IoT-31652-2023

Enabling Lightweight Device Authentication in Message Queuing Telemetry Transport Protocol

Narasimha Swamy S, *Graduated Student Member, IEEE* Dheeraj Manirathnam Anna, Vijayalakshmi M N and Kota Solomon Raju, *Senior Member, IEEE*

***Abstract*—Recent advancements in IoT have led to emergence of fascinating breakthroughs in diverse applications. Nowadays, the use-cases of smart home systems are augmenting as they provide functionalities like real-time monitoring and high degree of remote control. MQTT protocol is one of the most widely used messaging protocols in IoT-based applications including smart homes. This protocol lacks required security features owing to which, the intruders can launch variety of attacks easily. Stirred by this, we proposed a lightweight device authentication scheme for MQTT protocol. In this work, publisher/ subscriber, and broker use lightweight cryptographic operations to enable device authentication. Also, this mechanism utilizes the lightweight cryptographic keys such as One-time Key (OT_{Key}) and Tokens (T_i) to complete registration and authentication process respectively. Compared to other protocols, our approach reduces both communication and computation costs while maintaining the security demands. We put a prototype into practice to assess the performance of the proposed authentication mechanism. Further, we perform the formal analysis of the proposed authentication mechanism using AVISPA protocol analyzer tool. The proposed security mechanism is resistant to various attacks such as replay attack, device impersonate attack, malicious node attack, etc., and it enables the security features like device authentication and device anonymity in smart homes.**

***Index Terms*- Cuckoo Filters, Device Authentication, Edge Computing, IoT, MQTT, Security, Smart Homes**

NOMENCLATURE

AVISPA	Automated Validation of Internet Security Protocols and Applications
APIs	Application Interfaces
AMQP	Advanced Message Queuing Protocol
CoAP	Constrained Application Protocol
DDS	Data Distribution Services
MQTT	Message Queuing Telemetry Transport
HTTP	Hypertext Transfer Protocol
HMAC	Hash-based Message Authentication Code
IoT	Internet of Things
IBM	International Business Machines
SSL	Secure Socket Layer
TLS	Transport layer Security
VANETS	Vehicular Ad-Hoc Networks
XMPP	eXtensible Messaging and Presence Protocol
WiFi	Wireless Fidelity

Narasimha Swamy S is with the Academy of Scientific and Innovative Research (AcSIR), Ghaziabad, CSIR- National Aerospace Laboratories, Bengaluru and Department of Artificial Intelligence and Machine Learning, R V College of Engineering, Bengaluru, India (e-mail: narasimha.rmgm@gmail.com).

Dheeraj Manirathnam Anna and *Vijayalakshmi M N* are with the Department of Artificial Intelligence and Machine Learning, R V College of Engineering, Bengaluru, India (e-mail: dheerajma.ai21@rvce.edu.in, vijayalakshmi@rvce.edu.in).

Kota Solomon Raju is with the AcSIR, Ghaziabad and CSIR- National Aerospace Laboratories, Bengaluru, India (e-mail: kotasolomonraju@gmail.com)

I. INTRODUCTION

THE concept of IoT was introduced with the aim of improving the quality of life in the contemporary world [1]. IoT is an interconnection between heterogeneous devices connected to the global network through the internet and provides application services to anything at any-time, and anyplace [2]. It has been predicted that the number of devices connected to the internet is expected to reach 21 billion by 2025 [3]. Sensing and actuation, storing, processing, and sharing are the typical properties of IoT, and they assist in saving time in daily activities [4]. These advantages made IoT become the core technology for applications like smart homes, smart cities, smart health, smart transportation, smart agriculture, and others [5].

Smart home is the fastest growing IoT application, which provides application services like, monitoring, automation, and surveillance [6] while ensuring convenience, cost saving and security to the residents. The use of smart devices like lights, thermostats, fans, refrigerators, ovens, television, smart phones, laptops, tablets, doorbells, IP camera, and other devices are more prevalent in smart home implementations. The smart devices can be monitored and controlled using a handy Graphical User Interface (GUI), like a web browser or mobile applications. Moreover, they also allow for Machine-to-Machine (M2M) and Human-to-Machine (H2M) communications. An apt example for M2M communication would be a thermostat sending a notification to an air conditioner when the temperature exceeds 45^0 C. Similarly, instances of H2M communication include passing voice commands to household devices like infotainment systems, washing machines, water heaters, etc. Communication among these devices uses protocols like MQTT, CoAP, DDS, XMPP, AMQP, Websocket, HTTP, and Restful APIs [7]. However, wide-scale implementation of smart homes poses various challenges such as data confidentiality, device management, and non-availability of devices, i.e., security in general.

Accelerated growth of household IoT devices pose security

and privacy difficulties; this is one of the main concerns about smart homes since they produce significant amounts of sensitive and private data. It is revealed from the study that approximately 70% of smart home devices are vulnerable to a variety of cyberattacks [8]. Common cyber-attacks on smart home network include man-in-the-middle, Denial of Service (DoS), identity theft, data theft, and Distributed Denial of Service (DDoS) attacks [9].

Ensuring security features in any of the IoT applications is highly challenging as compared to the traditional networking applications; this is due to *1. Diverse ecosystem:* IoT applications comprise of heterogeneous devices with their own security characteristics, and protocols, *2. Limited computing and communication resources:* IoT devices often have limited storage and processing abilities. Due to this, providing secure communication and information with proper encryption technique is challenging. Most of the time IoT devices collect and share personal data; this data can be the target of cyber criminals, *3. User neglects:* Often, IoT users rely on the default passwords. Also, users fail to take the initial security measures such as firmware updates, authentication, etc. and 5. *Physical access:* Sometimes, IoT devices are easily accessible physically, due to this entire IoT network might be jeopardized. For instance, if a burglar has physical access to the smart door lock, an attacker can alter the access patterns and password. Figure 1 shows the various attacks that exist in the IoT three layered architecture.

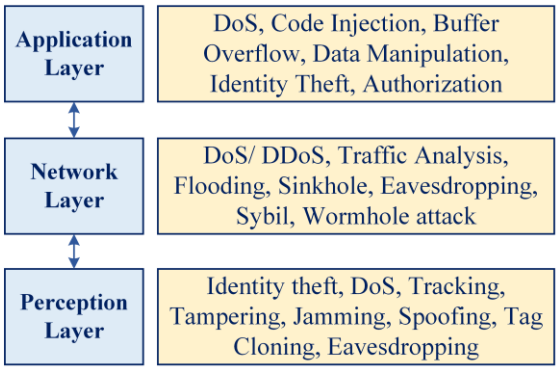

Fig. 1. Attacking Scenario in MQTT Protocol

There exists several security frameworks, protocols, and mechanisms to protect the IoT. However, these security entities require significant improvements in storage, communication, and energy aspects. The organizations like Internet Engineering Task Force (IETF), academicians and researchers are working hard to scale up the security features of the IoT networks. In conjunction with everything said above, we propose a lightweight device authentication scheme for MQTT. To implement and verify the proposed mechanism we use the smart home environment. Publishers, broker, and subscribers are the key communication parties of the MQTT protocol. In the proposed mechanism, first publishers and subscribers must be registered with the broker to send and receive the updates. For each data transaction, the broker verifies the publisher, locates the matching subscribers in the cuckoo filter, and then sends the data to the subscribers. The subscriber then consumes the data after authenticating the broker.

A. Motivation

MQTT is an open standard machine-to-machine protocol developed by IBM in 1999 [10]. Today, MQTT protocol is being used extensively in IoT environments. This is due to its features like lightweight, less computing power and network bandwidth requirements. However, the protocol supports only limited security services because of which distinct security lapses have been pointed out in its implementation. Regular security risks associated with the MQTT protocol are those of confidentiality, authorization, and authentication; these security issues were discussed in detail by Giuseppe Nebbione et al. [11]. Figure 2 depicts an illustration of an attack scenario over communications under the MQTT protocol. This work primarily focuses on enriching the existing MQTT protocol with lightweight device authentication scheme to safeguard devices from unauthorized access in an IoT environment.

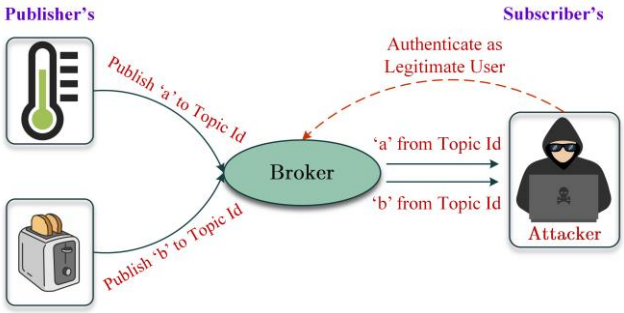

Fig. 2. Attacking Scenario in MQTT Protocol

B. Major Contributions

The current smart home environments require a strong and energy-efficient security algorithm to protect personal and sensitive data. Security in smart homes is one of the major domains of interest for research. This article presents the following on these grounds:

1. Overview of the IoT, smart homes, and edge computing including definitions and architecture.
2. Provided state of art-of the security mechanisms in smart homes.
3. Proposed reliable, and secure lightweight device authentication scheme for MQTT based smart homes.
4. Minimize computational overhead by utilizing lightweight cryptographic operations instead of traditional cryptographic operations.
5. Analyzing the proposed protocol formally using Automated Validation of Internet Security Protocols and Applications (AVISPA) protocol analyzer tool.

C. Organization of the Paper

The remaining part of the paper is structured as follows: Section II briefs the analysis of the IoT security mechanisms and dedicated security mechanisms developed for MQTT protocol. Section III presents the significant concepts, technologies, and architectures used in this research. Section IV summarizes the system design, system requirements, and notations used in this research. Section V discusses the implementation of the lightweight MQTT based device authentication scheme for smart homes. Then, Section VI highlights the informal and formal security analysis of the proposed device authentication protocol. Next, Section VII presents the results and discussion in detail. Finally, Section VIII concludes the proposed work.

II. Literature Review

MQTT is employed in a wide range of applications due to its features such as lightweight, asynchronous communication, Quality of Service (QoS) levels, low power consumption, and scalability. On the other hand, MQTT protocol is vulnerable to various cyber-attacks because it is developed for trustworthy IoT networks. Here, we provide the recent developments in the field of MQTT security.

