International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-6, April 2020

Lightweight Coap Based Authentication Scheme by Applying Two-Way Encryption for Secure Transmission

Pritam S. Salankar, Vinay Avasthi, Ashutosh Pasricha

Abstract*: With the widespread popularity of the Internet of Things (IoT), different sectors-based applications are increasingly developed. One of the most popular application layer protocols is the Constrained Application Protocol (CoAP), and the necessity of ensuring data security in this layer is crucial. Moreover, attackers target the vulnerabilities of IoT to gain access to the system, which leads to a security threat and violate privacy. Typically, user authentication and data encryption are applied for securing data communication over a public channel between two or more participants. However, most of the existing solutions use cryptography for achieving security, with the exception of high computation cost. Hence, these solutions fail to satisfy the resource-constrained characteristics of IoT devices. Therefore, a lightweight security mechanism is required for achieving both secure transmission and better performance. This paper proposes a Lightweight Authentication with Two-way Encryption for Secure Transmission in CoAP Protocol (LATEST) that provides a secure transmission between the server and IoT devices. This mutual authentication mechanism uses ROT 18 Cipher with XoR operation and 128-bit AES based encryption for securing the data transmission. The ROT18 Cipher is a monoalphabetic substitution cipher, which is a combination of ROT13 and ROT5. The proposed scheme employs symmetric encryption in both client and server for ensuring secure authentication and mutually confirm each other identity. In addition, the proposed LATEST scheme ensures confidentiality and integrity by being resistant to replay attacks, impersonation attacks, and modification attacks. The experimental evaluation demonstrates that the proposed LATEST scheme is lightweight and provides better security compared to the existing scheme.

Keywords* : IoT, Secure CoAP, Mutual Authentication, AES based Encryption, XoR operation, lightweight security.

I. INTRODUCTION

Recently, the Internet of Things (IoT) has seen a ubiquitous growth finding its applicability in various real-life domains such as healthcare, industrial appliances, automobile, entertainment, and so on [1].

Revised Manuscript Received on April 30, 2020.
***** Correspondence Author
Pritam S. Salankar* , PhD scholar at UPES, Dehradoon.
Vinay Avasthi, Associate Professor at School of Computer Science UPES Dehardun.
Ashutosh Pasricha, Ph.D from IIT, Delhi

The IoT specializes in providing smart objects through bridging physical objects with the underlying technologies such as pervasive computing, embedded devices, communication technologies, sensor networks, Internet protocols, and applications. As IoT applications deal with sensitive information, security is a significant concern, which emphasis on safeguarding both hardware and networks in the system. The main objectives of IoT security are to ensure the security of the users, preserve privacy and confidentiality, maintain data and devices of the IoT, and guarantee the availability of the services in the IoT environment [2]. The lightweight design of CoAP presents a proper alternative to overcome the resource constraints in traditional protocols such as HyperText Transfer Protocol (HTTP). The CoAP is a standardized application layer protocol that runs over the UDP to perform web-based data transfer [3]. It is particularly beneficial in satisfying requirements such as simplicity, low overhead, and multicast option and supports the transmission of messages between resource-starving physical objects over resource-limited communication networks. In the CoAP protocol, Datagram Transport Layer Security (DTLS) is a secure protocol for protecting the transmission of sensitive information supporting authentication and end-to-end security [4]. The DTLS provides a simple handshake protocol with numerous message exchanges in an asynchronous fashion. Although DTLS originally designed for securely transmitting data, its complex functionality refers to be suitable for networks having abundant resources.

All the protocol stacks in IoT are vulnerable to security attacks; However, application layer protocol being the top layer of the stack is open to prominent security attacks [5]. Due to the application dependent high-level functions in the application layer, a higher level of security is required apart from DTLS [6]. In the application layer, insecure web and cloud interfaces are vulnerabilities that may be an attack vector in an IoT system. For solving these issues, the adoption of authentication is necessary, which is the process of identifying users and devices in a network. Authentication grants access to authorized persons and non-manipulated devices, thereby mitigating attacks such as the reply attack, the Man-in-the-Middle attack, the impersonation attack, and the Sybil attack. Applying biometrics and multi-level authentication for access control provides better security at the application layer [7]. Even though applying high-level security is necessary,

it is more challenging than with a traditional network, due to the heterogeneity of the devices and protocols as well as the scale or the number of nodes in the system [8, 9]. Thus, the design of security features in the CoAP protocol has to consider heterogeneity, resource constraints, privacy, scalability, trust management, and unpreparedness in the IoT system [10]. Therefore, the current research focuses on designing a lightweight authentication mechanism in CoAP based IoT environment. The proposed methodology applies AES (Advanced Encryption Standard) based encryption and ROT18 with XoR operation for achieving a lightweight design with end to end security and reliability.

