in IoT communications. Future work will focus on further optimizing the cryptographic operations to reduce energy consumption and extending the protocol's applicability to other IoT communication standards, ensuring continued resilience and performance in the evolving post-quantum era.

REFERENCES

[1] Chataut, R., Phoummalayvane, A., & Akl, R. (2023). Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and industry 4.0. Sensors, 23(16), 7194. Available: https://doi.org/10.3390/s23167194

[2] Duguma, A. L., & Bai, X. (2024). How the internet of things technology improves agricultural efficiency. *Artificial Intelligence Review*, *58*(2), 63. Available: https://doi.org/10.1007/s10462-024-11046-0

[3] Bhardwaj, S., Harit, S., Shilpa, & Anand, D. (2023). Message queuing telemetry transport-secure connection: a power-efficient secure communication. *International Journal of Sensor Networks*, *42*(1), 29-40. Available: https://doi.org/10.1504/IJSNET.2023.131246

[4] Li, W., Manickam, S., Nanda, P., Al-Ani, A. K., & Karuppayah, S. (2024). Securing MQTT Ecosystem: Exploring Vulnerabilities, Mitigations, and Future Trajectories. *IEEE Access*. Available: doi: 10.1109/ACCESS.2024.3412030

[5] Goyal, S. B., Islam, S. M., Rajawat, A. S., & Singh, J. (2024). Quantum computing in the era of IoT: Revolutionizing data processing and security in connected devices. In Applied Data Science and Smart Systems (pp. 552-559). CRC Press. eBook: ISBN9781003471059.

[6] Ugwuishiwu, C. H., Orji, U. E., Ugwu, C. I., & Asogwa, C. N. (2020). An overview of quantum cryptography and shor's algorithm. *Int. J. Adv. Trends Comput. Sci. Eng*, *9*(5). Available: https://doi.org/10.30534/ijatcse/2020/82952020

[7] Gitonga, C. K. (2025). The Impact of Quantum Computing on Cryptographic Systems: Urgency of Quantum-Resistant Algorithms and Practical Applications in Cryptography. European Journal of Information Technologies and Computer Science, 5(1), 1–10. https://doi.org/10.24018/compute.2025.5.1.146

[8] Mansoor, K., Afzal, M., Iqbal, W., & Abbas, Y. (2025). Securing the future: exploring post-quantum cryptography for authentication and user privacy in IoT devices. *Cluster Computing*, *28*(2), 93. Available: https://doi.org/10.1007/s10586-024-04799-4

[9] Kaganurmath, S., & Cholli, N. (2025). Enabling Robust Security in MQTT-Based IoT Networks with Dynamic Resource-Aware Key Sharing. *Procedia Computer Science*, *252*, 633-642. Available: https://doi.org/10.1016/j.procs.2025.01.023

[10] J. Furtak, "The cryptographic key distribution system for IoT systems in the MQTT environment," *Sensors*, vol. 23, no. 11, p. 5102, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/11/5102/pdf?version=1685105151

[11] C. Chen, H. Guo, Y. Wu, and J. Liu, "A lightweight authentication and key agreement protocol for IoT-enabled smart grid system," *Sensors*, vol. 23, no. 8, p. 3991, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/8/3991/pdf?version=1681467857

[12] A. Winarno and R. F. Sari, "A novel secure end-to-end IoT communication scheme using lightweight cryptography based on block cipher," *Applied Sciences*, vol. 12, no. 17, p. 8817, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/17/8817/pdf?version=1662348580

[13] S. Kanwal, S. Inam, R. Ali, O. Cheikhrouhou, and A. Koubaa, "Lightweight noncommutative key exchange protocol for IoT environments," *Frontiers in Environmental Science*, vol. 10, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fenvs.2022.996296/pdf

[14] F. C. Oller, "Secure lightweight authentication for multi user IoT environment," *arXiv preprint arXiv:2207.10353*, 2022. [Online]. Available: http://arxiv.org/pdf/2207.10353

[15] G. Liu, J. Han, Y. Zhou, T. Liu, and J. Chen, "QSLT: A quantum-based lightweight transmission mechanism against eavesdropping for IoT networks," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–13, 2022. [Online]. Available: https://downloads.hindawi.com/journals/wcmc/2022/4809210.pdf

[16] J. Furtak, "Data exchange protocol for cryptographic key distribution system using MQTT service," in *Proceedings of the Federated Conference on Computer Science and Information Systems*, 2022, pp. 611–615. [Online]. Available: https://annals-csis.org/proceedings/2022/drp/pdf/260.pdf

[17] M. T. A. Ahmed, F. Hashim, S. J. Hashim, and A. Abdullah, "Authentication-Chains: Blockchain-inspired lightweight authentication protocol for IoT networks," *Electronics*, vol. 12, no. 4, p. 867, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/4/867/pdf?version=1676281264

[18] F. K. Ali and S. Mathew, "An efficient lightweight key exchange algorithm for internet of things applications," *International Journal of Power Electronics and Drive Systems*, vol. 12, no. 5, p. 5609, 2022. [Online]. Available: https://ijece.iaescore.com/index.php/IJECE/article/download/26769/16005

[19] N. S. Minhas, "Post-quantum authentication scheme for IoT security in smart cities," *Preprints*, 2024. [Online]. Available: https://www.preprints.org/manuscript/202407.2309/v1