Eduardo Buetas Sanjuan et al. proposed an encryption-based authentication scheme, which secures the MQTT communication in IoT networks; this scheme is implemented at publishers, subscribers, and brokers [12]. *Wei-Tsung Su et al.* proposed a security scheme to provide Things-to-Things and secure data exchange. The security scheme was implemented over raspberrypi3 [13]. *SeongHan Shin et al.* proposed an AugMQTT security framework, which provides potential security against passive attacks. The proposed framework does not require any certificate validation checks or revocation checks [14]. *Seyed Ahmad Soleymani et al.* built a secure trust model in vehicular ad hoc networks to authenticate the vehicles within the transmission range. The trust model comprises of distinct types of vehicles and fog nodes. Fog nodes in the trust model are used to conduct a series of security checks in VANETS using fuzzy rules. An authentication algorithm is designed and deployed in the fog nodes to authenticate the vehicles [15]. *Santiago Hernandez Ramos et al.* proposed a novel fuzzing technique to increase the security of the MQTT devices. The authors have evaluated the efficiency of the protocol using mosquito implementation. The experiments so conducted have revealed acceptable processing time [16]. *S A Abdymanapov et al.* proposed an expert system to assess the information security risk in learning management systems using fuzzy logic [17]. *Serin V. Simpson et al.* designed a secure approach to detect co-operative extortion attacks in smart cities. This approach uses the advantages of edge computing for timely identification of the malicious nodes. Fuzzy rules are implemented in the edge computing devices to mitigate and isolate the suspected adversary nodes in each smart city network. The extracted suspected nodes are re-examined based on the trust values generated by the trust model [18]. *Dooho Choi et al.* proposed a concept of two-factor fuzzy commitment scheme: - this uses the intrinsic and extrinsic factors of an IoT device to counter-measure the adversaries who steal the IoT devices physically and later try to retrieve the information from the device. To implement the proposed algorithm the authors have used PUF as an intrinsic noisy source and image from the camera as an external noisy source; this decreases the chance of extraction of right cryptographic key for the purposes of information extraction [19]. *Özlem Yerlikaya et al.* proposed an authentication and authorization mechanism for MQTT protocol. The authors have used the HMAC-based one-time password along with the one-time passwords for authentication. Also, they have used Advanced Encryption Standard (AES) for data confidentiality [20]. To improve the security of the MQTT protocol *Sijia Tian et al.* presented a Ciphertext Policy-Attribute-Based Encryption method which is combined with the existing PRESENT lightweight symmetric encryption technique [21]. *Abdur Rahman et al.* listed the analysis of device and data security issues in IoT environment. Also, the authors have modified the existing MQTT protocol using attribute encryption and elliptic curve cryptography [22]. *Shweta Iyer et al.* proposed two stage security architecture to enable the security of the MQTT protocol. In the first stage data is encrypted using the lightweight symmetric block cipher. In the later stage lightweight hash function is utilized to confirm the data integrity [23]. *Hassan Kurdi et al.* propose a fog computing based lightweight mutual authentication scheme for Industrial IoT (IIoT) applications. The authentication mechanism is run by the authentication manager placed in the broker [24]. Additionally, Table I shows the security-related work done on the MQTT protocol.

In precise, the existing works use the traditional cryptographic operation to achieve security; this type of implementations requires high computational and processing power. Also, these methods require more storage capabilities. Since most of the IoT devices are in remote location; it is hard to replace batteries oftentimes. Most of the existing works dealt with user authentication only. Keeping all these issues in mind, we proposed a lightweight device authentication scheme for MQTT protocol.

III. Background

This section describes the key concepts, protocols, and computing architectures used in the proposed work.

A. Concept of Smart Home

IoT plays a significant role in transforming traditional homes into smart homes by providing features like sensing, storing, analyzing, and sharing of data [34]. Also, smart home applications offer the various services such as

1) Home Automations

TABLE I. SECURITY IN MQTT PROTOCOL

Ref.	Solution Proposed	Gist of the Article	Application
[25]	Dynamic multi-broker Framework	▪ Enables the dynamic authentication and authorization. ▪ Implemented and evaluated the proposed framework using small scale IIoT testbed. ▪ The IIoT testbed composed of Programmable Logic Controls (PLCs), IIoT Gateways and MQTT brokers. ▪ Authors have not evaluated the effectiveness of the framework against the various cyber attacks	Industrial Internet of Things (IIoT)
[26]	Security Scheme Based on Enhanced Symmetric Algorithm	▪ Proposed Robust Security Scheme, which can be used in the MQTT-based IoT networks. ▪ This security mechanism uses both a Dynamic variant of the Advanced Encryption Standard (D-AES) and Key Policy Attribute based Encryption (KP-ABE) cryptographic algorithms to distribute the secret key to the publisher and subscriber. ▪ The mechanism enables the confidentiality and access control in MQTT.	-
[27]	Software-Defined Perimeter (SDP) Framework	▪ SDP enable the one more layer of security with or without using the SSL/TLS. ▪ SDP uses the traditional login mechanism i.e., single-packet authorisation (SPA) process, username, and password. ▪ The mechanism resilient against Denial-of-Service Attack (DoS)	-
[28]	Service Oriented Architecture	▪ The architecture enables the authentication, data discovery and distribution. ▪ The authentication service is based on the JSON Web Tokens and unique public key known to the devices. ▪ Authors have not evaluated the effectiveness of the architecture against the various cyber attacks	-
[29]	Group Communication Framework	▪ The proposed framework facilitates the group key distribution and secure group communication of MQTT messages. ▪ The proposed mechanism eliminates the message overhead by eliminating the SSL/ TLS from the communication	-
[30]	Secure End-to-End MQTT (SEEMQTT) Framework	▪ This framework allows the publishers to encrypt the published message. ▪ The framework uses the Shamir's key sharing scheme to generate and share the key with the peers. ▪ The framework also uses the Identity-based Encryption (IBE) to create the secure links between publishers and KeyStore. ▪ The subscriber retrieves the data based on the credential defined by the publishers.	-
[31]	Secure Communication System	▪ Enables the confidentiality and integrity features in MQTT-based IoT networks (TLS 1.3). ▪ The authors have not evaluated performance of the proposed system.	Microgrid
[32]	Lightweight Key-Sharing System	▪ The authors have used (k, n) threshold secret-sharing scheme to share the secret key between the publisher and subscriber. ▪ This key sharing mechanism was developed to eliminate the message and communication overhead caused by the SSL/ TSL.	-
[33]	MQTT aware TLS protocol (MQTLS)	▪ MQTLS defines the Client-to-Broker-to-Client (CBC) security semantics for the publisher/ subscriber model. ▪ The performance overhead of the proposed protocol is high during the initial setup.	-

It deals with the control and management of home appliances and other devices remotely.

2) Health Assistance

Health assistance brings efficient healthcare services to the general public's doorstep. Health assistance in smart homes involves collaboration between stakeholders like

doctors, nursing staff, elderly people/ patients, and laboratory personnel [35].

3) Smart Gardening

Renders gardening chores more accessible and efficient. For example, implementation of automated watering systems. Adoption of continuous monitoring techniques in smart gardening offers several benefits, including enhanced efficiency in time, water, and energy usage, as well as extending the lifespan of plants.

4) Security and Surveillance

IoT security systems employ sensors and cameras to collect data, monitor areas from a distance, and identify possible security breaches. Security and surveillance systems offer real-time updates, data analysis, and automation to enhance the effectiveness of security measures. These systems may incorporate functionalities such as intruder detection, video surveillance, access control, environmental monitoring, and alarm systems.

B. Edge Computing

Edge computing is an emerging computing technology, which draws upon the services offered by cloudlets and cloud computing services. These characteristics of edge computing reduce latency and transforms the communication between the IoT applications into a smoother one [37]. In this paper edge-server architecture is used; in which the IoT nodes are directly connected to edge nodes and utilizes the services offered by the edge node.

C. MQTT Protocol

MQTT is a publisher-subscriber based messaging protocol [38], used to exchange data between IoT devices. Both publishers and subscribers function as clients and the broker is primarily a server. The protocol is used to minimize requirements of network bandwidth and device resource while ensuring reliable delivery. Figure 3 shows the message flowing between the publisher, subscriber, and broker.

D. Security in MQTT

Currently, the MQTT protocol has been seeing widespread implementation for IoT based communications. According to a survey, MQTT provisions multiple authentication mechanisms including TLS. However, these security services need more resources. to protect communication between the IoT devices effectively. According to the survey, the major security threats of devices employing MQTT communication are those of authentication and message encryption [3][5].

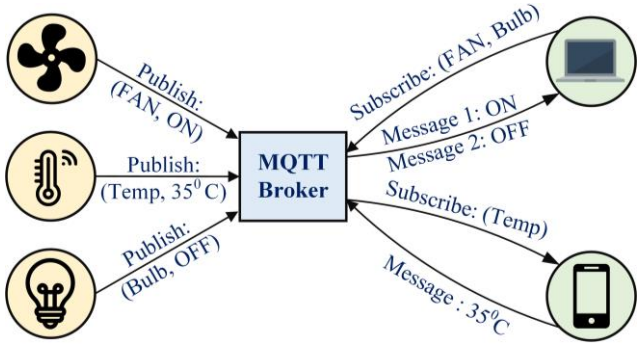

Fig. 3. Message Exchange in MQTT Protocol

Authentication: Within the MQTT protocol, the broker does not check the identities of the publisher and subscriber which leads to possible malicious and unauthorized access to the devices. These vulnerabilities can in turn create problems like tampered messages and overloading of the broker; this broadly reduces the performance of the broker [3][5].

Message Encryption: MQTT protocol does not give provision for message encryption by default. The participants i.e., publisher, subscriber, and broker, undertake communication of plain text; enabling intruders to eavesdrop on the communication and tamper with the messages [3][5].

E. Device Authentication

In an era when the world is getting increasingly connected device authentication is one of the promising solutions to enable protection of IoT devices and networks from unauthorized access. It also augments trust among the communication devices in a network. Device authentication mechanisms must be efficient (i.e., must allow for faster computations and lower memory footprints). Despite the emergence of various security standards, the IoT industry still does not draw upon a universal and reliable security framework [39].

F. Cuckoo Filters

Cuckoo filter is a probabilistic light-weight data structure used for querying purposes. The cuckoo filters use cuckoo hash tables which comprise of, say 'm' buckets. These buckets are used to store fingerprints of an item to be inserted. To avoid data collisions cuckoo uses partial key cuckoo hashing. Cuckoo filters support operations like insertion, deletion, and fingerprint lookup [40]. A cuckoo hash table is a collection of buckets in which each item meant to be inserted is mapped to two buckets (hence the filter maintains the two candidate buckets) based on the values produced by the two hash functions. During insertion, the fingerprint (Equation 1) of the item can be placed in either of the buckets. Cuckoo filter randomly deletes any of the fingerprints and re-inserts the deleted fingerprint into its alternative bucket when both the candidate buckets are occupied. This reallocation process ends when the fingerprint is inserted successfully or when the number of reallocation processes reaches the maximum threshold. During reallocation, the alternative candidate bucket can be identified using XOR operation between the current candidate bucket and deleted fingerprint (Equation 2). To be more precise, the two candidate buckets for the item's fingerprint are calculated as follows:

$$h_1(a) = hash(a) \quad (1)$$

$$h_2(a) = h_1(a) \oplus hash(fingerprint(a)) \quad (2)$$

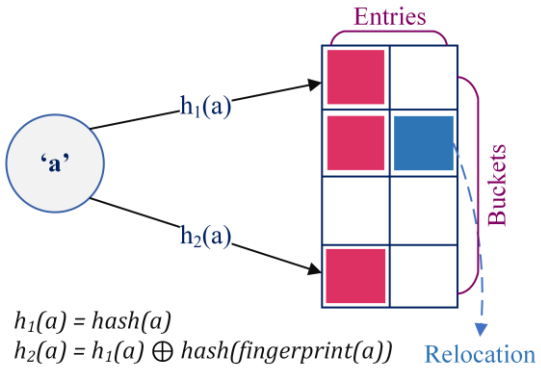

Fig. 4. Working of Cuckoo Filter

Cuckoo filters play a significant role in IP lookups, feature selection, visual tracking etc. [41][42]. Figure 4 depicts the working of the cuckoo filter.