A. Contributions of the Work

The proposed work has the following contributions:

- The proposed scheme aims to provide a lightweight security implementation in CoAP protocol using based AES encryption and mutual authentication mechanism
- The proposed work designs a mutual authentication mechanism for validating the user's identity in an IoT environment with lightweight processing capabilities.
- The proposed scheme adopts a ROT18 cipher with XoR operation and AES encryption algorithm to secure the authentication between the server and IoT devices.
- The proposed methodology adopts a lightweight encryption algorithm to achieve secure and robust authentication with better performance results.

II. RELATED WORKS

In the CoAP protocol, the traditional DTLS security provides secure data transmission with the exceptions of numerous message exchanges to establish a session. Therefore, it leads to an increase in communication overhead, energy consumption, and communication cost in the network. For tackling this issue, a different enhancement in the DTLS security has been implemented, such as packet compression schemes, TTP enclosures, and ECC optimization. For instance, A novel DTLS based authentication scheme [11] that uses header compression for reducing energy consumption in the network. This method handles origin authenticity, message integrity and lacks in the handling of key exchange. The two-way authentication mechanism [12] is based on a fully authenticated Datagram Transport Layer Security (DTLS) handshake. The scheme supports X.509 certificates containing Rivest, Shamir, and Adelman (RSA) keys. Even though the scheme reduces end-to-end latency and memory overhead, the use of the RSA algorithm significantly leads to heavy computation in the system.

The proposed scheme focuses on lightweight security solutions replacing DTLS security. With extensive research, it is possible to obtain a better understanding of the challenges faced in the existing schemes and thereby providing a better solution. The Short Message Authentication based on Message Authentication Codes (SMACK) [13] aims to protect the resource-constrained IoT devices from denial of sleep attack. The scheme efficiently detects invalid messages. At the same time, SMACK lacks in providing a reactive solution in case of DoS attack occurrence. The lightweight security mechanism proposed in [14] is based on hash and XoR operation in the industrial IoT environment. This lightweight scheme uses pre-shared keys between sensors and routers with the employment of TPM for storing secret keys. This work ensures low computational cost, communication, and memory overhead with support to trusted entities only.

Physical Unclonable Function (PUF) based authentication protocol [15] employs Elliptic Curve Cryptography (ECC) for reducing storage and heavy computation in the resource-constrained devices. The system uses the ECC variant of the ElGamal cryptosystem for encryption. Even though the scheme provides an experimental demonstration of the system performance, it requires a hardware change for its applicability. The authors presented the lightweight mutual authentication protocol [16] based on a public-key encryption system. Despite assuring tradeoff between efficiency and communication cost, the scheme fails to offer optimization of the protocol. The CoAP dependant authentication protocol [17] provides user authentication by allowing access to read/write commands to authenticated users. This method combines both Kerberos and RADIUS protocol to get a reliable authentication and access control mechanism. However, the scheme suffers from heavy computation.

A lightweight authentication and key exchange protocol presented in [18] is based on symmetric-key cryptography and the Hashed Message Authentication Code (HMAC)-based key derivation function (HKDF). The scheme supports two unique keys such as master key and session key, which are provided at the time of configuration. This scheme efficiently provides authentication, key exchange, confidentiality, and message integrity. As the scheme works without using a trusted third party (TTP), the Key exchange and authentication between two nodes require a prior establishment for sharing a secret between two nodes. A lightweight mutual authentication scheme [19] aims to verify the identities of the client in a CoAP based IoT environment. The scheme prefers a 128-bit AES algorithm for encryption to support the resource-constrained IoT nodes. Although the scheme effectively protects against eavesdropping, key fabrication, resource exhaustion, and denial of service attacks, it is vulnerable to Sybil attacks.

The mutual authentication scheme presented in [20] aims to overcome the problems of DTLS protocol using symmetric encryption function and the session key. This authentication scheme reduces the number of messages transmitted by adopting a simple handshake mechanism. However, due to the use of a pre-shared symmetric key, capturing of node leads to leakage of keys. A lightweight authentication and key agreement scheme [21] is based on the Signcryption algorithm designed between the public key cryptography and certificateless cryptography environment. This lightweight authentication achieves user anonymity, non-repudiation, key agreement fairness, and lightweight with secure communication only between legalized users. The user authentication and anonymity scheme [22] is developed for IoT based medical care system. For protecting the data, the authentication mechanism adopts an ECC based encryption scheme providing user anonymity using a dynamic identity mechanism in the authentication process.

This technique provides enhanced computational cost, achieves mutual authentication and session key security. The multi-factor authentication protocol [23] provides user authentication between a medical professional and a cloud server. The scheme uses ECC based secure authentication using multi factors such as password, smart card, and biometrics. This scheme achieves mutual authentication and forward secrecy. The authors in [24] presented a user authentication scheme based on the improved challenge-response mechanism to protect from a replay attack. This scheme provides an efficient mutual authentication mechanism and secure session key agreement. However, the scheme is observed to be vulnerable to attacks such as plaintext attacks, DoS attacks, and impersonation attacks. The two-level session key-based authentication mechanism [25] developed provides secure end-to-end users' communications from DDoS attacks in IoT in an IoT based predicted scenario. This lightweight scheme resists against reply, channel, forward and key regeneration attacks. The scheme lacks in focusing the key freshness and inter-cluster key sharing scenarios.

