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

Research Article

Performance Analysis of Dynamic Light Weight Cryptographic Algorithm for Message Queuing Telemetry Transport based IoT Communications

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
[2]*Professor and Dean CMR University, Bengaluru-INDIA*;Nagaraj.cholli@gmail.com
[3]*Professor, Department of Information Science Engineering, RV College of Engineering, Bengaluru-INDIA*; analamr@rvce.edu.in

ARTICLE INFO

Received: 28 Dec 2024

Revised: 18 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

This paper presents a novel approach aimed at developing a secure secret key-sharing system optimized for resource-constrained Internet of Things devices. Focusing on the Message Queuing Telemetry Transport protocol, the research endeavours to establish secure communication channels between Internet of Things devices and brokers, thereby enhancing the overall security of Message Queuing Telemetry Transport-based Internet of Things deployments. This research introduces a novel Dynamic Lightweight Authentication for Message Queuing Telemetry Transport mechanism tailored to the unique needs of Internet of Things devices operating under the Message Queuing Telemetry Transport protocol. Dynamic Lightweight Authentication for Message Queuing Telemetry Transport mechanism leverages an innovative lightweight Generalized Feedback Shift Register based Pseudo-Random Number Generator to generate ephemeral keys, ensuring secure communication while addressing the limitations of computational power and energy resources. Through a detailed comparative analysis with existing cryptographic solutions, the Dynamic Lightweight Authentication for Message Queuing Telemetry Transport mechanism demonstrates superior performance in terms of computational overhead, energy consumption, and execution time, while maintaining robust security against common attacks such as Man-in-the-Middle and Denial of Service). he proposed algorithm's adaptability and scalability are validated using the Cooja simulator, where simulated Internet of Things networks are subjected to various threat scenarios. The results confirm the Dynamic Lightweight Authentication for Message Queuing Telemetry Transport mechanism's efficacy, showcasing a significant reduction in resource utilization without compromising the strength of the security provided.

Keywords: Internet of Things (IoT), Dynamic Lightweight Authentication for MQTT Resource-Constrained Devices, Security, Pseudo-Random Number Generator, Ephemeral Key Generation.

INTRODUCTION

The emergence of the Internet of Things (IoT) has brought forth a new technological era marked by enhanced connectivity and automation across multiple sectors such as smart homes, healthcare, and industrial systems [1]. This interconnectedness, while beneficial in terms of efficiency and convenience, introduces significant cybersecurity challenges. A particular area of concern is the secure communication between IoT devices and brokers, especially within the framework of the Message Queuing Telemetry Transport (MQTT) protocol [2]. This protocol, designed for low-bandwidth and high-latency networks, is preferred for its efficiency but is vulnerable when implemented in systems with resource-limited devices [3].

Resource-constrained devices, with their limited computational capabilities, pose a challenge to implementing traditional cryptographic methods and security protocols, making them prone to various security threats [4]. This situation necessitates a novel approach to secure secret key sharing that accommodates the unique needs of IoT devices, ensuring secure communication channels without excessive computational demands.

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

Research Article

This paper proposes a lightweight secret key-sharing protocol tailored for IoT environments, employing innovative cryptographic techniques to alleviate the computational load on IoT devices. This solution aligns with the advancements in lightweight cryptography aimed at delivering robust security in systems with limited resources [5 - 7].

Moreover, a detailed security analysis and threat assessment are conducted to pinpoint and counteract potential vulnerabilities in MQTT-based IoT networks, using established cybersecurity risk assessment frameworks [8]. This research contributes to the IoT security domain by offering a viable solution to secure communications within the constraints of IoT devices, thereby enhancing the resilience and sustainability of IoT ecosystems. The main motivation for this is the rapid integration of IoT devices into essential infrastructures heightens security vulnerabilities, compounded by the limitations of these devices and the MQTT protocol's security shortcomings. This research is driven by the urgent need to improve MQTT-based IoT network security, aiming to maintain confidentiality, integrity, and availability of communications without overloading the devices' computational resources. The main objective of this work is to develop a lightweight secret key-sharing protocol for IoT devices, focusing on:

- Designing an efficient key-sharing mechanism with minimal computational and energy demands.
- Performing an in-depth security analysis to uncover and mitigate vulnerabilities in MQTT-based IoT networks.
- Assessing the protocol's computational efficiency, scalability, and security to ensure its real-world applicability.

LITERATURE REVIEW

The integration of IoT technologies into various sectors has brought about a transformative shift in how data is collected, processed, and utilized. Among the protocols facilitating IoT communication, the MQTT stands out for its lightweight design and efficiency, making it particularly suitable for resource-constrained IoT devices [3]. However, the constrained nature of these devices and the inherent simplicity of MQTT pose significant security challenges, necessitating the development of advanced security mechanisms that are both effective and resource-efficient.

MQTT, originally designed for minimal bandwidth and device resources, employs a publish/subscribe model that, while efficient, introduces security vulnerabilities such as eavesdropping, message forgery, and man-in-the-middle (MitM) attacks [6]. The protocol's lack of built-in security features has prompted researchers to explore various security enhancements, focusing on authentication, encryption, and integrity checks [4].

The unique constraints of IoT devices, including limited processing power, memory, and energy resources, preclude the use of traditional, computationally intensive security mechanisms. This has led to the exploration of lightweight cryptographic solutions tailored to the IoT context [5]. Alaba et al. [7] highlighted the necessity for lightweight security protocols in IoT, emphasizing the balance between security and resource efficiency.

Recent studies have proposed various lightweight encryption and authentication mechanisms for IoT devices. For example, Al Salami et al. [9] introduced a lightweight encryption scheme designed to secure MQTT communications with minimal computational overhead, demonstrating its applicability in real-world IoT scenarios. Similarly, Ethui et al. [10] developed a lightweight mutual authentication protocol for IoT systems, ensuring device and data security without significantly impacting device performance.

The field of lightweight cryptography has seen significant advancements aimed at addressing the security needs of resource-constrained environments. Singh et al. [11] explored the design of efficient cryptographic algorithms that offer robust security while being feasible for IoT devices. These algorithms are characterized by their low computational requirements and minimal energy consumption, making them ideal for securing MQTT-based communications. In addition to cryptographic algorithms, the development of secure key management and distribution mechanisms is crucial for IoT security. Pothumarti et al. [12] proposed a dynamic key management system that adapts to the changing network topology and device capabilities in IoT ecosystems, facilitating secure communication while managing resource constraints. Secure authentication and key management are pivotal in

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

Research Article

enhancing the security of MQTT communications. Abdelrazig et al. [13] introduced an innovative authentication framework for MQTT that leverages blockchain technology to ensure device authenticity and data integrity, offering a novel approach to secure IoT communications.