IV. System Design

This section describes the proposed system model, and the assumptions made during the implementation phase as well as the notations used in the authentication algorithm.

A. System Model

The system model of the smart home is exhibited in Figure 5. The system model comprises of three entities: end nodes, edge node, and registration authority.

End Nodes (Publisher/ Subscriber): These are an amalgamation of resource-rich and resource-constrained devices. End nodes are meant to capture data from the environment and provide real-time updates to the users. In this work, the end nodes act as both publishers and subscribers. Publishers publish the data and subscriber consumes the data via broker.

Edge Nodes (Broker): These devices are rich in resources and act as a broker in this work. The broker is an intermediary entity that enables communication between publishers and subscribers. The role of the broker is to gather data from the publishers and forward it to the corresponding subscriber.

Registration Authority (R_A): This device is rich in resources and used during both device registration and data transmission phase.

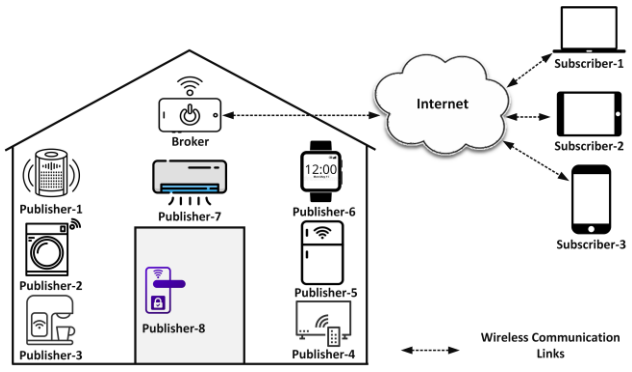

Fig. 5. System Model

B. Assumptions

The following assumptions were made during the implementation of the proposed device authentication algorithm.

1) The publishers are resource constraint in terms of processing, memory, and energy. Nevertheless, subscribers and brokers are not.
2) The broker is known to the publishers and subscribers which they want to register.
3) The devices used in the smart home networks are static.

C. System Requirements

Ensuring security in an IoT framework calls for the following requirements:

1) Security Requirements
   - Mutual Authentication: The participating entities i.e., publisher, subscriber and broker should authenticate each other to prevent unauthorized access and impersonation attacks.
   - Freshness of the Message: This ensures that the received message is fresh and acts as a counter to mitigate replay and denial of service attack.
   - Confidentiality: Meant to ensure privacy with respect to the information.
   - Authorization: The access rights of the sender and receivers are verified to mitigate instances of unauthorized access.
2) Functional Requirements
   - The security mechanism should support multi-factor authentication.
   - The security mechanism must provide secure communication between the publisher and subscriber through the broker.
3) Performance Requirements
   - The key performance requirements of the IoT devices are computation and communication cost.
   - The security mechanism so deployed should incur less communication and computation cost.
   - Delay in the authentication process must be as low as possible.

D. Notations Used

In this work, various notations have been used to achieve security in MQTT based IoT networks. The notations used in the device authentication algorithm are listed in Table II.

TABLE II. Notations Used

Notations	Description
$S_{ID} = \{S_1, S_2 \ldots\ldots S_n\}$	Identity of the Subscribers
$P_{ID} = \{P_1, P_2 \ldots\ldots P_n\}$	Identity of the Publishers
$E_{ID} = \{E_1, E_2 \ldots\ldots E_n\}$	Identity of the edge node

AP_{ID}	Alias identity of the publisher
AS_{ID}	Alias identity of the subscriber
R_A	Registration Authority
$R_i = \{R_1, R_2 .... R_n\}$	Random number
OT_{Key}	One-Time Secret Key
$M = \{M_1, M_2 ..... M_n\}$	Messages exchanged between the end nodes and edge nodes
$\|\|$	Concatenation Operations
$\oplus$	XOR Operation
$h(.)$	One-way hash function
$H_{MAC}(.)$	Hashed Message Authentication Code
R_{REQ}	Registration Request
R_{RESP}	Registration Response
$TS = \{TS_1, TS_2 ... T_N\}$	Time Stamp
T_{CS}	Current Time Stamp
ΔT	Maximum allowable time
T_i	i^{th} Token used in authentication
T_C	No. of times Concatenation Operation used
T_h	No. of times One-way hash function used
T_{XOR}	No. of times XOR function used
T_{HMAC}	No. of times Hashed Message Authentication codes used

V. Proposed System

This section describes the MQTT based device authentication mechanism. The proposed mechanism is composed of registration and authentication phases. In this, the broker needs to authenticate the publisher and subscriber needs to authenticate the broker to receive the updates from the end nodes. For this, both publishers and subscribers should register with the broker.

A. Summary of the Proposed Work

R_A creates an OT_{Key} at the time of registration and shares it with the publisher/ subscriber, and broker. The publisher/ subscriber uses this key and sends the registration request. The broker checks the correctness of the received OT_{Key} with the OT_{Key} it has. If both OT_{Key} matches, the publisher/ subscriber is registered with the broker; these registration details are stored in the Cuckoo filter for further use.

At the time of data transfer, the publisher, subscriber, and broker receive T_i and this is used as a unique session key for each transaction. The publisher transmits the data to the broker using the received T_i. The broker checks the T_i received from the publisher against its own T_i received from the R_A. If both T_i matches, the message and the T_i are forwarded to the corresponding subscribers. After receiving the message from the broker, the subscriber checks the correctness of the T_i against his own T_i received from R_A. If both T_i are correct, the subscribers consume the data, otherwise the data is discarded. Figure 6 summarizes the working of proposed device authentication mechanism in MQTT protocol.

B. One-Time Key (OT_{Key}) Generation

In this phase, the R_A generates the OT_{Key} and Algorithm 1 depicts the pseudo code. This phase comprises of two scenarios.

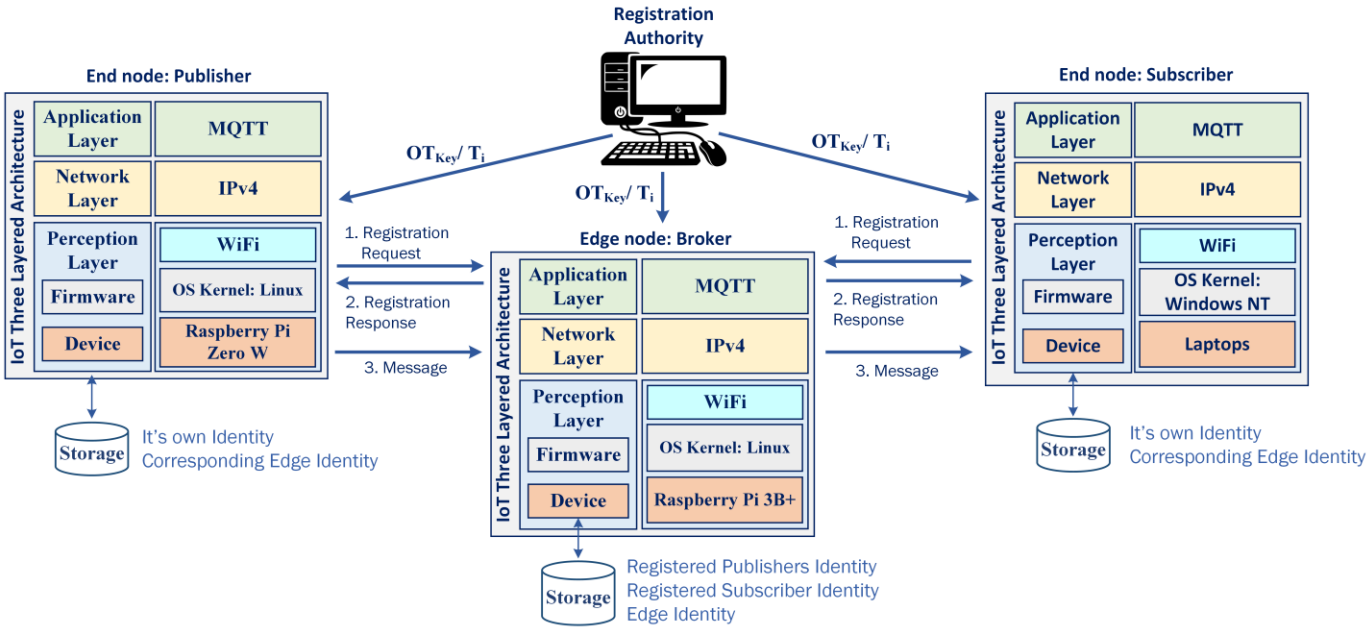

Fig. 6. Working of Enabling Lightweight Device Authentication in MQTT Protocol

Algorithm 1: ***OT_{Key} Generation***

INPUT: Square Matrix

OUTPUT: 8-bytes OT_{Key}

1. **PARTICIPANTS:** R_A, P_{ID}, S_{ID}, *and* E_{ID}
2. **Generate** Random Square Matrix
3. Repeat i from 0 to n
4. Repeat j from 0 to n
5. if (i == j) then
6. PrincipalDiagonal ← RandomMatrix[i][j]
7. endif
8. if (i+j) == (n-1) then
9. SecondaryDiagonal ← RandomMatrix[i][j]
10. endif
11. end loop
12. end loop
13. OT_{Key}= Concat (PrincipalDiagonal, SecondaryDiagonal)

Scenario 1: In this case OT_{Key} is generated by the R_A and shared with the publisher and broker; this helps in registering the publishers with the broker.

Scenario 2: In this case OT_{Key} is generated by the R_A and shared with the subscriber and broker; this helps in registering the subscriber with the broker. This helps subscribers to receive updates from the publisher via broker.

C. Registration Phase

The end nodes can communicate with the broker after the completion of registration process. Figure 7 and Figure 8 depict the overall steps in the registration process.

STEP-1: Performed by end-nodes

Always, end nodes initiate the registration process by generating the random number R_1 and OT_{Key}. An OT_{Key} is generated using the OT_{Key} algorithm; the sequence of steps is given in Algorithm 1. First, end node computes the unique identity for the publisher (i.e., P_{Id}=$D_{Serial}\oplus R_1$). Next, unique alias identity of the end device (i.e., $AP_{ID} = P_{ID}\oplus E_{ID}$) and edge device (i.e., $AE_{ID} = E_{ID}\oplus R_1$) are computed by the end node. Also, the end device computes the alias key (i.e., $A_{Key} = OT_{Key}\oplus P_{ID}$) and secret key (i.e., $S_{Key} = H_{MAC}(R_1, E_{ID})$). Also, end devices compute H_1 (i.e., $H_1 = h(E_{ID}||P_{ID}||S_{Key})$). Finally, the end node creates the R_{REQ} and shares it with the intended E_{ID}. The R_{REQ} message comprise of AP_{ID}, AE_{ID}, S_{Key}, A_{Key}, H_1, TS_1.