Notably, owing to the resource constraints of IoT devices, the CoAP protocol faces significant challenges for providing efficient and secure communication between the server and end devices with lightweight computation capabilities. Even though DTLS-based schemes support a wide range of cipher suites for security provisioning, it is suitable for networks having an abundance of resources. The resource-consuming, complex cipher suites of DTLS do not consider the physical characteristics of IoT devices. The major problem faced during the usage of symmetric encryption is that as the same key is employed for encryption and decryption, insecure authentication leads to leakage of the key that compromises the security of both server and client. Similarly, the use of public-key cryptography such as ECC is utilized in the traditional computation for protecting and securing the data transmission. Even though this encryption algorithm provides high memory efficiency than random schemes, the pre-distributed key schemes require a large number of keys to be loaded onto the nodes before deployment, and the capturing of nodes leads to leakage of the keys. Thus, there is a necessity of providing a secure authentication mechanism with lightweight computation.

III. SYSTEM MODEL

Consider a set of IoT devices N = {N1, N2, · · ·, Nn} communicate with the server S. The client-server structure is adopted to represent the devices as nodes and their connections as links. Initially, before the actual authentication process, the available crypto suites and implicit certificates are shared between the server and end IoT devices. In the proposed methodology, the ROT18 cipher and AES encryption algorithm are applied as crypto suites for protecting messages during data transmission. ROT18 is a combination of ROT13 and ROT5. ROT13 is similar to Caesar cipher, which involves simple letter substitution replacing a letter with the 13th letter after it, in the alphabet. Similar to ROT13, the ROT5 scrambling method applies to numeric digits (0 to 9). In cryptography, the simple XOR cipher is a type of additive cipher, that quickly encrypts and decrypts the strings. The other crypto suite in the proposed methodology is AES based encryption. Originally, AES is a variant of Rijndael, which supports a 128 bits fixed block size and various key sizes such as 128, 192, or 256 bits. In the proposed scheme, AES encryption, which is a symmetric block cipher has a key size of 128 bits. It works in Cipher Block Chaining (CBC) mode, which is used for the payload encryption. The key size of 128-bit is sufficient for most of the objects in the IoT paradigm mainly due to the limited resources. Both the server and the end-user (IoT devices) acquire the security credentials before the process starts, along with an assumption that a highly resource-rich server and nodes are known during the registration phase. During the authentication phase, the proposed scheme enables the server and end devices to communicate through a secure network and mutually authenticate each other. At the end of the authentication process, the result of the authentication is output 1 (Accept) or 0 (Reject) respectively. The communication sequence between the two parties (the server and the IoT device) is a unique session and a session identifier S_{id} is used for distinguishing each session.

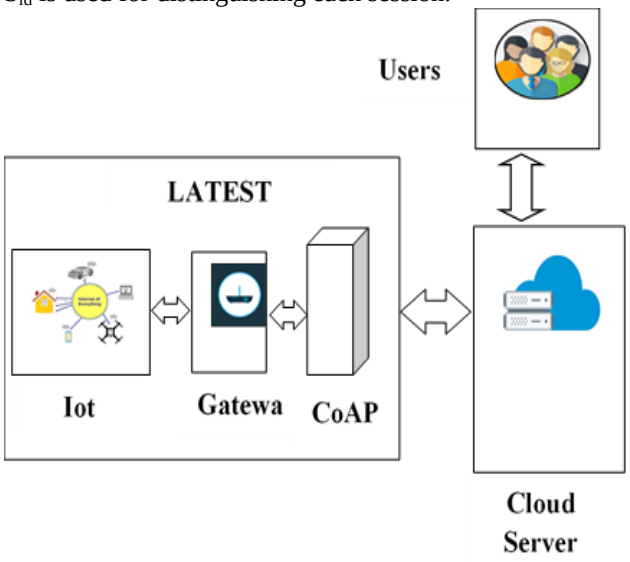


Fig. 1: Architechure Diagram Of Proposed Methodology

CoAP Security Stack: CoAP is a web transfer protocol that relies on a structure consisting of two logically divided layers. The first layer is called the request/response layer, which implements the RESTful paradigm. It allows message interchanges asynchronously among CoAP clients and servers supporting unicast and multicast interactions. The second layer called the message layer is designed for retransmitting lost packets, and CoAP relies on the message layer for reliability. This layer works with four types of messages: CON (Confirmable), NON (non-confirmable), ACK (Acknowledgement), and RST (reset). The CON messages are utilized for ensuring reliable communication, with an acknowledgment from the receiver side which is either a positive or negative response. Whereas, non-confirmable (NON) messages are applicable for unreliable communication where the sender does not expect an ACK as confirmation.