Moreover, the integration of lightweight Public Key Infrastructures (PKIs) and certificate management systems has been investigated to provide scalable and secure authentication for IoT devices. The work by Khan, Minhaj & Salah [14] on lightweight PKIs for IoT highlights the potential of such systems in enhancing the security of MQTT without overwhelming device resources.Despite these advancements, several challenges remain in securing MQTT for resource-constrained IoT devices. The trade-off between security strength and resource consumption continues to be a central concern. Future research directions may include the exploration of quantum-resistant cryptographic algorithms for IoT, given the emerging threat posed by quantum computing [15].

Additionally, the integration of Artificial Intelligence (AI) and Machine Learning (ML) techniques in IoT security, presents a promising avenue for developing adaptive and resilient security mechanisms capable of identifying and mitigating novel threats in real-time [16].

The literature underscores the critical need for security enhancements in MQTT-based IoT communications, particularly for resource-constrained devices. While significant progress has been made in developing lightweight cryptographic solutions, ongoing research is essential to address the evolving security challenges in the IoT landscape. The exploration of security enhancements for MQTT-based IoT communications, particularly for resource-constrained devices, has unveiled a pertinent research gap that directly aligns with the objectives of this paper. Despite the advancements in lightweight cryptographic solutions and authentication protocols tailored for IoT environments, there exists a notable gap in the development and integration of dynamic, resource-efficient authentication mechanisms that can seamlessly adapt to the evolving security needs and operational constraints of IoT devices. The approaches should aim to develop authentication mechanisms that are not only adaptive and scalable but also cognizant of the energy and computational constraints of IoT devices, ensuring that security enhancements do not detract from the primary functionalities and service quality of the IoT ecosystem. This paper aims to contribute to bridging this research gap by proposing a novel dynamic lightweight authentication mechanism tailored for MQTT-based IoT communications, focusing on ensuring robust security while adhering to the stringent resource constraints of IoT devices. With the literature survey carried out, this work concisely define the problem to address the research related to secure communication channels between IoT devices and brokers that do not compromise device functionality.

PROPOSED WORK

To address the critical challenge of securing MQTT-based IoT communications within resource-constrained environments, this paper proposes a novel lightweight authentication mechanism, termed “Dynamic Lightweight Authentication for MQTT (DLA-MQTT).” DLA-MQTT is specifically designed to provide a robust yet computationally efficient authentication process for IoT devices, leveraging the unique characteristics of MQTT protocol and the constrained nature of IoT devices. The novelty of DLA-MQTT lies in its dynamic authentication scheme, which combines ephemeral key generation with lightweight cryptographic hashes and minimalistic handshake protocols to ensure security without the computational overhead typically associated with traditional cryptographic methods.

CONCEPTUAL FRAMEWORK

Figure 1 represents the conceptual framework for the DLA-MQTT mechanism as applied to MQTT-based IoT networks. At the centre of the framework is the MQTT Broker, acting as the communication hub that orchestrates the flow of messages between the connected IoT devices. These devices, depicted as smart devices around the periphery of the MQTT Broker, are the network's endpoints that initiate the communication process.

Each smart device is equipped with an ephemeral key generation module, indicating the production of temporary session keys crucial for securing each communication session. These keys are generated in a resource-aware manner, considering the constraints of the smart devices, which are often limited in computational power and energy reserves. Once a session key is generated, the device commences the handshake process to and from the MQTT Broker. This

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

Research Article

process encapsulates the authentication exchange, leveraging the lightweight cryptographic hashes to ensure the integrity and authenticity of the messages without introducing significant computational overhead. Upon successful authentication, a secure communication channel is established between the devices to the MQTT Broker. This channel ensures that all subsequent communications are encrypted, safeguarding the data integrity and confidentiality against potential eavesdroppers or unauthorized entities.

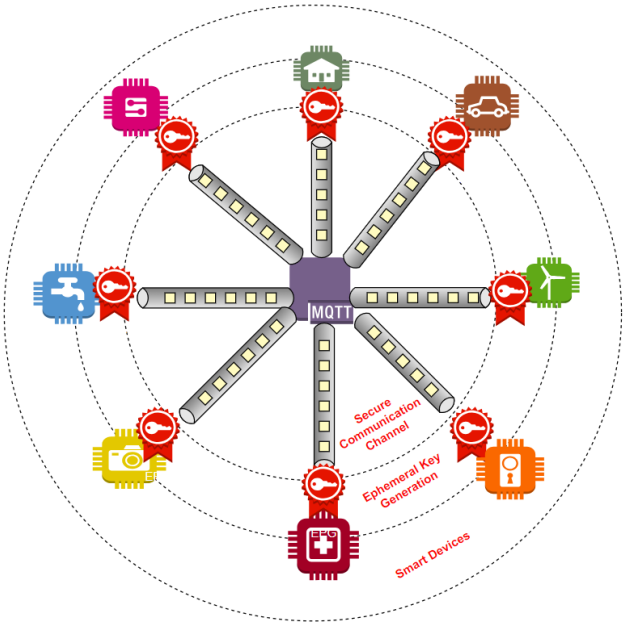

Figure 1. Conceptual Framework of DLA-MQTT Mechanism for MQTT-Based IoT Networks

The figure 1 given above effectively represents the essential operational phases of the DLA-MQTT mechanism, namely key generation, handshake protocol, and secure communication. It emphasizes the mechanism's appropriateness for IoT environments, where the importance is placed on security, efficiency, and resource preservation. The circular dots in the figure highlights the distributed nature of the network, where the broker enables secure interactions among the different IoT devices.

DLA-MQTT operates on a dynamic security model that adapts to the operational context of IoT devices, considering their energy constraints, processing capabilities, and the need for periodic updates in security credentials to mitigate the risk of long-term key compromise. The mechanism comprises three core components:

EPHEMERAL KEY GENERATION

Utilizing lightweight Pseudo-Random Number Generators (PRNGs), DLA-MQTT generates short-lived session keys for each authentication session. These keys are used only once, reducing the risk associated with key compromise. The ephemeral key generation process, which is the core of DLA-MQTT, is intended to generate temporary session keys dynamically and for use only during a single communication session. This approach is predicated on the use of lightweight PRNGs, optimized for minimal computational load and energy consumption. The keys generated are inherently short-lived, thereby significantly reducing the window of vulnerability in case of potential key compromise. This transient nature of session keys serves as a fundamental security feature, ensuring that each session maintains a unique encryption domain, thereby obfuscating patterns that could be exploited by adversaries.

MINIMALISTIC HANDSHAKE PROTOCOL

Inspired by the simplicity of the MQTT protocol, the handshake protocol in DLA-MQTT is designed to minimize the number of messages exchanged between the IoT device and the broker. This protocol leverages the existing MQTT connection process, embedding authentication within the CONNECT and CONNACK messages to avoid additional network overhead. DLA-MQTT incorporates a streamlined handshake protocol that is meticulously designed to

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

Research Article

minimize the exchange overhead inherent in traditional authentication processes. This protocol is ingeniously embedded within the MQTT's existing CONNECT and CONNACK message exchange framework, thereby leveraging the protocol's inherent efficiency and reducing the need for additional network traffic. The handshake protocol initiates with the IoT device embedding the newly generated session key within the payload of the MQTT CONNECT message. Accompanying this key is a lightweight cryptographic hash of the message, serving as a means for the receiving broker to verify the authenticity and integrity of the incoming connection request.