STEP 2 - Performed by the edge node

After receiving the R_{REQ} message from the end nodes, the E_{ID} checks the freshness of the message by calculating the ΔT (i.e., $\Delta T = T_{CS}-TS_1$?). Later, publisher identity is regenerated (i.e., $P_{ID}^{|} = AP_{ID}\oplus E_{ID}$). Next, random number $R_1^{|}$ will be regenerated (i.e., $R_1^{|}$= $E_{ID}\oplus AE_{ID}$). Also, the edge node computes the $OT_{Key}^{|}$and $S_{Key}^{|}$ (i.e., $OT_{Key}^{|} = A_{Key}\oplus P_{ID}$ and $S_{Key}^{|}$= $H_{MAC}(R_1^{|}, E_{ID})$). Finally, $H_1^{|}$ is calculated (i.e., $H_1^{|} = h(E_{ID}||P_{ID}^{|}||S_{Key}^{|})$). After these calculations, the edge node performs the originality check of the parameters OT_{Key}, H_1, S_{Key}, and E_{ID} by comparing with the regenerated parameters $OT_{Key}^{|}$, $H_1^{|}$, $S_{Key}^{|}$, and ($P_{ID}^{|} = AP_{ID}\oplus E_{ID}$) respectively. On successful completion of all security checks, the broker sends the R_{RESP} to the end device. R_{RESP} is a tuple and it comprise of AE_{ID} (i.e., $E_{ID}\oplus R_2$), S_{Key} (i.e., HMAC (R_2, E_{ID})), and H_2 (i.e., $H_2 = E_{ID}||P_{ID}||S_{Key}$).

STEP 3- Performed by the end nodes

Upon receiving the R_{RESP} message from the edge node, the freshness of the message is checked first by calculating the ΔT (i.e., $\Delta T = T_{CS}-TS_2$?). Subsequently, the end node recomputes the $R_2^{|}$ and $S_{Key}^{|}$ (i.e., $R_2^{|} = AE_{ID}\oplus E_{ID}$ and $S_{Key}^{|} = H_{MAC}(R_2^{|}, E_{ID})$) respectively. Furthermore, edge node regenerates $H_2^{|}$ (i.e., $H_2^{|}$=h($E_{ID}||P_{ID}||S_{Key}^{|}$)). After these computations, end node checks for the valid edge node (i.e., $E_{ID} == E_{ID}^{|}$). Also end node checks for the integrity of the received message by comparing H_2 and $H_2^{|}$ (i.e., $H_2 == H_2^{|}$). The registration phase is successful only when the listed procedure is executed with no errors.

In the case of subscriber registration S_{ID} will be used instead of P_{ID}. Apart from that other parameters will be the same. Figure 7 and Figure 8 depict sequence of steps executed during the publisher and subscriber registration respectively.

D. Authentication Phase

Both device authentication and data transmission happen during this phase. In this mechanism, all parties initiate the device authentication process soon after receiving the messages from the communication entities. This section presents the authentication process between the publisher and broker, and broker and subscribers.

STEP-1: Performed by R_A

1. R_A generates the 8-bits Token (i.e.,T_i) for each session and shares with the registered publisher, subscriber and broker using secure channel; this token will be used for further communication among the publisher, subscriber, and the broker.
2. Up on receiving the T_i from R_A, the P_{ID}, S_{ID} and E_{ID} checks for ΔT (i.e., $\Delta T = T_{CS}-T_1$?); if the ΔT is less than the maximum allowable transmission time P_{ID}, S_{ID} and E_{ID} stores the T_i and it is used by all the devices at the time of authentication and data transmission.

STEP-2: Performed by Publisher (P_{ID})

In this step, the publisher performs the following steps to get connected with the broker.

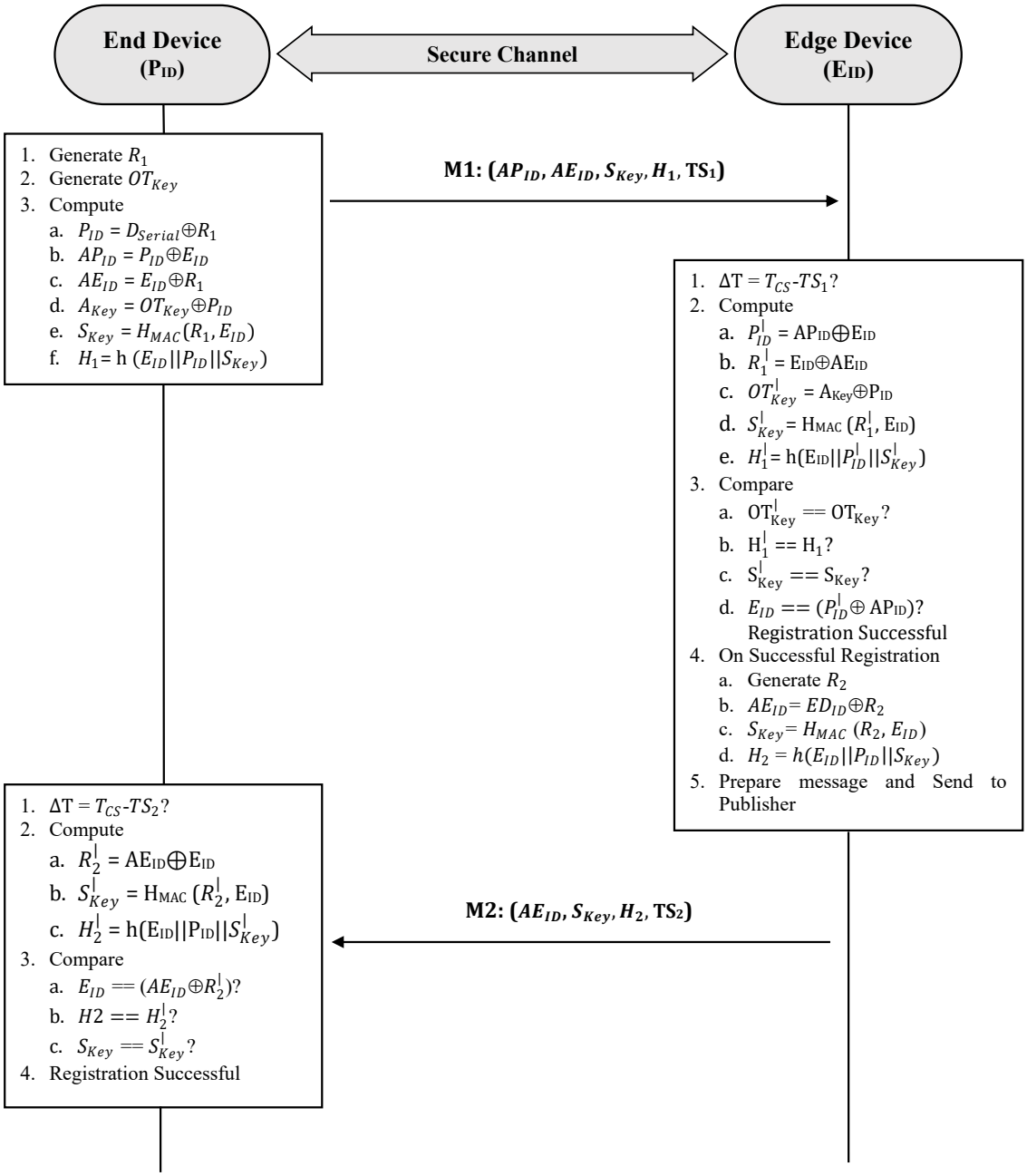

Fig. 7. Publisher Registration

1. P_{ID} computes the AP_{ID} by performing the XOR operation between the P_{ID} and E_{ID} (i.e., $AP_{ID} = P_{ID} \oplus E_{ID}$). Also, P_{ID} generates the M_d by performing the hash operation on the concatenated message (i.e., $M_d = h\ (P_{ID} \| E_{ID} \| T_i)$); this message is used at the receiver side to check the integrity of the message.
2. P_{ID} creates the Masked Data (MD) by using Sensed Data (SD), T_i and R_i (i.e., $MD = SD_i \oplus T_i \oplus R_i$)
3. Finally, the P_{ID} prepares the message and forwards it to the broker over a public channel. The message comprises of AP_{ID}, MD, M_d, R_i, and T_{CS} (i.e., $M = (AP_{ID}, MD, M_d, R_i, T_{CS})$) and it is sent to the broker.

STEP-3: Performed by Broker (E_{ID})

In this step, the broker gets the data from the publisher and forwards it to its corresponding subscriber by performing the series of steps.

1. Upon receiving the message from the publisher, the broker verifies the ΔT; if the MTU is less than ΔT, the broker performs the following Operations.
2. The broker recomputes the $P_{ID}^{|}$ by using AP_{ID}, and E_{ID} (i.e., $P_{ID}^{|} = AP_{ID} \oplus E_{ID}$).
3. Then, the $SD^{|}$ is obtained using MD, T_i and R_i (i.e., $SD^{|} = MD \oplus T_i \oplus R_i$)
4. Finally, the broker regenerates the $MD^{|}$ using $P_{ID}^{|}$, and its own identity E_{ID} and received T_i (i.e., $MD^{|} = SD_i^{|} \oplus T_i \oplus R_i$); this id used for the integrity check.