Reset (RST) represents the negative acknowledgment messages sent when the server wakes up from sleep mode and lose the context of the previous state. In CoAP, the messages are encoded in a simple binary format with a short fixed-length binary header and components. The RESTful structure in CoAP depends on GET, POST, PUT, and DELETE methods. Traditionally, DTLS is primarily

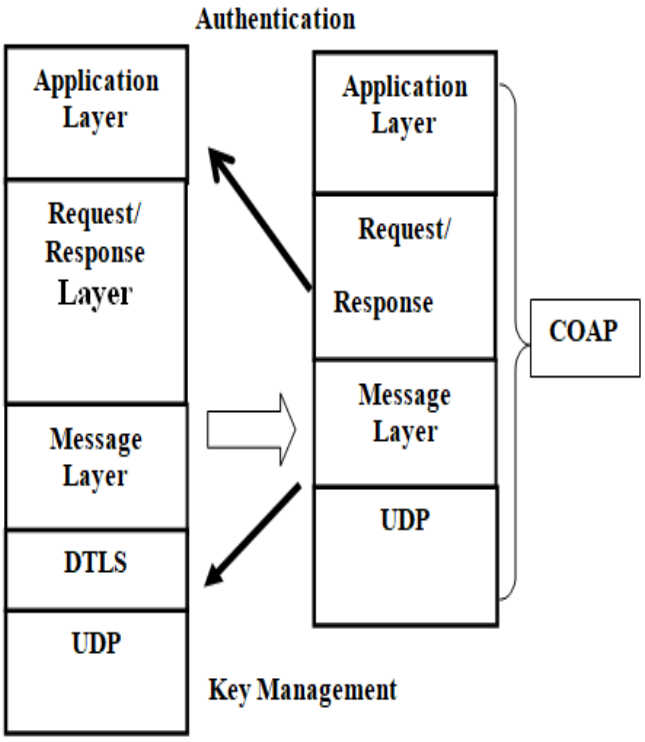


Fig. 2: COAP Security Stack

designed as a security protocol with CoAP for specified facilities such as automatic key management, data encryption, and authentication. However, to replace heavy DTLS, the security implementations are done on a request/response layer and the message layer with a necessity of lightweight computation.

A Security Requirements

The primary security requirements in the IoT environment are confidentiality, integrity, availability, authentication, privacy, resource limitations, and lightweight solution.

Confidentiality*:* Confidentiality refers to the secure transfer of data between the IoT devices and server without any disclosure of any sensitive information to any external third parties other than the communicating end systems.

Integrity*:* Integrity refers to the non-modification of the data along its traversal path in the network. Attackers can either alter the data when it is stored in the node or during data transmissions. Read and write protections and authentication methods provide solutions to these issues.

Availability*:* Availability of software refers to the ability of the IoT applications to provide services for everyone at different places simultaneously. IoT systems need to display sufficient resiliency to sustain availability under desired levels as well as they need to guarantee a certain level of performance requested by their applications.

Privacy*:* Privacy mainly advocates the users to have the tools to control dynamically the data collected, stored, and shared. The user's request has to correlate and satisfy existing policies for making decisions such as granting data access or not.

Authentication: Authentication is the way to confirm the identify the information of IoT entities such as devices and users and to prevent malicious attackers from gaining unauthorized access.

Resource limitations: Most of the embedded sensors have limited resource capabilities in terms of computation, memory, and battery. Since there is a necessity of reliable cryptographic solutions in IoT, computationally expensive solutions need to be avoided. Apart from this, ensuring a high level of security with minimal energy consumption is a hard challenge.

Lightweight Solutions*:* Lightweight solutions are a unique and significant security feature in consideration with the computing and energy capabilities of the devices involved in the IoT.

IV. OVERVIEW OF PROPOSED METHODOLOGY

In IoT, device-server authentication is fundamental; however, most IoT devices are resource-constrained devices, and they also need to transmit sensed data periodically. Therefore, there is a necessity of providing an authentication mechanism with lightweight processing capabilities. The proposed methodology enables a lightweight mutual authentication mechanism to check the validity of server-client identity during data transmission. During the initial session setup, the clients share a 128-bit AES pre-shared secret, P_K, and another key (P_X) with the server. Each IoT device has a unique identifier (ID) associated with it, which enables the server to perform a table lookup for identity verification. The pre-shared secret is known only to the server and the clients to whom it belongs. The security of the authentication is ensured using the symmetric encryption algorithm. Due to the use of symmetric encryption, a single key is shared for both encryption and decryption processes.

In the proposed work, there are three steps in the authentication process. In the first step, the clients initiate the process by registering themselves to the server. During this phase, both server and each client share knowledge of the pre-shared keys, which later is applied for encrypting the request and response messages, respectively. The client sends a request packet consisting of a randomly generated number (R_i)

or nonce and User Identity (ID), which are ciphered using the ROT18 algorithm and then encrypted using an XOR operation. This encrypted packet is sent to the server. In the second step, the server decrypts the request packet using the preshared key and performs an inverse ROT18 operation to retrieve the original packet. The initial authentication of the client is performed in this step. Then, the server performs addition operation on randomly generated numbers (R_j) and R_i. Then the server sends the added nonce (R) with encrypted R_j to the client. Both server and client generate the random numbers R_j and R_i, respectively, using a pseudo-random generator assigned for each session.