LIGHTWEIGHT CRYPTOGRAPHIC HASHES

To further reduce the computational load, DLA-MQTT employs lightweight cryptographic hashes for message integrity and authentication verification. These hashes are selected based on their low computational complexity and suitability for hardware with limited processing capabilities. The selection and deployment of cryptographic hashes within DLA-MQTT are guided by the principle of computational efficiency, essential for maintaining the operational viability of resource-constrained IoT devices. These lightweight hashes are characterized by their reduced computational complexity, making them well-suited for devices with limited processing capabilities. The role of these hashes is twofold: first, to ensure the integrity of the messages exchanged during the handshake process; and second, to provide a mechanism for the broker to authenticate the identity of the connecting device. The careful selection of these hashes is crucial, balancing the need for security with the imperative to conserve device resources.

INTEGRATION WITH MQTT PROTOCOL

The integration of DLA-MQTT with the MQTT protocol is seamless, designed to complement the protocol's lightweight, publish/subscribe model without imposing significant additional overhead. The DLA-MQTT handshake is initiated as part of the MQTT connection setup, with the ephemeral session key and the corresponding hash being transmitted within the CONNECT message. This integration ensures that the authentication process is not only secure but also efficient, avoiding the introduction of unnecessary steps or messages that could burden the network, or the devices involved.

ALGORITHM DESIGN AND DEVELOPMENT

The DLA-MQTT mechanism is primarily composed of a set of algorithms, each of which is carefully crafted to perform functions within the security architecture. The fundamental algorithms of DLA-MQTT guarantee the resilience, effectiveness, and adjustability of the protocol to the resource-limited characteristics of IoT devices. The algorithms facilitate safe and effective device authentication and communication integrity by fusing cutting-edge cryptographic techniques with the low overhead requirements of MQTT.

EPHEMERAL KEY GENERATION

Algorithm 1 focuses on the generation of ephemeral keys using a Generalized Feedback Shift Register (GFSR)-based PRNG. The keys are designed to be short-lived and bound to the context of a single session, which significantly elevates the security posture by reducing the temporal attack surface that is often exploited in key compromise scenarios.

ALGORITHM 1: EPHEMERAL KEY GENERATION USING GFSR-BASED PRNG

The ephemeral key generation algorithm is a crucial component of the DLA-MQTT mechanism, designed to produce a unique, temporary session key (K_{sess}) for each communication session. This algorithm employs a GFSR-based PRNG to ensure high-quality randomness and security. The detailed steps, along with cryptographic notations and mathematical equations, are outlined below:

Step 1: GFSR Initialization

1: *State Initialization:* The GFSR is initialized with a state vector $S = (s_0, s_1, \ldots, s_{n-1})$, where (n) is the length of the register, and each (s_i) is a bit.

$$S = (s_0, s_1, \ldots, s_{n-1}) \quad \ldots\ldots (1)$$

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

Research Article

2: *Seed Selection:* A seed value ($Seed$) is derived from a combination of device-specific parameters (e.g., device ID, timestamps) and environmental factors (e.g., ambient noise, sensor readings) to ensure variability and unpredictability in the key generation process.

$$Seed = Hash(DeviceParams \parallel EnvFactors) \quad \text{..... (2)}$$

3: *State Seeding:* The seed value $Seed$ is used to initialize the state vector S, ensuring that the initial state is unpredictable.

$$S_0 = SeedFunction(Seed) \quad \text{..... (3)}$$

Step 2: GFSR Iteration

1: *Feedback Function:* Define a feedback function ($f(S)$) that takes the current state (S) as input and produces a new bit based on a predefined linear or nonlinear combination of bits in (S).

$$s_{new} = f(s_i, s_j, \dots) \; for \; i, j \in \{0,1,\dots,n-1\} \quad \text{..... (4)}$$

2: *State Update:* Update the state vector (S) by shifting all bits to the left and inserting the new bit (s_{new}) at the end.

$$S' = (s_1, s_2, \dots, s_{n-1}, s_{new}) \quad \text{..... (5)}$$

Step 3: Key Generation

1: *Bit Sampling:* Sample a sequence of bits from the updated state vector (S') to form the session key (K_{sess}). The number of bits sampled depends on the desired key length.

$$K_{sess} = (s'_0, s'_1, \dots, s'_{k-1}) \text{ where k is the key length} \quad \text{..... (6)}$$

2: *Key Construction:* Concatenate the sampled bits to construct the ephemeral session key (K_{sess}), which will be used for securing the communication session.

$$K_{sess} = \bigoplus_{i=0}^{k-1} s'_i \quad \text{..... (7)}$$

The ephemeral key generation algorithm outlined above leverages the GFSR-based PRNG to ensure that each communication session between an IoT device and the MQTT broker is secured with a unique, high-quality random key. This approach enhances the security of the DLA-MQTT mechanism, providing robust protection against various cryptographic attacks while maintaining efficiency and scalability for resource-constrained IoT devices.

AUTHENTICATION PAYLOAD CREATION

Following the key generation, Algorithm 2 constructs the authentication payload, which is a composite of the ephemeral key, device identifier, and nonce. The payload is hashed to ensure integrity, forming an indispensable part of the authentication process.

ALGORITHM 2: AUTHENTICATION PAYLOAD CREATION

The Authentication Payload Creation algorithm is a vital process in the DLA-MQTT mechanism, where the ephemeral session key (K_{sess}) generated by the GFSR-based PRNG is embedded into the MQTT CONNECT message along with additional authentication data. This payload is then secured with a lightweight cryptographic hash to ensure integrity and authenticity. The steps, notations, and equations involved in creating the authentication payload are detailed below:

Step 1: Payload Composition

1: ***Device Identifier Incorporation:*** Include the unique device identifier (ID_{dev}) to ensure that the payload can be unequivocally associated with the sending device.

$$P_{ID} = ID_{dev} \quad \text{..... (8)}$$

2: ***Session Key Embedding****:* Embed the ephemeral session key (K_{sess}) generated in Algorithm 1 into the payload. This key is essential for securing the communication session.

$$P_{Key} = K_{sess} \quad \text{..... (9)}$$

3: ***Nonce Addition:*** Append a nonce (N) to the payload to prevent replay attacks. The nonce can be a timestamp or a counter that ensures the payload is fresh.

$$P_{Nonce} = N \quad \text{..... (10)}$$

4: **Payload Construction:** Construct the final payload (P) by concatenating (P_{ID}), (P_{Key}), and (P_{Nonce}).