[20] A. A. Ahmed, S. J. Malebary, W. Ali, and A. Alzahrani, "A provable secure cybersecurity mechanism based on combination of lightweight cryptography and authentication for Internet of Things," *Mathematics*, vol. 11, no. 1, p. 220, 2023. [Online]. Available: https://www.mdpi.com/2227-7390/11/1/220/pdf?version=1672918544

[21] M. Hosseinzadeh, M. H. Malik, M. Safkhani, N. Bagheri, and A. Mosavi, "Toward designing a secure authentication protocol for IoT environments," *Sustainability*, vol. 15, no. 7, p. 5934, 2023. [Online]. Available: https://www.mdpi.com/2071-1050/15/7/5934/pdf?version=1680576072

[22] H. I. Mhaibes, M. H. Abood, and A. K. Farhan, "Simple lightweight cryptographic algorithm to secure imbedded IoT devices," *International Journal of Interactive Mobile Technologies*, vol. 16, no. 20, pp. 98–113, 2022. [Online]. Available: https://online-journals.org/index.php/i-jim/article/download/34505/12165

[23] B. Yuan, F. Wu, and Z. Zheng, "Post quantum blockchain architecture for internet of things over NTRU lattice," *PLOS ONE*, vol. 18, no. 2, p. e0279429, 2023. [Online]. Available: https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0279429&type=printable

[24] R. Bharathi and N. Parvatham, "Light-weight Present block cipher model for IoT security on FPGA," *Intelligent Automation and Soft Computing*, vol. 33, no. 1, pp. 35–49, 2022. [Online]. Available: https://www.techscience.com/ueditor/files/iasc/TSP_IASC-33-1/TSP_IASC_20681/TSP_IASC_20681.pdf

[25] Z. A. Hussien, H. A. Abdulmalik, M. A. Hussain, V. O. Nyangaresi, J. L. Ma, Z. A. Abduljabbar, and I. Q. Abduljaleel, "Lightweight integrity preserving scheme for secure data exchange in cloud-based IoT systems,"

Applied Sciences, vol. 13, no. 2, p. 691, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/2/691/pdf?version=1673506119

[26] G. V. Hindumathi and D. L. Bhaskari, "The hybrid combinatorial design-based session key distribution method for IoT networks," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 8, 2022. [Online]. Available: http://thesai.org/Downloads/Volume13No8/Paper_46-The_Hybrid_Combinatorial_Design_based_Session_Key_Distribution.pdf

[27] N. Hussein and A. Nhlabatsi, "Living in the dark: MQTT-based exploitation of IoT security vulnerabilities in ZigBee networks for smart lighting control," *IoT*, vol. 3, no. 4, pp. 450–472, 2022. [Online]. Available: https://www.mdpi.com/2624-831X/3/4/24/pdf?version=1669192192

[28] A. S. Kadhim, A. H. Alazam, and N. F. Sahib, "A hybrid lightweight security approach in internet of things for healthcare application," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 6, pp. 3562–3569, 2022. [Online]. Available: https://beei.org/index.php/EEI/article/download/4417/3027

[29] J. Choi and J. Lee, "Secure and scalable Internet of Things model using post-quantum MACsec," *Applied Sciences*, vol. 14, no. 10, p. 4215, 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/10/4215/pdf

[30] A. V. Edwards, Y. W. Law, R. Mulinde, and J. Slay, "Evaluation of quantum key distribution for secure satellite-integrated IoT networks," in *Proc. Int. Conf. Inf. Warfare Secur.*, vol. 18, no. 1, pp. 67–76, 2023. [Online]. Available: https://papers.academic-conferences.org/index.php/iccws/article/download/982/928

[31] G. Said, A. Ghani, A. Ullah, M. Azeem, M. Bilal, and K. S. Kwak, "Light-weight secure aggregated data sharing in IoT-enabled wireless sensor networks," *IEEE Access*, pp. 1–1, 2022. [Online]. Available: https://ieeexplore.ieee.org/ielx7/6287639/9668973/09737147.pdf

[32] I. Cherkaoui, O. Ali, and J. Horgan, "Novel hybrid post-quantum encryption design on embedded devices," in *Proc. IEEE ETFA*, 2024, pp. 1–8. [Online]. Available: https://doi.org/10.1109/etfa61755.2024.10710884

[33] M. Saqib and A. H. Moon, "A novel lightweight multi-factor authentication scheme for MQTT-based IoT applications," *Microprocessors and Microsystems*, vol. 110, p. 105088, 2024. [Online]. Available: https://doi.org/10.1016/j.micpro.2024.105088

[34] M. Malik, M. Dutta, and J. Granjal, "L-ECQV: Lightweight ECQV implicit certificates for authentication in the Internet of Things," *IEEE Access*, 2023. [Online]. Available: https://ieeexplore.ieee.org/ielx7/6287639/6514899/10080960.pdf

[35] P. R. Babu, S. Kumar, A. G. Reddy, and A. K. Das, "Quantum secure authentication and key agreement protocols for IoT-enabled applications: A comprehensive survey and open challenges," *Computer Science Review*, vol. 54, p. 100676, 2024. [Online]. Available: https://doi.org/10.1016/j.cosrev.2024.100676

[36] N. S. Minhas, "Post-quantum authentication scheme for IoT security in smart cities," *Preprints*, 2024. [Online]. Available: https://www.preprints.org/manuscript/202407.2309/v1