The server encrypts the entire response packet using the 128 bit AES symmetric key, which is shared by both server and client. Finally, the client on receiving the response decrypts the data using the symmetric key and performs subtraction to get the random number sent by the server. Then the client compares the obtained value with the decrypted random value to authenticate the device.

V. LATEST MECHANISM

Unlike DTLS based encryption techniques, the proposed lightweight authentication with two-way encryption for Secure Transmission in CoAP Protocol (LATEST) aims to achieve reliable communication between server and IoT devices. In the view of efficiency computation, the proposed scheme uses lightweight encryption algorithms for providing secure authentication. Besides, the LATEST scheme uses a nonce-based authentication to avoid the time-synchronization problem. Considering the communication between two participants, such as server and IoT devices, the scheme involves three phases such as provisioning phase, session initialization phase, and mutual authentication phase. The notations utilized in the proposed scheme are shown in table 1.

Table I: Definitions of Notation in Proposed Methodology

Notations	Definitions
γ_{client}	Client(IoT device) request packet
γ_{server}	CoAP Server Response packet
$\oplus$	Exclusive-OR operation
$\|\|$	Concatenation operator
O_{id}	Object ID
P_K	128-bit AES symmetric Key
P_X	128 bit Key
R_i	Random number generated by the client
R_j	Random number generated by the server

A. Secure Authentication Procedure

Fig. 3 represents the request-response model of the proposed scheme. The step by step procedure of the proposed mechanism is listed below.

Step1: Initialization and provisioning

Initially, the proposed scheme starts with the provisioning phase. The provisioning phase is a prerequisite offline phase, where the clients share a 128-bit AES pre-shared secret, P_K, and 128-bit symmetric key (P_X) for performing XoR operation with the server. Each client (IoT device) is assigned with a unique object ID (O_{id}) and preshared keys (P_K, P_X). The server is provided with all the objects IDs and preshared keys, which are maintained by both server and client.

Step 1.1: Each client generates a random number (R_i) using a pseudo-random generator, and each device owns a unique object ID (O_{id}).

Step 2: Session Initiation

After the provisioning phase, the server and client, initiate the session in the session initiation phase. During this phase, the client sends the session initiation request attached to the ID and random number I created using a pseudo-random generator. Before sending this conformable request, the ROT18 cipher is applied to the request packet. Then, the XoR operation is performed before sending it to the server. The encrypted request packet from the client is represented as

$$\gamma_{Client} = P_X \oplus \{ROT18\{O_{id} \,||\, R_i\}\} \quad (1)$$

Step 2.1: The client attaches its random number(R_i) and object ID(O_{id}) in the request packet and uses ROT 18 on the request packet to convert into an unrecognizable format.

Step 2.2: Then, the ciphered request packet is encrypted by performing XoR operation with a pre-shared symmetric key (P_X) and sends the encrypted request packet to the server.

Step 3: Initial Authentication*:* On receiving the request packet, the server decrypts the packet using the symmetric key and performs a ROT18 cipher operation again on the packet to retrieve the original request packet. In this phase, the initial authentication of the client is performed using the retrieved object ID. On authorization, if the object IDs do not match, the server sends an unauthorized response code and restricts the client in initiating the session.

Step 3.1: The server retrieves the object ID(O_{id}) and a random number (R_i) from the packet by decrypting using P_X symmetric key and ROT18 shift operation.

Step 3.2*:* On successful deciphering, the server confirms the identity of the client by comparing the received O_{id} with the IDs in the lookup table maintained.

Step 3.3: The server generates a random number (R_j) for the mutual authentication phase.

Step 4: Server response and Challenge

The server sends the authentication response consisting of the summed random values (Ri, Rj) and encrypted random number R_j to the client for performing mutual authentication.

This response packet is then encrypted using a 128 bit AES encryption algorithm. The encrypted response packet is sent to the client for authentication. The encrypted payload of the server is represented as

$$\gamma_{Server} = P_K\{R||P_X \oplus \{R_j\}\} \quad (2)$$

Where

$$R = R_i + R_j$$

Step 4.1: The server performs addition operation on the R_j and R_i, presented as R.

Step 4.2: The server combines both R and Rj using concatenation operation (||) where R_j is encrypted using XoR operation ($\oplus$).

Step 4.3: The server encrypts the entire response packet using the preshared key (P_K) and sends it to the client.

Step 5: Mutual authentication

On receiving the encrypted response packet, the legitimate client uses the symmetric key shared with the server to decipher the packet.

Then, it obtains the random number of the server (R_j) by subtracting its random number from R. Then compares the obtained random number with the decrypted random number attached in the response packet. If both the nonce matches, then the authentication is successful, and it leads to the confirmation of the identity of legitimate devices in both ends.

Step 5.1: The client decrypts the response packet using the preshared key (P_K**)** and authenticates itself**.**

Step 5.2: The client retrieves R_j from R and decrypts the attached Rj using XoR operation.