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

Research Article

$$P = P_{ID} \parallel P_{Key} \parallel P_{Nonce} \quad \text{..... (11)}$$

Step 2: Hash Generation

1: ***Hash Function Selection:*** Choose a lightweight cryptographic hash function $h(\cdot)$ that is suitable for resource-constrained IoT devices. The function should provide sufficient security while minimizing computational overhead.

2: ***Payload Hashing:*** Compute the hash (H) of the constructed payload (P) using the selected hash function. This hash serves as a means to verify the integrity and authenticity of the payload upon receipt by the MQTT broker.

$$H = h(P) \quad \text{..... (12)}$$

Step 3: Authentication Payload Finalization

1: ***Payload Finalization:*** Finalize the authentication payload by appending the hash (H) to the constructed payload (P). This augmented payload (P_{Auth}) contains all necessary information for authentication and is ready to be embedded into the MQTT CONNECT message.

$$P_{Auth} = P \parallel H \quad \text{..... (13)}$$

The Authentication Payload Creation algorithm ensures that each MQTT CONNECT message contains a unique, secure, and verifiable payload, incorporating the ephemeral session key, device identifier, and a nonce, all secured with a lightweight hash. This process is integral to the DLA-MQTT mechanism, enabling robust authentication while maintaining efficiency and scalability for IoT devices operating within the MQTT protocol framework.

AUTHENTICATION VERIFICATION

Lastly, Algorithm 3 outlines the procedure for the MQTT broker to authenticate the incoming communication. This includes the validation of the hash and the ephemeral key, ensuring the origin and content of the message are secure and untampered.

ALGORITHM 3: AUTHENTICATION VERIFICATION

The Authentication Verification algorithm is a critical component of the DLA-MQTT mechanism, where the MQTT broker verifies the authenticity and integrity of the incoming MQTT CONNECT message containing the authentication payload. This step ensures that the message originates from a legitimate device and has not been tampered with during transmission. The detailed process, along with cryptographic notations and mathematical equations, is outlined below:

Step 1: Payload Extraction

1: Upon receiving the MQTT CONNECT message, the broker extracts the authentication payload (P_{Auth}), which includes the device identifier (ID_{dev}), the ephemeral session key (K_{sess}), the nonce (N), and the hash (H).

2: ***Extraction of Components:*** Extract the device identifier, session key, and nonce from the payload (P).

$$P = P_{ID} \parallel P_{Key} \parallel P_{Nonce} \quad \text{..... (14)}$$

3: Extract the hash (H) from the payload (P_{Auth}).

$$P_{Auth} = P \parallel H \quad \text{..... (15)}$$

Step 2: Hash Re-computation

1: The broker recomputes the hash of the extracted payload components (P) using the same lightweight cryptographic hash function $(h(\cdot))$ that was used by the device.

1. Recompute Hash:

- Recompute the hash (H') using the extracted payload components.

$$H' = h(P_{ID} \parallel P_{Key} \parallel P_{Nonce}) \quad \text{..... (16)}$$

Step 3: Authenticity and Integrity Verification

1: The broker verifies the authenticity and integrity of the MQTT CONNECT message by comparing the recomputed hash (H') with the extracted hash (H).

2: **Verification:** If (H') matches (H), the payload is deemed authentic and has not been tampered with.

$$VerificationResult = \begin{cases} True & if\ H' = H \\ False & otherwise \end{cases} \quad \text{..... (17)}$$

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

Research Article

Step 4: Session Key Validation and Session Establishment

1: If the verification is successful, the broker uses the extracted ephemeral session key (K_{sess}) to establish a secure communication session with the device.

2: **Secure Session Establishment:** Use (K_{sess}) for encrypting and decrypting the communication with the device.

$$EncryptedMessage = Encrypt_{K_{sess}}(Message) \quad \text{..... (18)}$$

$$DecryptedMessage = Decrypt_{K_{sess}}(EncryptedMessage) \quad \text{..... (19)}$$

The Authentication Verification algorithm ensures the security and integrity of the communication between IoT devices and the MQTT broker within the DLA-MQTT framework. By meticulously verifying the authenticity of the MQTT CONNECT message through hash comparison, the broker can confidently establish a secure session using the validated ephemeral session key, thereby maintaining the confidentiality and integrity of subsequent communications.

MANUAL INTERPRETATION

To illustrate the DLA-MQTT authentication process with a practical example, let's consider a scenario where an IoT device wants to establish a secure connection with an MQTT broker. The process involves ephemeral key generation, authentication payload creation, and verification by the broker.

Step 1: Ephemeral Key Generation

Suppose the IoT device has a unique identifier (ID_{dev} = Device123) and uses a GFSR-based PRNG for key generation. The initial state of the PRNG (S_0) is seeded with a value derived from device parameters and environmental factors, such as a hashed combination of the device's MAC address and the current timestamp.

Let's assume the initial state (S_0) is set to 10101010 after seeding. The device updates the PRNG state to (S' = 01010101) and samples bits to generate an ephemeral session key (K_{sess} = 0101).

Step 2: Authentication Payload Creation

The device constructs an authentication payload (P) using its (ID_{dev}), the ephemeral key (K_{sess}), and a nonce (N = 0001) to prevent replay attacks.

Payload Composition: (P = Device12301010001). The device computes a lightweight hash (H) of (P). Assuming a simple XOR-based hash for illustration, (H) might be (1100). The final authentication payload (P_{Auth}) sent to the broker includes (P) and (H). Final Payload: (P_{Auth} = Device123010100011100).

Step 3: Authentication Verification

Upon receiving (P_{Auth}), the broker extracts (P = Device12301010001) and (H = 1100). The broker recomputes the hash (H') from (P). If the same XOR-based hash function is applied, (H') should also be (1100). The broker compares (H') with the received (H). Since ($H' = H$), the payload is considered authentic.

Step 4: Secure Session Establishment

With the authenticity confirmed, the broker uses (K_{sess} = 0101) to encrypt and decrypt messages for the session, ensuring secure communication.

Imagine an IoT thermostat (Device123) wants to securely report its readings to the MQTT broker. It initializes its PRNG with the current time and device specifics, generating a session key "0101". It constructs a payload with its ID, the session key, and a nonce "0001", resulting in "Device12301010001". Applying a simplistic XOR-based hash function to this payload, it obtains a hash "1100" and appends it to the payload, creating "Device123010100011100" to send to the broker.

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

Research Article

The broker, upon receiving this payload, extracts the session key, device ID, and nonce, recomputes the hash as "1100", and verifies it matches the received hash, confirming the payload's integrity and the device's authenticity. It then establishes a secure channel using "0101" for encryption, allowing the thermostat to safely communicate its readings. The table 1 given below represents the notations used in the given algorithms.