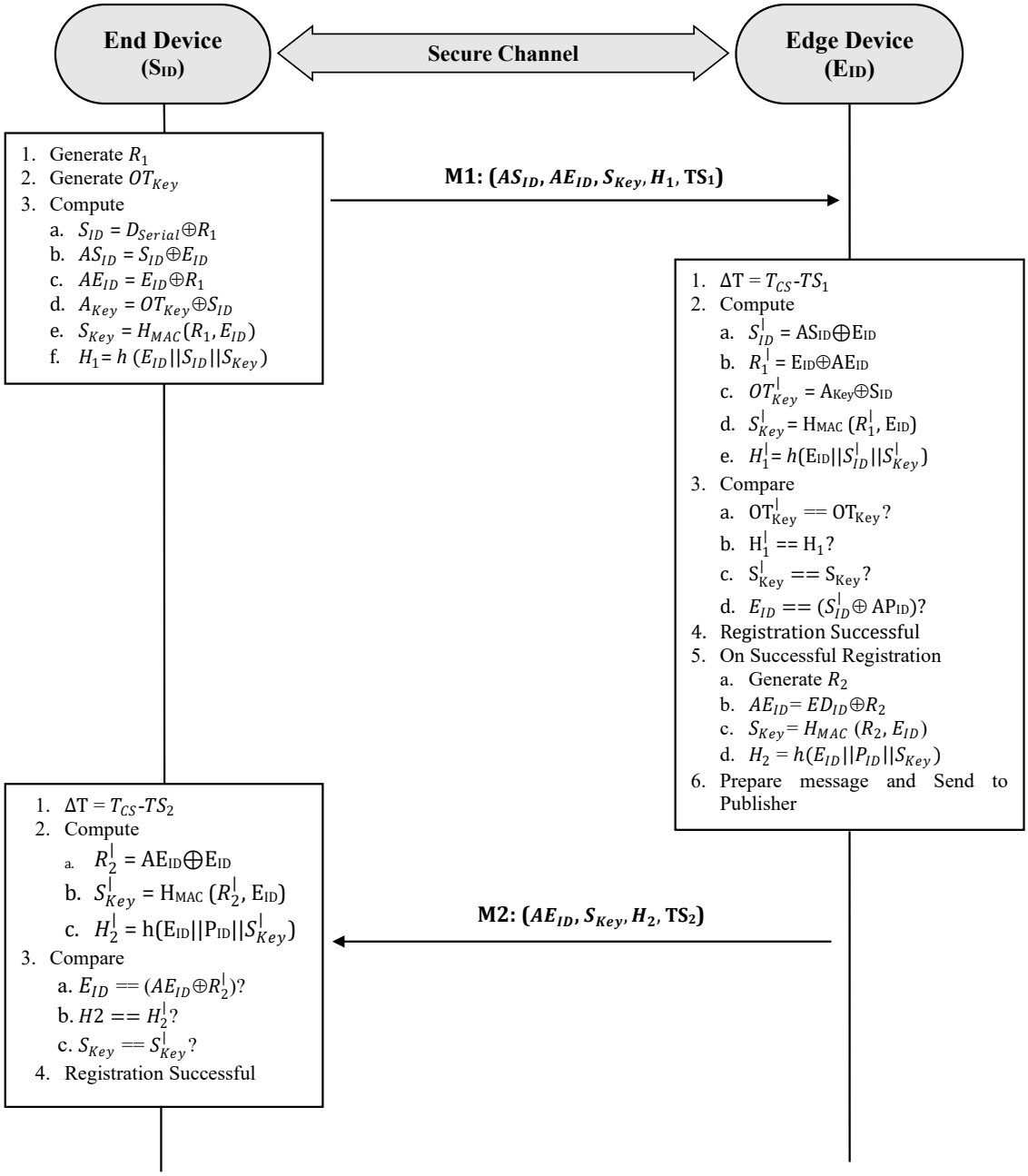

Fig. 8. Subscriber Registration

5. The broker finds the corresponding subscribers of the P_{ID} from the subscriber list, and computes the AE_{ID} by using S_{ID} and E_{ID} (i.e., $AE_{ID} = S_{ID} \oplus E_{ID}$)
6. The broker computes the M_d by performing the hash operation on the concatenated message (i.e., $M_d = H(S_{ID}||E_{ID}||T_i)$; this message is used at the receiver side to check the integrity of the message
7. Then, the broker checks for the integrity of the message using the generated $P_{ID}^{|}$, its own identity E_{ID} and received T_i (i.e., $MD = P_{ID}^{|} \oplus E_{ID} \oplus T_i$)
8. Finally, the broker prepares the message and forwards it to the corresponding subscribers over a public channel. The message comprises of AE_{ID}, MD, M_d, R_i and TS_1(i.e., $M_1 = (AE_{ID}, MD, M_d, R_i$ and $TS_1)$

STEP-3: Performed by the Subscriber (S_{ID})

1. Upon receiving the message from the broker, the S_{ID} verifies the ΔT; if the MTU is greater than ΔT, the message will be discarded by the S_{ID}.
2. The S_{ID} recomputes the $E_{ID}^{|}$ by using E_{ID} and its own identity (i.e., $E_{ID}^{|} = AE_{ID} \oplus S_{ID}$), and $E_{ID}^{|}$ is compared with the E_{ID}; this authenticates the existence of the broker
3. Then, S_{ID} checks for the integrity of the message using the generated E_{ID}, R_i, T_i, (i.e., $M_d = H(E_{ID}, S_{ID}, T_i)$)
4. Finally, the SD is obtained using MD, T_i and R_i(i.e., $SD_i = MD_i \oplus T_i \oplus R_i$) and this data is used to take further actions

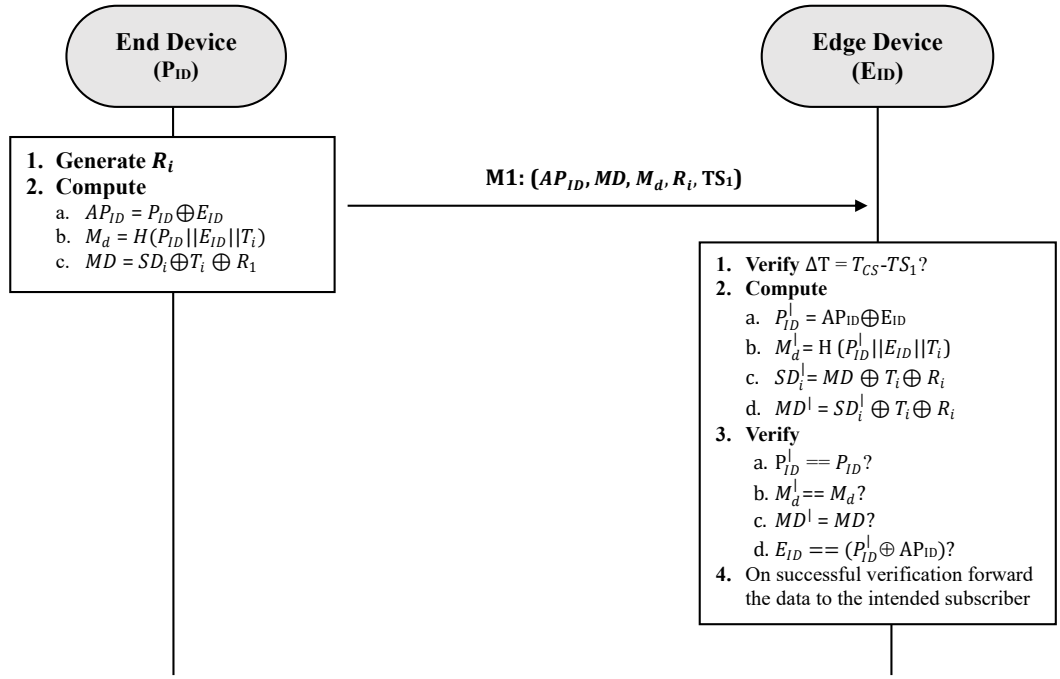

Fig. 9. Publisher and Broker Authentication

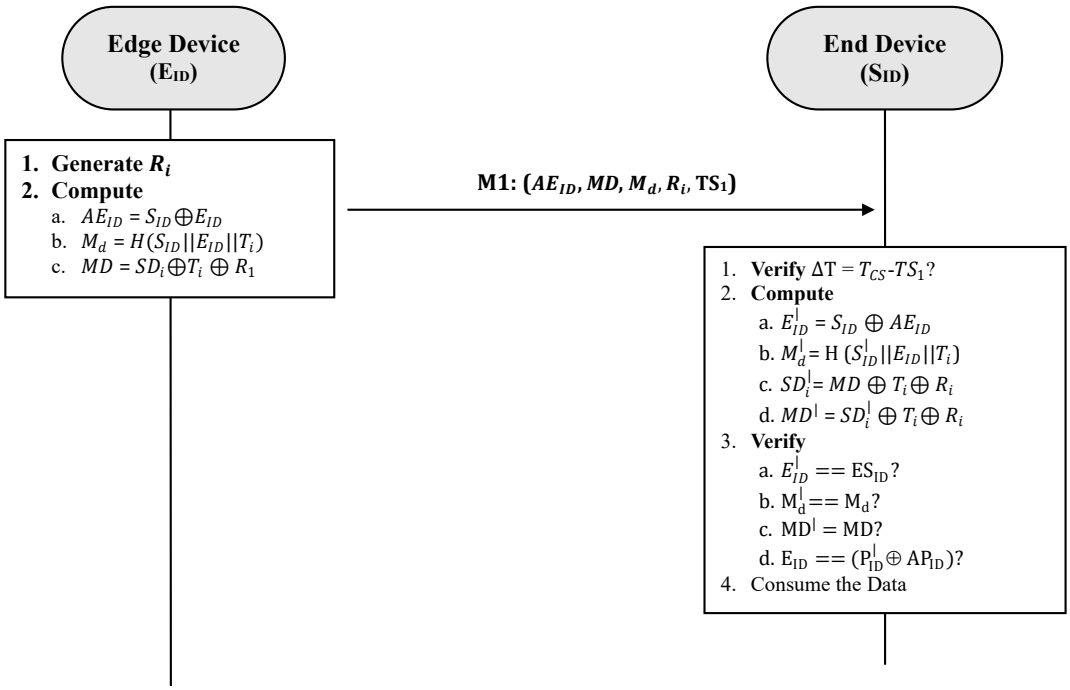

Fig. 10. Broker and Subscriber Authentication

Figures 9 and Figure 10 show the detailed steps used in the publisher-broker and broker-subscriber authentication.

VI. Security Analysis

This section presents the detailed security analysis (Formal and Formal Security Analysis) of the proposed security mechanism.

A. Informal Security Analysis

This section will focus on demonstrating that the proposed method can ensure crucial security needs. In this approach, we often include the testing and observe the system's behavior.

1) Replay Attack

In this mechanism, all the interactions are subject to rigorous time synchronization. After receiving the message, the maximum transmission delay ΔT is always determined. For example, during the registration or authentication step, an attacker establishes a new session to send a message M. If there is any delay in getting the messages, the connection will be terminated. Token T_i is used by the P_{ID} and S_{ID}, and E_{ID} and it is dynamic in nature. For intruder it is very difficult to extract T_i. The scenario of the reply attack is depicted in Figure 11.

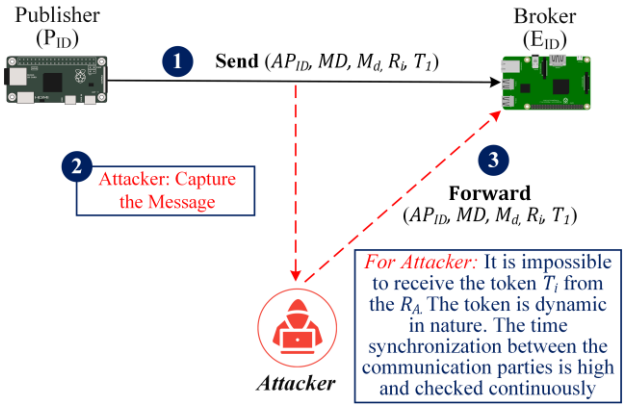

Fig.11. Replay Attack

2) Malicious Node Detection

If an attacker utilizes incorrect device P_{ID} and S_{ID}, the broker (E_{ID}) will quickly detect the rogue node from the list during the authentication operation. Figure 12 depicts the malicious bode attack scenario.

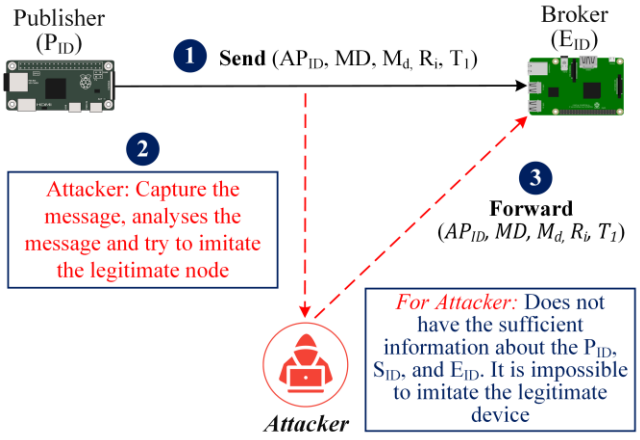

Fig.12. Malicious Node Attack

3) Device Anonymity

Device anonymity is made possible by the proposed authentication mechanism; IoT devices in the network use anonymous identities to conceal their original identities. AP_{ID}, AS_{ID}, and AE_{ID} clearly conceals the identity of publisher, subscriber, and brokers respectively.