Step 5.3: The client compares the obtained values and authenticates the server.

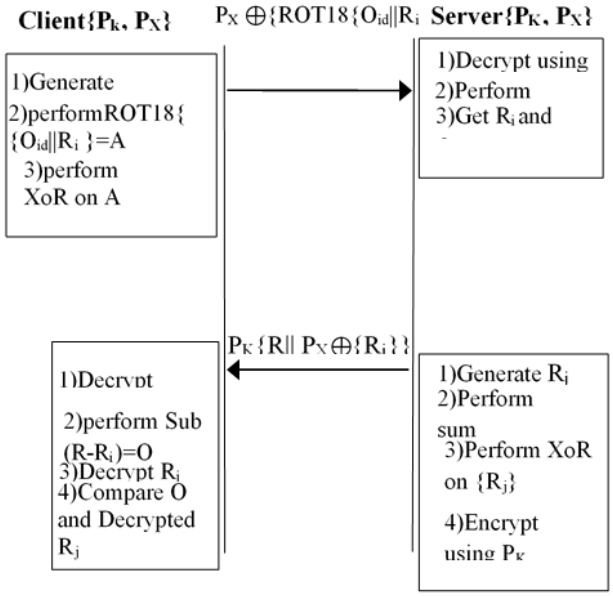


Fig. 3: Proposed Authentication Mechanism Between Server-Client Communication

B. Security Analysis

The security analysis of the proposed authentication mechanism is based on the ability of the system to handle different attacker models and attain a secure mutual authentication between server and IoT devices. The proposed LATEST mechanism can protect the data from different attackers as discussed below.

Resistance to Message replay attack and Impersonation attack: In the replay attack, the adversary re-sends a valid message sent already in order to disturb the traffic flow. Impersonation attackers steal the identity of IoT devices in order to impersonate as valid IoT devices. A random number is attached by both client and server while transferring the request and response message to protect the proposed scheme from replay attacks. Even if an adversary intercepted the message and tried to impersonate a valid IoT device by replaying the message immediately, the server can reject the request because the nonce (R_i) in the replayed messages would be invalid. Similarly, the IoT device also checks and confirms the random number(R_j) sent by the server to prevent replay attacks.

Resistance to Modification attack: The modification attacker changes the packets for affecting the integrity of the information. In the IoT environment, an adversary attempts to modify the authentication and reply messages. However, due to the use of both AES encryption and ROT18 cipher with XoR operation, the scheme can ensure that information cannot be modified. Therefore, this attack is detected because an attacker has no way to obtain the value of the random number to generate a legitimate message as a strong crypto suite encrypts it. If an attacker tries to transmit a modified packet to either server or client, the packet can be easily identified as mutual authentication is performed. Thus, our scheme ensures message integrity.

Perfect forward secrecy: The proposed LATEST mechanism achieves better forward secrecy as even if the preshared key is compromised, the random number established by trustful entities is not affected, since, for each session, the unique random number (RI and R_j) are generated for the client (IoT devices) and server respectively.

Mutual authentication: A mutual authentication is achieved in the proposed mechanism. Both client and server verify their identity through request and response message. In the general authentication procedure, the server authenticates the IoT device through the request message while through response message, the client and server authenticate each other. Even if the attacker intercepts the messages and wants to forge a valid server or IoT device, it has to generate a valid request /response message to either server or client, respectively. However, it is not feasible for the attacker to compute the valid message because he does not know the secure key (P_K, P_X) and the random numbers (R_i, R_j).

C. Storage Cost and Computation Cost

The efficiency of the proposed LATEST mechanism is evaluated in terms of storage cost and computational cost. The computational cost is calculated by the number of symmetric key encryption/decryption operations, the number of signature/verification operations, and the number of random number generation.

Storage Cost: In the proposed LATEST mechanism, as the client is the resource-constrained IoT device, there is a requirement of reducing storage cost. Therefore, the proposed scheme focuses on reducing the computational cost and storage cost in the client-side compared to the server.

Table 2 represents the storage cost of the proposed LATEST scheme, and notations are referred to as given in Table 1. Table 2 shows that the storage cost of the client-side is 768 bits, while the storage cost of the server is 896 bits. Similarly, in comparison to the proposed scheme, the existing scheme has considerable storage costs in terms of 1024 bits on the client-side. The notations considered in evaluating the storage cost of the existing scheme are

O_{id} - node Id of IoT devices
R_d - Random number of client
R_s - Random number of Server
S_K - Session key shared between clients and server
K_d - Preshared Key shared between clients and server
[M1]- $R_d||O_{id}$
[M2]- $k_d[R_s||R_d||S_K]$
[M3]- $S_k[R_s]$

Table II: Storage cost of the Proposed LATEST scheme and Existing Mutual Authentication Scheme