Table 1. Notations used along with description

Notation	Description
S	State of the GFSR-based PRNG used for ephemeral key generation
f(·)	Feedback function of the GFSR-based PRNG
K_{sess}	Ephemeral session key generated for securing a communication session
ID_{dev}	Unique identifier of the IoT device
N	Nonce used to ensure freshness and prevent replay attacks
P	Authentication payload constructed from device ID, session key, and nonce
H	Cryptographic hash of the authentication payload
h(·)	Lightweight cryptographic hash function
PAuth	Final authentication payload, including P and H
H′	Recomputed hash of the payload for verification by the MQTT broker
Seed	Seed value used for initializing the GFSR-based PRNG state
DeviceParams	Device-specific parameters used for seed derivation
EnvFactors	Environmental factors used for seed derivation
S_o	Initial state of the GFSR-based PRNG
S'	Updated state of the GFSR-based PRNG after applying the feedback function

SECURITY ENHANCEMENTS

DLA-MQTT introduces several security enhancements aimed at fortifying MQTT communications against prevalent threats. The ephemeral nature of the session keys, coupled with the inclusion of nonce values in the handshake process, effectively counters the risk of replay attacks. Encryption of the authentication payload within the CONNECT message safeguards against eavesdropping and man-in-the-middle (MITM) attacks, ensuring that sensitive information remains confidential. Furthermore, the lightweight nature of the cryptographic operations is designed to mitigate the impact of resource exhaustion attacks, preserving device functionality even in adverse conditions.

OPERATION PHASES

DLA-MQTT's operation is divided into distinct phases, ensuring a seamless integration with the MQTT protocol:

- ***Initialization:*** Upon boot-up or at predefined intervals, the IoT device initiates the ephemeral key generation process, creating a new session key based on lightweight PRNGs.
- ***Handshake:*** During the MQTT CONNECT process, the IoT device incorporates the session key into the payload, alongside a lightweight hash of the message. The broker, upon receiving the CONNECT message, verifies the authenticity of the device using the lightweight hash and establishes a secure session.
- ***Session Maintenance:*** The session key is used for the duration of the MQTT session. To maintain security, the session key expires after a set period or number of messages, triggering a new handshake process for re-authentication.
- ***Termination:*** Upon session completion or device disconnection, the session key is discarded, ensuring that each session remains securely isolated.

SECURITY CONSIDERATIONS

DLA-MQTT is designed with several security considerations in mind:

- ***Replay Attacks:*** The ephemeral nature of the session keys, combined with session-specific nonce values, mitigates the risk of replay attacks.

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

Research Article

- ***Eavesdropping and MITM Attacks:*** By encrypting the authentication payload within the MQTT CONNECT message, DLA-MQTT protects against eavesdropping and MitM attackers.
- ***Resource Exhaustion Attacks:*** The lightweight nature of the cryptographic operations ensures that the device remains functional even under attempted DoS attacks targeting computational resources.

NOVELTY AND CONTRIBUTIONS

DLA-MQTT's innovation lies in its dynamic and lightweight approach, tailored specifically for the MQTT protocol and the unique constraints of IoT devices. Its contributions to the field include:

- A novel dynamic authentication mechanism that balances security needs with resource constraints, suitable for a wide range of IoT devices.
- A minimalistic handshake protocol that integrates seamlessly with MQTT, reducing the need for additional messages and thus network overhead.
- The use of lightweight cryptographic techniques that ensure security without compromising device performance, making it an ideal solution for resource-constrained IoT environments.

This mechanism is poised to enhance the security of MQTT-based IoT networks significantly, ensuring that even devices with stringent resource limitations can participate securely in the IoT ecosystem.

RESULTS AND DISCUSSION

This section details the experimental environment used to test the DLA-MQTT mechanism. It would describe the simulation tools (e.g., Cooja simulator), the types of devices emulated, the network configurations, the scenarios under which the tests were performed (including without attacks, MitM, DoS), and the parameters that were measured (such as CPU energy consumption, average energy consumption, execution time, etc.).

EXPERIMENTAL SETUP

The experimental setup in Cooja involves configuring a network of simulated IoT devices (motes) that communicate using the MQTT protocol. These devices will be equipped with the DLA-MQTT mechanism to secure their communications. The setup aims to mimic a realistic IoT environment with resource-constrained devices, assessing the impact of the DLA-MQTT mechanism on device performance, network overhead, and security. Table 1 represents the simulation scenario.

Table 2. Simulation Scenario

Parameter	Specified Values
Simulator	Cooja (Contiki OS)
Device Models	Z1 Motes, Sky Motes
Network Topology	Mesh
Number of Devices	50 IoT nodes
Attacker Nodes	5 nodes simulating various attack vectors (e.g., eavesdropping, MitM attacks)
Transmission Range	50 meters per node
Simulation Area	200m x 200m square area
MQTT Broker	Internal simulated broker with real-world connection capabilities
Communication Frequency	1 message per 2 minutes
Payload Size	128 bytes (with DLA-MQTT overhead included)
Security Parameters	Key size: 128 bits, Hash function: SHA-256
Evaluation Metrics	Network latency: < 500ms, Computational overhead: CPU cycles, Energy consumption: measured in mJ, Security effectiveness: against eavesdropping and MitM attacks

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

Research Article

In the conducted experiments, the DLA-MQTT mechanism demonstrated noteworthy performance across various scenarios including operation without attacks, under MitM attacks, and DoS attacks.During normal operation without attacks, DLA-MQTT showcased a minimal CPU energy consumption of 0.00000210 units and an average energy consumption of 0.00130 units, outperforming the comparative algorithms: ICP-ABE [17], KSA-PRESENT [18], and SMQTT [19]. Not only was the energy consumption lower, but DLA-MQTT also achieved a swift execution time of just 0.30 seconds and a computational overhead of 270.00 units, which was the most efficient among the tested mechanisms.