4) Device Impersonate Attack

If an adversary $Device_A$ attempts to mimic a genuine device in the smart home network first it needs to register with the broker using the OT_{Key} generated by the R_A. After the successful registration the broker maintains a separate publisher and subscriber list. Since the adversary does not have its presence in either the list, the broker can easily identify the adversary trying to impersonate. Device-impersonate attack scenario of the proposed mechanism is depicted in Figure 13.

5) Device Authentication

Device authentication takes place after all communication parties' publisher and broker, and subscriber and broker authenticate each other. During this process, the information is exchanged between publisher and subscriber through broker; these communication parties verify each other prior to information transmission.

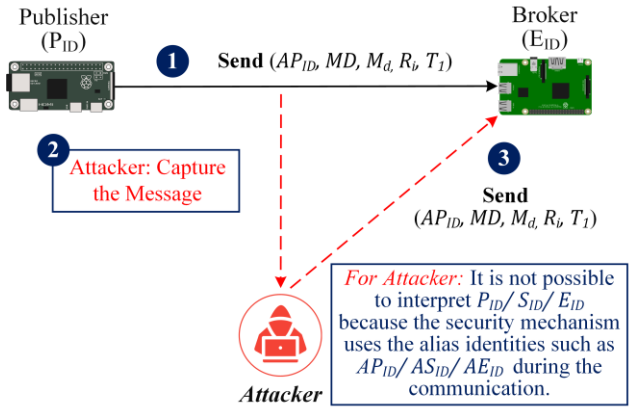

Fig.13. Impersonate Attack

B. Formal Security Analysis

In this work, Automated Validation of Internet Security Protocols and Applications (AVISPA) protocol analyzer is utilized to validate the proposed authentication mechanism. Here, we describe the publisher, subscriber, and broker as roles. Also, we define the environment, sessions, and goals to validate the strength of the proposed protocol. After defining the roles, environment, sessions, and goals, we executed the security protocol analyzer using the On-the-Fly Model-Checker (OFMC) module.

Claim 1: The proposed lightweight mutual device authentication protocol requires minimum resources for the computation.

Proof: The proposed authentication algorithm that relies on lightweight cryptographic functions like hashing, XOR and Concatenation Operations. Compared to other cryptographic functions, the above-mentioned cryptographic functions perform better in resource constrained devices. Also, the algorithm completes the mutual authentication in a shorter duration, this is due to the use of light-weight cryptography.

Claim 2: The proposed protocol enhances the security feature of the MQTT protocol.

Proof: MQTT protocol does not provide the device authentication by default. However, the proposed work facilitates the devices authentication in publishers, brokers, and subscribers; these devices can share the information after the successful completion of the registration and authentication process.

Figures 14 and 15 shows the AVISPA results generated during the publisher-broker and broker-subscriber authentication.

```
% Publisher Authentication
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/publisher_authentication.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.00s
  visitedNodes: 20 nodes
  depth: 4 plies
```

Fig. 14. Publisher-Broker Authentication

```
% Subscriber Authentication
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/subscriber_authentication.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.00s
  visitedNodes: 20 nodes
  depth: 4 plies
```

Fig. 15. Broker-Subscriber Authentication

VII. Results and Discussion

In this work, small scale experimental smart home network is created, and it comprises of end nodes, and edge nodes. End nodes act as a publisher/ subscriber and the edge node is treated as the brokers. It's assumed that the publishers are resource constrained devices. The broker and the subscriber are rich in resources. In this work, RaspberryPi Zero W devices were used as the publishers, devices like laptops and mobile phones were used as the subscribers, and RaspberryPi 3 B+ devices were used as the brokers. The laptop with 16GB RAM and 1TB SSD is used as the R_A and which initiates the device registration process by running the Python Graphical User Interface (GUI). Also, R_A generates the OT_{Key} and T_i, which are used during the device registration, and authentication process respectively. These devices are deployed in the 30*30 feet area.

A. Storage Cost

IoT devices often have limited memory. However, the proposed security mechanism stores some vital information required for device authentication. Memory usage is one of the significant components of performance study.

Publisher (P_{ID}) / Subscriber (S_{ID}): The publisher/subscriber stores its own identity P_{ID}/S_{ID}, E_{ID}, and OT_{Key}, which are of 64-bits each. The publisher/ subscriber uses the OT_{Key} during the registration phase. After successful completion of the registration process, the publisher/subscriber flush out the OT_{Key}. from the device memory. During the authentication, publisher/subscriber and broker use the T_i (64-bits) as a session key. The publisher/subscriber, broker identity, and OT_{Key} /T_i specified sizes are 64 bits each. Then the storage cost at publisher/ subscriber are 64+64+64 = 192 bits.

Broker (E_{ID}): It stores, P_{ID}, S_{ID} and subscription details of the subscriber. The broker guides the messages from the publisher to its corresponding subscribers; for this reason, the broker maintains the publisher and subscriber list; this list comprises of publisher, subscriber identities along with the topics registered for. The size of the topic is 32-bits; each entry in the publisher/subscriber table is 64+64+32 = 160 bits. The size of the publisher/subscriber table is dynamic in nature; the size of the table increases as the number of registered devices increases.

B. Communication Cost

In this work, totally six messages were exchanged between the P_{ID}, S_{ID} and E_{ID}. During registration phase, P_{ID} and S_{ID} sends the registration request messages to E_{ID}. Four messages (i.e., two registration request messages from publisher and subscriber, and two registration response messages from broker to publisher and subscriber) were exchanged between them. On successful registration, the E_{ID} sends the registration response message to the P_{ID} and S_{ID} respectively. At the time of authentication procedure, P_{ID} shares its identity, T_i and data with the E_{ID}, and it forwards its identity, data received from the P_{ID} with the corresponding S_{ID}; Totally, two messages were shared between them.

C. Computation Cost

IoT devices are operated with constrained resources, including limited processing power, memory, and energy. Analyzing and optimizing the computational cost is essential.

1) OT_{Key} Generation

The R_A generates the OT_{Key}, which is then shared with the P_{ID}, S_{ID} and E_{ID}. During the device registration phase, P_{ID} and S_{ID} utilize this to register with E_{ID}. The R_A takes 0.328 milliseconds to generate 8-byte OT_{Key}.

2) Insertion and Searching

E_{ID} implements the Cuckoo filter to store and track of registered publishers and subscribers. Insertion and searching are faster in Cuckoo filters compared to the other data structures. In this work, Cuckoo filter stores the 8-byte publisher, subscriber identities, and subscriber to publisher mapping. The amount of time taken by the broker to insert and search for a particular publisher and subscriber is 0.015 milliseconds.

3) Device Registration and Authentication

Both P_{ID} and S_{ID} should register with the E_{ID} before sending the data. Figure 16 and Figure 17 depicts the amount of time taken by P_{ID} and S_{ID} to register with the broker at different epochs. The publishers have taken more time to register; this is due to the publishers having less computation resources compared to subscribers. The average registration time of the publisher and subscribers are 0.15 seconds and 0.043 seconds, respectively. Figures 18 and 19 depict the energy consumed by the publisher and subscriber respectively during registration phase. Proposed security mechanism comprised of

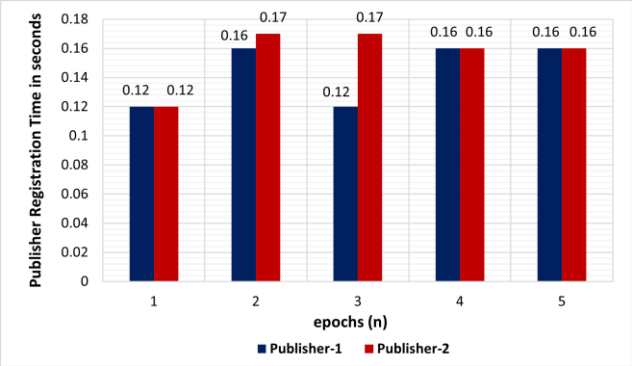

Fig. 16. Publisher Registration Time (in Seconds)

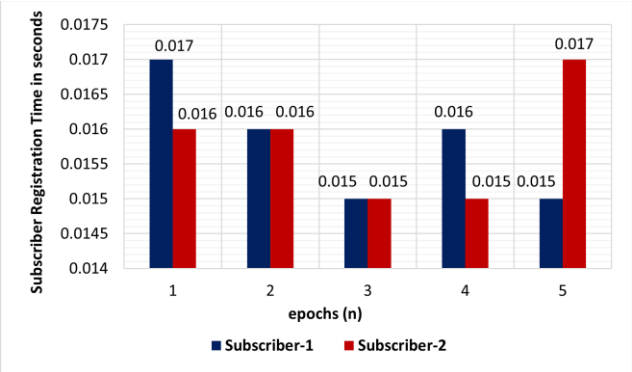

Fig. 17. Subscriber Registration Time (in Seconds)

1. Publisher-broker authentication: In this, broker authenticates the publisher and 2. Broker-subscriber.

authentication: In this, the subscriber authenticates the broker, and it takes less time; this is due to the rich resources possessed by the subscriber. The average publisher-broker authentication time is 0.037 seconds, and the average broker-subscriber authentication is 0.0054 seconds. Figure 20 shows the publisher-broker authentication time at different time intervals. Here, both publishers (i.e., Publisher-1 and Publisher-2) have taken the same amount time in most of the time intervals. Figure 21 depicts the broker-subscriber authentication at different epochs and both the subscribers (i.e., Subscriber-1 and Subscriber-2) have taken the same amount of time for authentication. Figures 22 and 23 depict the energy consumption of the publishers and subscribers during the authentication phase.

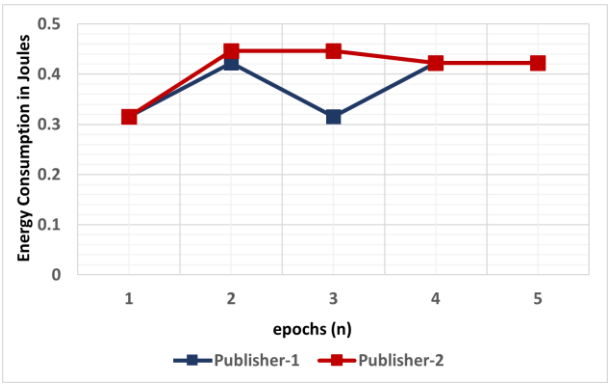

Fig.18. Energy Consumption of Publishers during Registration (in Joules)

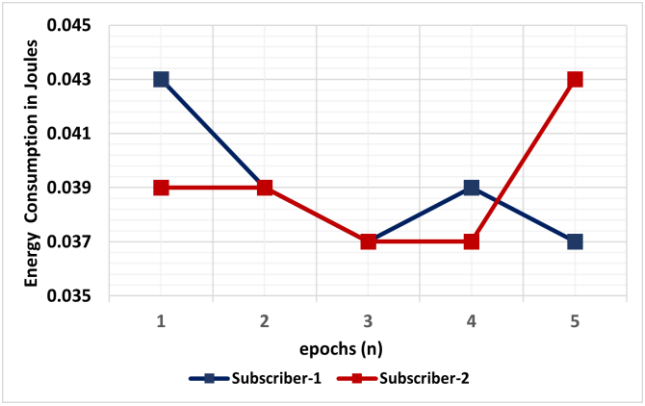

Fig. 19. Energy Consumption of Subscribers during Registration (in Joules)