Proposed LATEST Scheme			Existing Mutual Authentication scheme		
Parameters	**Client**	**Server**	**Parameters**	**Client**	**Server**
γ_{client}	✓	-	O_{id}	✓	✓
O_{id}	✓	✓	R_d	✓	✓
R_i	✓	✓	R_s	✓	✓
R_j	-	✓	S_K	✓	✓
γ_{Server}	-	✓	K_d	✓	✓
R	✓	✓	[M1]	✓	-
P_K	✓	✓	[M2]	-	✓
P_X	✓	✓	[M3]	✓	-
Total Storage Cost(bits)	768	896	Total storage cost (bits)	1024	768

Computation Cost: The operation performed, and key sizes to evaluate the computation cost of the proposed scheme. The computation cost of the proposed scheme on the client-side is represented as $2*C_{XoR}+C_{ROT}+C_{AES}+C_{Ri,}$ and the computation cost of the server is presented as $2*C_{XoR}+C_{ROT}+C_{AES}+C_{Rj.}$ In the existing scheme, the computation cost of the client-side is $C_{Ri}+C_{AES}+C_{S,}$ and the computation cost of the server-side is $2* C_S+C_{Rj}+C_{Aes}$.

Where,

C_{RoT}- Cost of RoT18 Cipher operation

C_{AES}- Cost of AES based encryption/decryption

C_{Ri}- Cost of a random number of the client

C_{Rj}-Cost of a random number of the server

C_S-Cost of Session Key Encryption/Decryption

C_{XoR}-Cost of XoR Operation

VI. PERFORMANCE EVALUATION

The proposed work uses the Cooja simulator of the Contiki operating system to evaluate the performance of the proposed LATEST mechanism. The performance comparison is performed between the proposed LATEST mechanism and the existing mutual authentication scheme [20]. In Cooja, the server and clients use wismote mode while the border router uses Z1-Mote, as border router is used only for routing and does not require large storage space compared to server and clients. This work simulates the proposed LATEST in a 100 X 100 m^2 area, where 28 client nodes, one server, and one border router are deployed. The communication range of each node is set to 50m. The message is sent with a data interval of 10 Sec with a size of 127 bytes, and UDP configures the transport layer. The propagation model is the UDGM model. The performance analysis between the proposed LATEST and existing mutual authentication is evaluated in terms of performance metrics such as energy consumption, overhead, and delay. The performance metrics used in the proposed scheme are

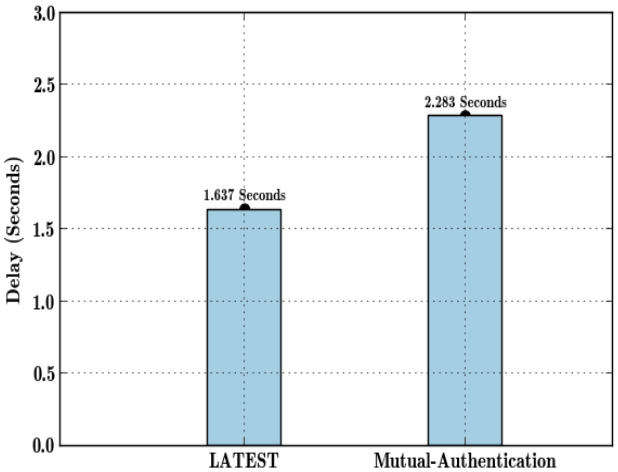
Delay: The delay is defined as the time required for a node to process a confirmable message and issue an acknowledgment.

Message Size Overhead*:* The message size overhead is defined as the total length of the header in the packets transmitted, and it is denoted in terms of bytes.

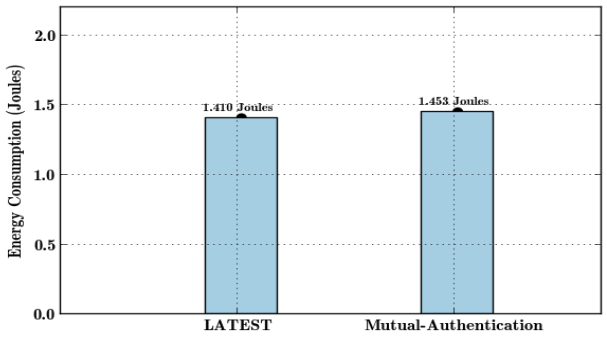
Energy Consumption*:* It is the amount of joules consumed to deliver the data from source to destination.