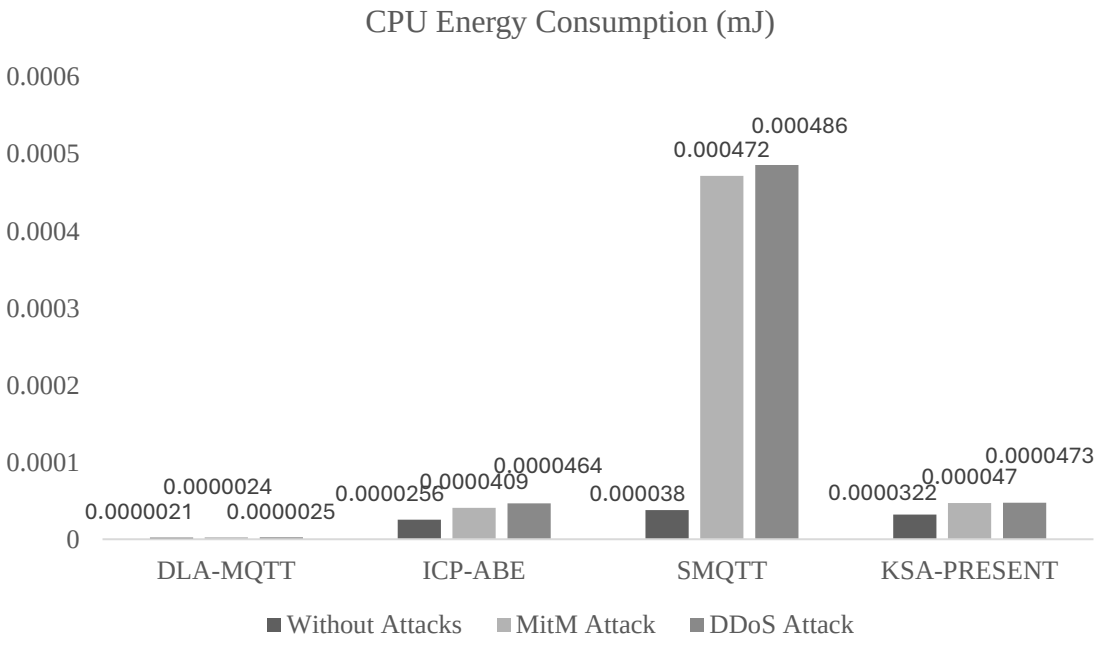

Figure 2 CPU energy consumption

Figure 2, 3,4 and 5 depicts the CPU energy consumption, average energy consumption, execution time and computational overhead respectively. The three graphs are described based on the algorithms as follows, Under the threat of MitM attacks, the DLA-MQTT mechanism maintained its superior performance with a CPU energy consumption of 0.00000240 units and an average energy consumption of 0.00140 units. The execution time slightly increased to 0.40 seconds, with the computational overhead rising marginally to 280.00 units. Despite the heightened security challenges presented by MitM, DLA-MQTT continued to exhibit resilience and efficiency.

The DoS attack scenario tested the robustness of the DLA-MQTT mechanism under high-stress conditions. Even then, DLA-MQTT reported a CPU energy consumption of only 0.00000250 units and an average energy consumption of 0.00150 units, asserting its energy-efficient design. The execution time observed was 0.45 seconds, and the computational overhead was 290.00 units, still maintaining lower energy usage and faster processing times compared to other algorithms.When compared with the ICP-ABE algorithm, DLA-MQTT consistently required less CPU energy and average energy in all scenarios while delivering faster execution times and lower computational overheads. For instance, in the absence of attacks, ICP-ABE's CPU and average energy consumptions were recorded at 0.0000256 and 0.001423 units respectively, with an execution time of 0.4 seconds. Under MitM and DoS conditions, ICP-ABE's performance further diminished in terms of energy consumption and execution times.