TABLE III. VARIOUS CRYPTOGRAPHIC OPERATIONS USED IN THE PROPOSED SECURITY MECHANISM

	$P_{ID} \rightarrow E_{ID}$		$E_{ID} \rightarrow S_{ID}$	
Phase	**Publisher (P_{ID})**	**Broker (E_{ID})**	**Subscriber (S_{ID})**	**Broker (E_{ID})**
Registration	$4*T_C+2*T_h+5*T_{XOR}+2*T_{HMAC}+2*R_i$	$4*T_C+2*T_h+5*T_{XOR}+2*T_{HMAC}+2*R_i$	$4*T_C+2*T_h+5*T_{XOR}+2*T_{HMAC}+2*R_i$	$4*T_C+2*T_h+5*T_{XOR}+2*T_{HMAC}+2*R_i$
Authentication	$2*T_C+1*T_h+3*T_{XOR}$	$2*T_C+1*T_h+5*T_{XOR}$	$2*T_C+1*T_h+3*T_{XOR}$	$2*T_C+1*T_h+5*T_{XOR}$

The proposed secure MQTT protocol has 4.66% time overhead as compared to the MQTT protocol. This negligible overhead is due to the implementation of the security mechanism. This is calculated with the help of equation 3. Table III highlights the different cryptographic operations used in the proposed security mechanism. Table IV emphasizes comparison between the proposed security mechanism and the other related security mechanisms. The registration phase in the proposed mechanism has a higher computation overhead compared to the other security mechanism. At the time of authentication phase, the proposed security mechanism reduces the number of computations. Also, our algorithm eliminated the use of traditional cryptographic operations.

$$DataTrans_{Time} = \sum_{i=1}^{n} MQTT_{Original}$$

$$Auth\&DataTrans_{Time} = \sum_{i=1}^{n} MQTT_{Proposed}$$

$$X = Max(DataTrans_{Time}, Auth\&DataTrans_{Time})$$

$$Y = Min(DataTrans_{Time}, Auth\&DataTrans_{Time})$$

$$Time_{Overhead} = \left(\frac{X-Y}{n}\right) * 10) \quad (3)$$

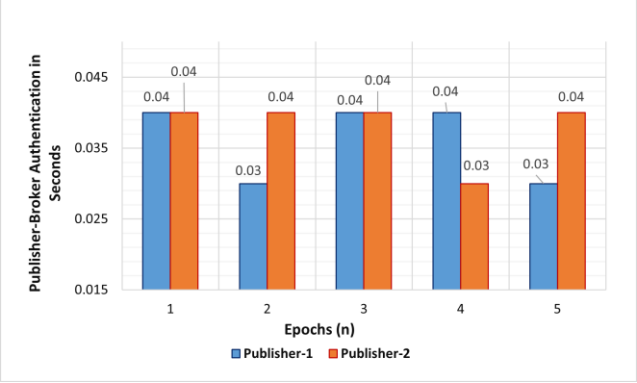

Fig. 20. Publisher-broker Authentication

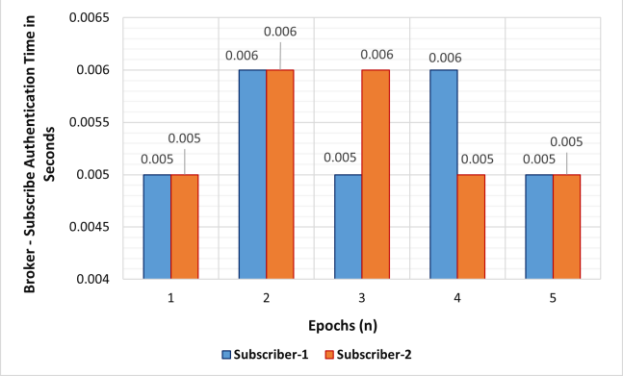

Fig. 21. Broker-Subscriber Authentication Time (in Seconds)

VIII. Conclusion

Motivated by the current security issues in IoT, we proposed a novel mutual lightweight device authentication mechanism.

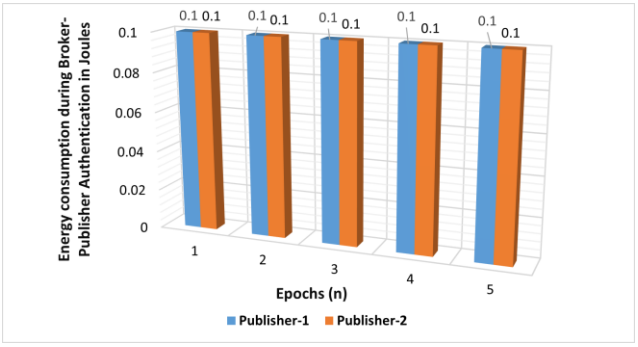

Fig. 22. Energy Consumption of the Publishers during Authentication

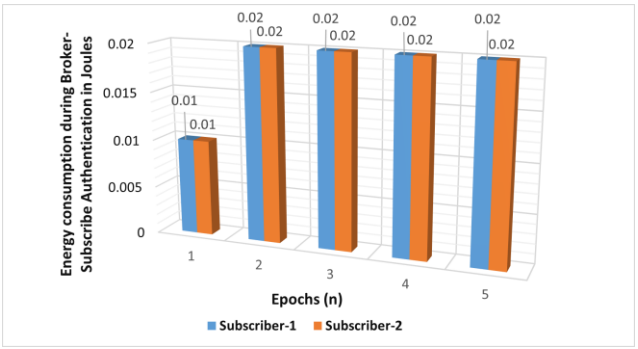

Fig. 23. Energy Consumption of the Subscribers during Authentication

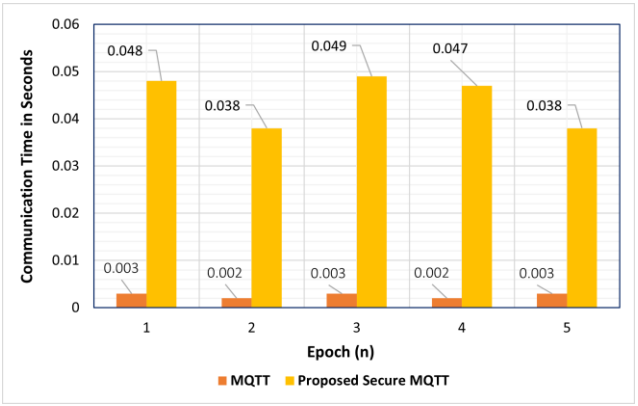

Fig. 24. MQTT VS. Proposed (in Seconds)

TABLE IV. Comparative Analysis of Proposed Security Mechanisms with Other Mechanisms

Security Mechanisms	Registration	Authentication
[43]	$1T_C+2T_h+1T_{XOR}$	$10T_C+14T_h+10T_{XOR}+2R_i$
[44]	$7T_h+4\ T_C$	$17T_C+6T_h+5T_E+5T_C$
Proposed	$16T_C+8T_h+20T_{XOR}+8T_{HMAC}+8R_i$	$8T_C+4T_h+16T_{XOR}$

$T_E \rightarrow$ *Symmetric Key Encryption Technique*
$T_D \rightarrow$ *Symmetric* Key Decryption technique

This mechanism is applied in the real-time smart home network, where communication happens using publisher-subscriber pattern. The security mechanism was implemented practically and evaluated in a small-scale, yet a realistic smart

home testbed. A series of experiments were performed to examine the energy-efficiency and security strength of the proposed mechanism. Also, we demonstrated that the proposed protocol is resistant against various attacks. Furthermore, the proposed lightweight security mechanism exhibits improved computation, communication and storage cost compared to other related lightweight mechanisms. Our security mechanism is designed for the MQTT protocol and best suited for the IoT-enabled smart homes.

REFERENCES

[1] E. N. Neshenko, Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations," IEEE Communication Surveys & Tutorials, vol. 21, no. 3, pp. 2702–2733, 2019, doi: 10.1109/COMST.2019.2910750.

[2] Syed Wasif Abbas Hamdani, Abdul Waheed Khan, Naima Iltaf, and Waseem Iqbal, "DTMSim-IoT: A Distributed Trust Management Simulator for IoT Networks," in IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calgary, Canada: IEEE, 2020, pp. 491–498. doi: 0.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00091.

[3] Chintan Patel and Nishant Doshi, "A Novel MQTT Security Framework in generic IoT Model," in 3rd International Conference on Computing and Network Communication, Procedia Computer Science, Elsevier, 2020, pp. 1399–1408.

[4] Muhammad Nauman Khan, Asha Rao, and Seyit Camtepe, "Lightweight Cryptographic Protocols for IoT-Constrained Devices: A Survey," IEEE Internet Things J, vol. 8, no. 6, pp. 4132–4156, 2021, doi: 10.1109/JIOT.2020.3026493.

[5] J. Roldán-Gómez, J. Carrillo-Mondéjar, J. M. Castelo Gómez, and S. Ruiz-Villafranca, "MQTT Security," Applied Sciences, vol. 12, no. 21, p. 10991, 2022, doi: https://doi.org/10.3390/app122110991.

[6] Ziarmal Nazar Mohammad, Fadi Farha, Adnan O. M. Abuassba, Shunkun Yang, and Fang Zhou, "Access control and authorization in smart homes: A survey," Tsinghua Science and Technology, vol. 26, no. 6, pp. 906–917, 2021, doi: 10.26599/TST.2021.9010001.

[7] A. Amjad, F. Azam, M. W. Anwar, and W. H. Butt, "A Systematic Review on the Data Interoperability of Application Layer Protocols in Industrial IoT," IEEE Access, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 96528–96545, 2021. doi: 10.1109/ACCESS.2021.3094763.

[8] E. Kovacs, "70 Percent of IoT Devices Vulnerable to Cyberattacks: HP," Security Week, Jul. 29, 2014. https://www.securityweek.com/70-iot-devices-vulnerable-cyberattacks-hp/ Accessed on 29 November 2023.

[9] R. Heartfield et al., "A taxonomy of cyber-physical threats and impact in the smart home," Computers and Security, vol. 78. Elsevier Ltd, pp. 398–428, Sep. 01, 2018. doi: 10.1016/j.cose.2018.07.011.

[10] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communications Surveys and Tutorials, vol. 17, no. 4, pp. 2347–2376, Oct. 2015, doi: 10.1109/COMST.2015.2444095.

[11] A. J. Hintaw, S. Manickam, M. F. Aboalmaaly, and S. Karuppayah, "MQTT Vulnerabilities, Attack Vectors and Solutions in the Internet of Things (IoT)," IETE Journal of Research. Taylor and Francis Ltd., 2021. doi: 10.1080/03772063.2021.1912651.

[12] E. B. Sanjuan, I. A. Cardiel, J. A. Cerrada, and C. Cerrada, "Message Queuing Telemetry Transport (MQTT) Security: A Cryptographic Smart Card Approach," IEEE Access, vol. 8, pp. 115051–115062, 2020, doi: 10.1109/ACCESS.2020.3003998.

[13] W.-T. Su, W.-C. Chen, and C.-C. Chen, "An Extensible and Transparent Thing-to-Thing Security Enhancement for MQTT Protocol in IoT Environment," 2023. [Online]. Available: https://www.eclipse.org/paho/.