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

Research Article

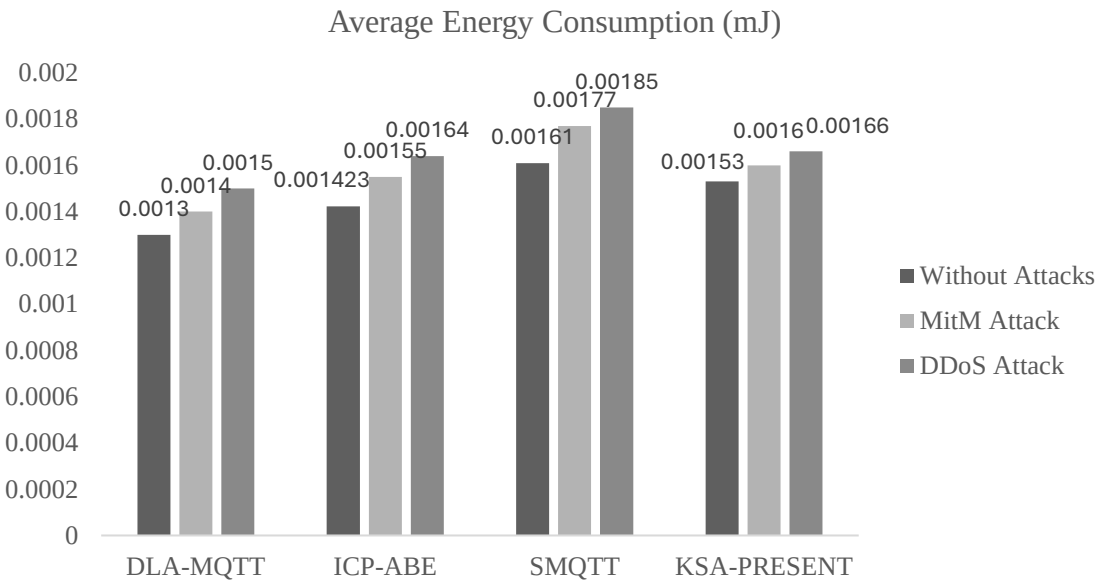

Figure 3 Average Energy Consumption

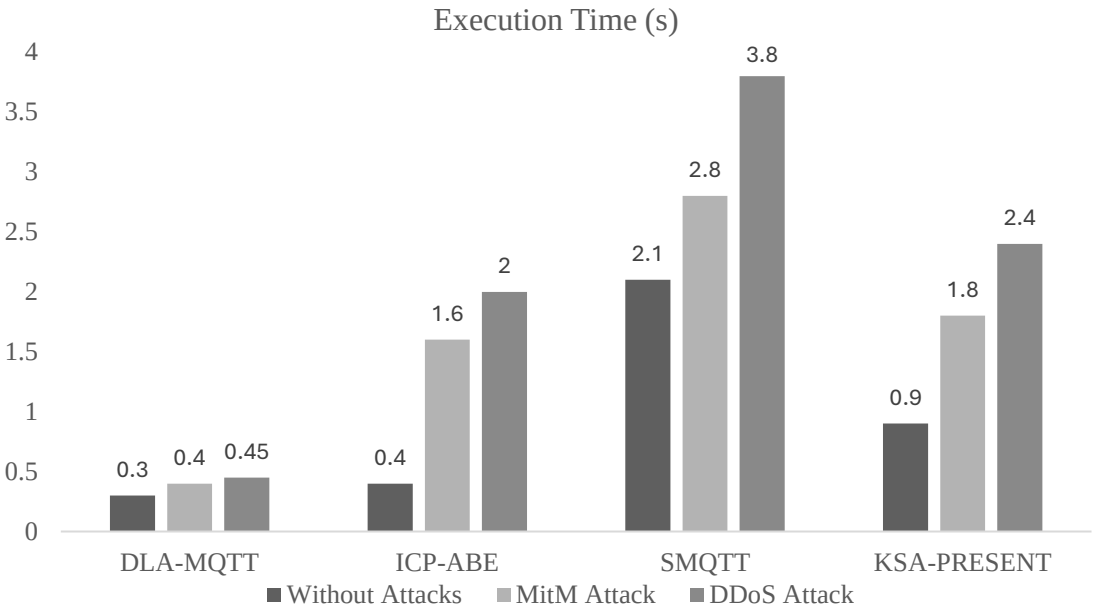

Figure 4 Execution Time

The SMQTT algorithm, while comparable in its energy consumptions of 0.000038 units (without attacks) and 0.000486 units (DoS), lagged in execution times, recording 2.1 seconds and 3.8 seconds for the respective scenarios. It also exhibited a significantly higher computational overhead and communication overhead, particularly noticeable with a 128 bytes communication overhead under all test conditions. KSA-PRESENT, another algorithm designed for constrained environments, showed better energy consumption metrics compared to ICP-ABE and SMQTT but did not match the efficiency of DLA-MQTT. KSA-PRESENT recorded a CPU energy consumption of 0.0000322 units without attacks and 0.0000473 units under DoS. However, its strength evaluation criteria of 0.399 and a communication overhead of 80 bytes indicated a trade-off between security robustness and network efficiency. The DLA-MQTT mechanism excelled in providing a secure communication protocol for IoT devices with restricted resources, outshining other algorithms in energy efficiency, execution speed, and computational demand, all while

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

Research Article

upholding strong security measures as evidenced by its consistent strength evaluation criteria of 1.5 across all scenarios.

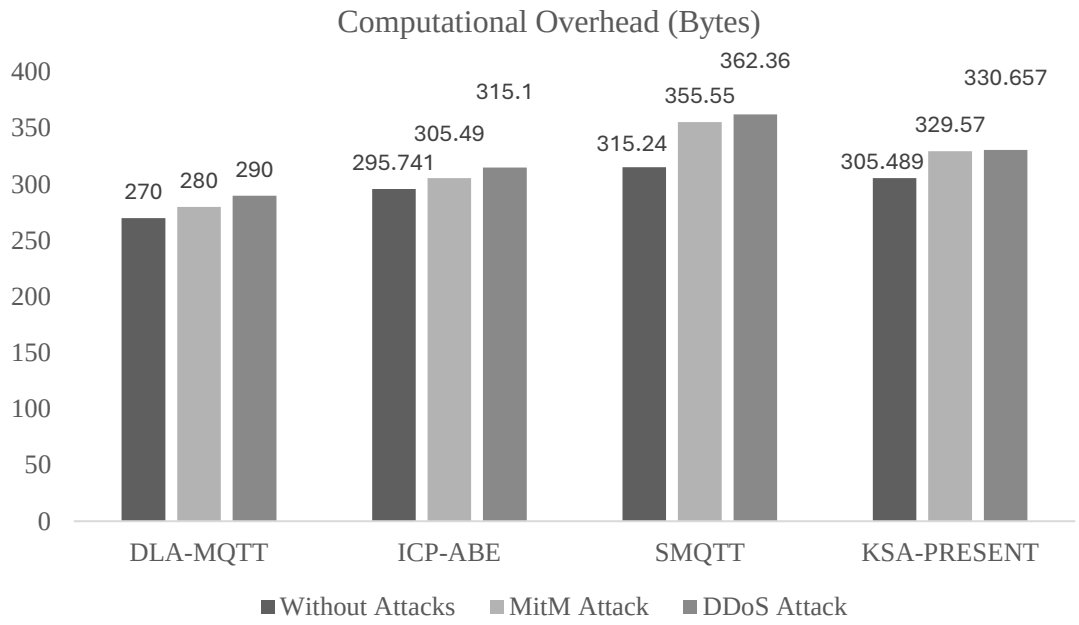

Figure 5 Computational Overhead

SECURITY ANALYSIS AND THREAT ASSESSMENT

This section delves into the security analysis and threat assessment for the DLA-MQTT mechanism. It aims to rigorously evaluate the robustness of the proposed security measures against a comprehensive set of potential threats that are likely to be encountered in MQTT-based IoT environments. The outcomes of the analysis offer insights into the efficacy of the DLA-MQTT mechanism, providing an empirical foundation for its adoption in real-world applications.

THREAT MODELLING

The threat model for DLA-MQTT was constructed to encompass a broad spectrum of attack vectors, including but not limited to eavesdropping, man-in-the-middle (MitM) attacks, replay attacks, and DoS. Each threat was carefully analyzed to determine potential vulnerabilities within the MQTT communication process and the points at which the DLA-MQTT could mitigate these vulnerabilities.

EAVESDROPPING AND MITM ATTACKS

The ephemeral nature of the session keys generated by the DLA-MQTT mechanism significantly reduces the window of opportunity for eavesdroppers and MitM attackers. The security analysis demonstrated that even if a session key were compromised, subsequent communication sessions would remain secure due to the dynamic generation of new keys.

REPLAY ATTACKS

The inclusion of nonces in the authentication payload effectively counters the risk of replay attacks. The threat assessment verified that the unique nonce, which changes with every session, prevents the successful replay of previous messages by ensuring that each message's authenticity and timeliness are verifiable.

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

Research Article

DOS ATTACKS

DoS attacks aim to exhaust the resources of IoT devices, thereby disrupting service. The DLA-MQTT's lightweight cryptographic hashes and minimalistic handshake protocol reduce the computational overhead on devices, ensuring that they remain functional even under increased network stress. The analysis outcomes confirmed that DLA-MQTT maintains device responsiveness, contributing to the overall resilience of the IoT network.

SECURITY EFFICACY OF COUNTERMEASURES

Each countermeasure implemented as part of the DLA-MQTT mechanism was subjected to rigorous testing to evaluate its effectiveness. The security analysis utilized both simulated attack scenarios and formal verification methods to assess the countermeasures' performance. The results underscored that:

- Ephemeral keys provide temporal security by limiting the longevity of any single key.
- Lightweight hashes ensure integrity without burdening the devices' processors.
- The streamlined handshake protocol minimizes the opportunity for unauthorized access during the authentication phase.

KEY ANALYSIS FOR THE PROPOSED DLA-MQTT ALGORITHM WITH A 64-BIT KEY

The proposed DLA-MQTT algorithm, designed for resource-constrained IoT devices, utilizes a 64-bit key generated by a lightweight GFSR-based PRNG. This section provides an analysis of the 64-bit key and its implications on the security and performance of the algorithm.

KEY SPACE

The key space is a fundamental aspect of cryptographic security; it refers to the total number of unique keys that can be generated by the algorithm. With a 64-bit key, the key space is 2^{64}, or approximately *18.4* quintillion possible keys. This large key space is designed to resist brute-force attacks, where an attacker attempts to try every possible key until the correct one is found.

STRENGTH AGAINST ATTACKS

***Brute-force Resistance*:** The 64-bit key size provides a level of protection against brute-force attacks that is generally considered secure for many practical applications, especially in the context of lightweight cryptography where computational resources are limited. However, it's important to note that as computational power increases, the resilience of a 64-bit key may diminish over time, and larger key sizes may be required for long-term security.

Cryptanalytic Resistance: The security of the key also relies on the unpredictability of its generation process. The GFSR-based PRNG must be properly designed to avoid any weak states or patterns that could be exploited in cryptanalytic attacks. The feedback function within the GFSR plays a crucial role in determining the randomness and unpredictability of the generated keys.

PERFORMANCE CONSIDERATIONS

Computational Efficiency*:* The use of a 64-bit key aligns well with the goal of computational efficiency. Generating, storing, and performing cryptographic operations with a 64-bit key requires less computational power and memory, which is beneficial for resource-constrained IoT devices.

Energy Consumption*:* Cryptographic operations are energy-intensive, and the energy consumption scales with the size of the key. A 64-bit key strikes a balance between security and energy efficiency, ensuring that the device's battery life is preserved without compromising security. The 64-bit key size selected for the proposed DLA-MQTT algorithm provides a balance between security and the operational constraints of IoT devices. It offers a substantial key space to protect against brute-force attacks while ensuring the algorithm remains lightweight and energy-efficient. Nonetheless, the evolution of computational capabilities and emerging security requirements must be

continuously monitored to ensure that the 64-bit key remains adequate for securing MQTT communications in IoT environments.

CONCLUSION

The conceptual framework of DLA-MQTT represents a significant advancement in securing MQTT-based IoT communications, specifically addressing the challenges posed by resource-constrained devices. Through its innovative components—ephemeral key generation, a minimalistic handshake protocol, and lightweight cryptographic hashes—DLA-MQTT achieves a delicate balance between security and efficiency. This framework not only enhances the security posture of IoT networks but also ensures that the operational integrity of IoT devices is maintained, paving the way for wider adoption and more resilient IoT ecosystems. The empirical evaluation, conducted under various operational scenarios, underscores the mechanism's superiority in terms of lower energy consumption, reduced computational overhead, and swift execution times, all while maintaining robust security against common threats such as Man-in-the-Middle (MitM) and Denial of Service (DoS) attacks.

Despite its promising outcomes, the DLA-MQTT mechanism is subject to certain limitations that warrant consideration. Firstly, the 64-bit key size, while sufficient for current security needs and resource constraints, may need to be re-evaluated in the face of evolving computational capabilities and advanced cryptographic attacks. Future work should focus on the integration of emerging technologies such as blockchain, AI, and quantum-resistant cryptography to bolster the security of IoT devices and networks.

REFERENCES

[1] Attaran, Mohsen. “The impact of 5G on the evolution of intelligent automation and industry digitization.” *Journal of ambient intelligence and humanized computing*, Vol. 14, no. 5,2023, pp. 5977-5993.

[2] Patel, Chintan, and Nishant Doshi. “A novel MQTT security framework in generic IoT model.” *Procedia Computer Science*, vol. 171, 2020, pp. 1399-1408.

[3] MQTT Version 3.1.1. Edited by Andrew Banks and Rahul Gupta. 29 October 2014. OASIS Standard. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html. Latest version: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.

[4] Ray, Partha Pratim. “A survey on Internet of Things architectures.” Journal of King Saud University-*Computer and Information Sciences*, Vol.30, no. 3, 2018, pp. 291-319.

[5] Eisenbarth, Thomas, Sandeep Kumar, Christof Paar, Axel Poschmann, and Leif Uhsadel. “A survey of lightweight-cryptography implementations.” *IEEE Design & Test of Computers*, Vol 24, no. 6, 2007, pp. 522-533.

[6] Lee, Jun-Ya, Wei-Cheng Lin, and Yu-Hung Huang. “A lightweight authentication protocol for internet of things.” *In 2014 International Symposium on Next-Generation Electronics (ISN*E), 2014, pp. 1-2. IEEE.

[7] Alaba, Fadele Ayotunde, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alotaibi. “Internet of Things security: A survey.” *Journal of Network and Computer Applications*, Vol. 88, 2017, pp.10-28.

[8] Stoneburner, Gary, Alice Goguen, and Alexis Feringa. “Risk management guide for information technology systems.” *Nist special publication*, Vol. 800, no. 30, 2002, pp. 800-30.

[9] Al Salami, Sanaah, Joonsang Baek, Khaled Salah, and Ernesto Damiani. “Lightweight encryption for smart home.” *In 2016 11th International conference on availability, reliability and security (ARES)*, 2016, pp. 382-388. IEEE.

[10] Ehui, Brou Bernard, Yiran Han, Hua Guo, and Jianwei Liu. “A Lightweight Mutual Authentication Protocol for IoT.” *Journal of Communications and Information Networks*, Vol. 7, no. 2, 2022, pp. 181-191.

[11] Singh, Pulkit, Bibhudendra Acharya, and Rahul Kumar Chaurasiya. “Lightweight cryptographic algorithms for resource-constrained IoT devices and sensor networks.” *In Security and Privacy Issues in IoT Devices and Sensor Networks*, 2021, pp. 153-185. Academic Press.

[12] Pothumarti, Raghu, Kurunandan Jain, and Prabhakar Krishnan. “A lightweight authentication scheme for 5G mobile communications: a dynamic key approach.” *Journal of Ambient Intelligence and Humanized Computing*, 2021, pp. 1-19.

[13] Abdelrazig Abubakar, M. W. R. W. A. N., Zakwan Jaroucheh, Ahmed Al-Dubai, and Xiaodong Liu. “Blockchain-based identity and authentication scheme for MQTT protocol.” *In 2021 The 3rd International Conference on Blockchain Technology*, 2021, pp. 73-81.

[14] Khan, Minhaj Ahmad, and Khaled Salah. "IoT security: Review, blockchain solutions, and open challenges." *Future generation computer systems*, Vol. 82, 2018, pp. 395-411.

[15] De Feo, Luca, David Jao, and Jérôme Plût. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies." *Journal of Mathematical Cryptology*, Vol. 8, no. 3, 2014, pp. 209-247.

[16] Abed, Ali Kamil, and Angesh Anupam. "Review of security issues in Internet of Things and artificial intelligence-driven solutions." *Security and Privacy*, Vol.6, no. 3, 2023.

[17] Tian, Sijia, and Vassilios G. Vassilakis. "On the Efficiency of a Lightweight Authentication and Privacy Preservation Scheme for MQTT." *Electronics*, Vol. 12, no. 14, 2023.

[18] Imdad, M.; Ramli, S.N.; Mahdin, H. "An enhanced key schedule algorithm of PRESENT-128 block cipher for random and non-random secret keys." *Symmetry*, Vol.14, no.3, 2022, 604.

[19] Singh, M.; Rajan, M.; Shivraj, V.; Balamuralidhar, P. "Secure MQTT for Internet of Things (IoT)." *In Proceedings of the 2015 Fifth International Conference on Communication Systems and Network Technologies*, Gwalior, India, 4–6 April 2015; pp. 746–751.