[14] SeongHan Shin, Kazukuni Kobara, Chia-Chuan Chuang, and Weicheng Huang, "A security framework for MQTT," in IEEE Conference on Communications and Network Security (CNS), Philadelphia, USA: IEEE, 2016, pp. 432–436. doi: 10.1109/CNS.2016.7860532.

[15] S. A. Soleymani et al., "A secure trust model based on fuzzy logic in vehicular Ad Hoc networks with fog computing," IEEE Access, vol. 5, pp. 15619–15629, Jul. 2017, doi: 10.1109/ACCESS.2017.2733225.

[16] S. Hernández Ramos, M. T. Villalba, and R. Lacuesta, "MQTT Security: A Novel Fuzzing Approach," Wirel Commun Mob Comput, vol. 2018, 2018, doi: 10.1155/2018/8261746.

[17] S. A. Abdymanapov, M. Muratbekov, S. Altynbek, and A. Barlybayev, "Fuzzy Expert System of Information Security Risk Assessment on the Example of Analysis Learning Management Systems," IEEE Access, vol. 9, pp. 156556–156565, 2021, doi: 10.1109/ACCESS.2021.3129488.

[18] S. V. Simpson and G. Nagarajan, "A fuzzy based Co-Operative Blackmailing Attack detection scheme for Edge Computing nodes in MANET-IOT environment," Future Generation Computer Systems, vol. 125, pp. 544–563, Dec. 2021, doi: 10.1016/j.future.2021.06.052.

[19] D. Choi, S. H. Seo, Y. S. Oh, and Y. Kang, "Two-Factor Fuzzy Commitment for Unmanned IoT Devices Security," IEEE Internet Things J, vol. 6, no. 1, pp. 335–348, Feb. 2019, doi: 10.1109/JIOT.2018.2837751.

[20] Ö. Yerlikaya and G. Dalkılıç, "Authentication and Authorization Mechanism on Message Queue Telemetry Transport Protocol," 2018 3rd International Conference on Computer Science and Engineering

(UBMK), Sarajevo, Bosnia, and Herzegovina, 2018, pp. 145-150, doi: 10.1109/UBMK.2018.8566599.

[21] [21] S. Tian and V. G. Vassilakis, "On the Efficiency of a Lightweight Authentication and Privacy Preservation Scheme for MQTT," Electronics, vol. 12, no. 14, p. 3085, Jul. 2023, doi: 10.3390/electronics12143085.

[22] A. Rahman, S. Roy, M. S. Kaiser, and M. S. Islam, "A Lightweight Multitier S-MQTT Framework to Secure Communication between low-end IoT Nodes," 2018 5th International Conference on Networking, Systems and Security (NSysS), Dhaka, Bangladesh, 2018, pp. 1-6, doi: 10.1109/NSysS.2018.8631379.

[23] S. Iyer, G. V. Bansod, P. N. V and S. Garg, "Implementation and Evaluation of Lightweight Ciphers in MQTT Environment," 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Msyuru, India, 2018, pp. 276-281, doi: 10.1109/ICEECCOT43722.2018.9001599.

[24] H. Kurdi and V. Thayananthan, "A Multi-Tier MQTT Architecture with Multiple Brokers Based on Fog Computing for Securing Industrial IoT," Applied Sciences, vol. 12, no. 14, p. 7173, Jul. 2022, doi: 10.3390/app12147173.

[25] M. Amoretti, R. Pecori, Y. Protskaya, L. Veltri and F. Zanichelli, "A Scalable and Secure Publish/Subscribe-Based Framework for Industrial IoT," in IEEE Transactions on Industrial Informatics, vol. 17, no. 6, pp. 3815-3825, June 2021, doi: 10.1109/TII.2020.3017227.

[26] A. J. Hintaw, S. Manickam, S. Karuppayah, M. A. Aladaileh, M. F. Aboalmaaly and S. U. A. Laghari, "A Robust Security Scheme Based on Enhanced Symmetric Algorithm for MQTT on the Internet of Things," in IEEE Access, vol. 11, pp. 43019-43040, 2023, doi: 10.1109/ACCESS.2023.3267718.

[27] A. Refaey, A. Sallam, A. Shami, "On IoT applications: a proposed SDP framework for MQTT," in Electronics Letters, Vol. 55, pp. 1201–1203, Oct. 2019, doi: 10.1049/el.2019.2334.

[28] F. Azzedin and T. Alhazmi, "Secure Data Distribution Architecture in IoT Using MQTT," Applied Sciences, vol. 13, no. 4, p. 2515, Feb. 2023, doi: 10.3390/app13042515.

[29] Hung-Yu Chien, Pei-Chih Lin, Mao-Lun Chiang, "Efficient MQTT Platform Facilitating Secure Group Communication," Journal of Internet Technology, Vol. 21, pp. 1921-1940, doi: 10.3966/160792642020122107007

[30] M. Hamad, A. Finkenzeller, H. Liu, J. Lauinger, V. Prevelakis and S. Steinhorst, "SEEMQTT: Secure End-to-End MQTT-Based Communication for Mobile IoT Systems Using Secret Sharing and Trust Delegation," in *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3384-3406, 15 Feb.15, 2023, doi: 10.1109/JIOT.2022.3221857.

[31] A. Kondoro, I. B. Dhaou and H. Tenhunen, "Enhancing the Security of IoT-enabled DC Microgrid using Secure-MQTT," 2020 6th IEEE International Energy Conference (ENERGYCon), Gammarth, Tunisia, 2020, pp. 29-33, doi: 10.1109/ENERGYCon48941.2020.9236448.

[32] T. Noguchi, M. Nakagawa, M. Yoshida, and A. G. Ramonet, "A Secure Secret Key-Sharing System for Resource-Constrained IoT Devices using MQTT," *2022 24th International Conference on Advanced Communication Technology (ICACT)*, PyeongChang Kwangwoon_Do, Korea, Republic of, 2022, pp. 147-153, doi: 10.23919/ICACT53585.2022.9728781.

[33] H. Lee, J. Lim, and T. Kwon, "MQTLS: Toward Secure MQTT Communication with an Untrusted Broker," 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2019, pp. 53-58, doi: 10.1109/ICTC46691.2019.8940001.

[34] L. Y. Rock, F. P. Tajudeen, and Y. W. Chung, "Usage and impact of the internet-of-things-based smart home technology: a quality-of-life perspective," Univers Access Inf Soc, 2022, doi: 10.1007/s10209-022-00937-0.

[35] S. Tian, W. Yang, J. M. Le Grange, P. Wang, W. Huang, and Z. Ye, "Smart healthcare: making medical care more intelligent," J Glob Health, vol. 3, no. 3, pp. 62–65, 2019, doi: 10.1016/j.glohj.2019.07.001.

[36] S. Sharma, "Smart Home Gardening Management System: A Cloud-Based Internet-of- hings IoT Application in VANET," in 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Khanpur: IEEE, 2020, pp. 1–5. doi: 10.1109/ICCCNT49239.2020.9225573.

[37] G. Cui et al., "Efficient Verification of Edge Data Integrity in Edge Computing Environment," IEEE Trans Serv Comput, vol. 15, no. 6, pp. 3233–3244, Nov. 2022, doi: 10.1109/TSC.2021.3090173.

[38] B. Mishra and A. Kertesz, "The use of MQTT in M2M and IoT systems: A survey," IEEE Access, vol. 8, pp. 201071–201086, 2020, doi: 10.1109/ACCESS.2020.3035849.

[39] A. Badhib, S. Alshehri, and A. Cherif, "A Robust Device-to-Device Continuous Authentication Protocol for the Internet of Things," IEEE Access, vol. 9, pp. 124768–124792, 2021, doi: 10.1109/ACCESS.2021.3110707.

[40] P. Reviriego, J. Martinez, D. Larrabeiti, and S. Pontarelli, "Cuckoo Filters and Bloom Filters: Comparison and Application to Packet Classification," IEEE Transactions on Network and Service Management, vol. 17, no. 4, pp. 2690–2701, Dec. 2020, doi: 10.1109/TNSM.2020.3024680.

[41] L. Luo, D. Guo, O. Rottenstreich, R. T. B. Ma, X. Luo, and B. Ren, "A Capacity-Elastic Cuckoo Filter Design for Dynamic Set Representation," IEEE Transactions on Network and Service Management, vol. 18, no. 4, pp. 4860–4874, Dec. 2021, doi: 10.1109/TNSM.2021.3099433.

[42] W. Han et al., "Cuckoo Search and Particle Filter-Based Inversing Approach to Estimating Defects via Magnetic Flux Leakage Signals," IEEE Trans Magn, vol. 52, no. 4, Apr. 2016, doi: 10.1109/TMAG.2015.2498119.

[43] A. Esfahani *et al*., "A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment," in *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 288-296, Feb. 2019, doi: 10.1109/JIOT.2017.2737630.

[44] Ankur Lohachab, and Karambir, "ECC based inter-device authentication and authorization scheme using MQTT for IoT networks," in Journal of Information Security and Applications, Volume 46, 2019, Pages 1-12, ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2019.02.005.

NARASIMHA SWAMY S. received the bachelor's and M. Tech. degrees from Visveswaraya Technological University (VTU), in 2010 and 2013 respectively. He is currently pursuing Ph. D. degree in the field of Internet of Things with the Academy of Scientific and Innovative Research (AcSIR), Ghaziabad, CSIR-National Aerospace Laboratories, Bengaluru, India. He is also working as an Assistant Professor with the Rashtreeya Vidyalaya College of Engineering (RVCE). He is the author of many articles in international journals, conferences, and book chapters of high repute, including IEEE, Elsevier, and Springer. His research interests include IoT Protocols, Edge Computing/ intelligence, IoT Lightweight Security, and NoSQL databases.

DHEERAJ MANIRATHNAM ANNA. Currently pursuing his bachelor's degree majoring in Artificial Intelligence and Machine with the Rashtreeya Vidyalaya College of Engineering. His research interests include Internet of Things, Edge Computing, IoT Security, Natural language Processing, Blockchain, Machine Learning and Data Visualization.

VIJAYALAKSHMI M N received Ph. D. from Mother Teresa Women's University, Tamil Nadu, in 2010. Currently, she is associated with the Rashtreeya Vidyalaya College of Engineering as an associated professor. She is the author of many articles in international journals, conferences, and book chapters of high repute, including IEEE, Elsevier, and Springer. Her research interests include Data mining, Artificial Intelligence and Machine Learning, Database Management Systems, IoT, and Data Visualization.

SOLOMON RAJU KOTA (Senior Member, IEEE) received the B.E degree from Andhra University, in 1997, and M.E. degree from BITS, Pilani in 2003. He received a Ph. D. degree from IIT Roorkee in 2008. Currently, he is working as a Chief Scientist & Head, ICTD, CSIR-National Aerospace Laboratories, Bengaluru, India. He is the author of many articles in international journals, conferences, and book chapters of high repute, including IEEE, Elsevier, Springer, IGI Global, and De Gruyter. His research interests include advanced electronic system engineering, include High Performance Distributed Computing, reconfigurable computing, Cyber-Physical Systems, Edge Intelligence, Industry 4.0, 5G and beyond communication technologies, IoT Security, Deployment of Hardware for Artificial Intelligence and Structural Health Monitoring, and securing Critical Infrastructure.