VII. SIMULATION RESULTS

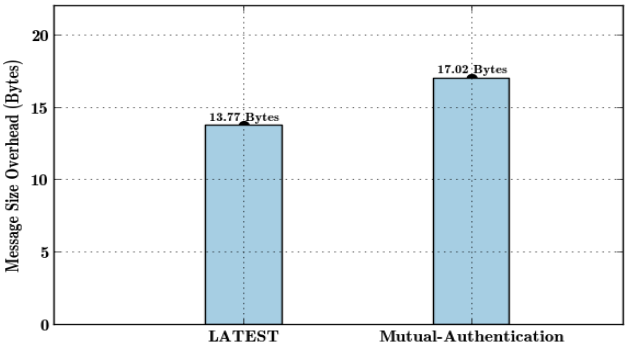
The simulation results are obtained for the proposed LATEST scheme and mutual authentication scheme by comparing the performance in terms of delay, overhead, and energy consumption. fig. 4 shows the performance analysis between the LATEST scheme and the existing mutual authentication scheme. As shown in fig. 4(a), the delay of the proposed scheme is 1.637 seconds, while the delay of the existing mutual authentication is 2.283 seconds. In the existing scheme, the session key is used for encrypting the data transmitted through the encrypted communication channel. This session key and nonce created by the server are sent to the server for encrypting the authentication step. However, since the server generates the session key generation and distribution function, a delay occurs as the client has to fetch the encrypted random number and symmetric key, decrypt the encrypted packet, and use the symmetric key to encrypt its request packet attaching the received random number of the server whereas the proposed scheme uses nonce based authentication and AES based encryption for achieving both mutual authentication and secured transmission of data. In fig. 4 (b), the message size overhead of the proposed scheme is 13.77 bytes as the scheme reduces the number of handshake messages by simplifying the handshaking process compared to the existing mutual authentication scheme. In the proposed scheme, there is no separate key distribution for encrypting authentication, while for mutual authentication and session key distribution are done as separate processes. In fig.4(c), the amount of energy spent in the entire authentication process is 1.410 joules for the proposed scheme, and the existing scheme the energy



(a)



(b)



(c)

Fig. 4: Simulation Results

consumption is 1.453 joules. The energy consumption is more in the existing scheme due to increased handshake messages for authentication compared to the proposed scheme.

VIII. CONCLUSION

The proposed LATEST mechanism presented a lightweight and secure data transmission in the application layer using a random number based authentication and lightweight encryption method for CoAP protocol. The proposed work adapted a combination of the ROT18 cipher method and XoR operation for encrypting the request packet, while AES based encryption is considered for encrypting the response packet. The scheme ensures system security and a secure authentication process with minimal keyspace. The performance of the proposed scheme illustrates that it delivers robust performance in terms of message size overhead, energy consumption, and delay.

REFERENCES

1. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M, "Internet of things: A survey on enabling technologies, protocols, and applications", IEEE communications surveys & tutorials, Vol.17, No.4, pp.2347-2376, 2015.
2. Hassan, W.H, "Current research on Internet of Things (IoT) security: A survey", Computer Networks, Vol.148, pp.283-294, 2019.
3. Shelby, Z., Hartke, K. and Bormann, C, "The constrained application protocol (CoAP)", 2014.
4. Nastase, L, "Security in the internet of things: A survey on application layer protocols", In 2017 21st International Conference on Control Systems and Computer Science (CSCS), pp. 659-666, 2017.
5. RADOVICI, A., Cristian, R.U.S.U. and ŞERBAN, R, "A Survey of IoT Security Threats and Solutions", In 2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet), pp. 1-5, 2018.
6. Alaba, F.A., Othman, M., Hashem, I.A.T. and Alotaibi, F, "Internet of Things security: A survey", Journal of Network and Computer Applications, Vol.88, pp.10-28, 2017.
7. Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T. and Koucheryavy, Y, "Multi-factor authentication: A survey", Cryptography, Vol.2, No.1, p.1, 2018.
8. Ferrag, M.A., Maglaras, L.A., Janicke, H., Jiang, J. and Shu, L, "Authentication protocols for Internet of Things: A comprehensive survey", Security and Communication Networks, 2017.
9. El-hajj, M., Fadlallah, A., Chamoun, M. and Serhrouchni, A, "A survey of internet of things (IoT) Authentication schemes", Sensors, Vol.19, No.5, p.1141, 2019.
10. Oh, S.R. and Kim, Y.G, "Security requirements analysis for the IoT", In 2017 International Conference on Platform Technology and Service (PlatCon), pp. 1-6, 2017.
11. Esfahani, A., Mantas, G., Matischek, R., Saghezchi, F.B., Rodriguez, J., Bicaku, A., Maksuti, S., Tauber, M.G., Schmittner, C. and Bastos, J, "A lightweight authentication mechanism for M2M communications in industrial IoT environment", IEEE Internet of Things Journal, Vol.6, No.1, pp.288-296, 2017.
12. Raza, S., Shafagh, H., Hewage, K., Hummen, R. and Voigt, T, "Lithe: Lightweight secure CoAP for
13. the internet of things", IEEE Sensors Journal, Vol.13, No.10, pp.3711-3720, 2013.
14. Kothmayr, T., Schmitt, C., Hu, W., Brünig, M. and Carle, G, "DTLS based security and two-way authentication for the Internet of Things", Ad Hoc Networks, Vol.11, No.8, pp.2710-2723, 2013.
15. Esfahani, A., Mantas, G., Matischek, R., Saghezchi, F.B., Rodriguez, J., Bicaku, A., Maksuti, S., Tauber, M.G., Schmittner, C. and Bastos, J, "A lightweight authentication mechanism for M2M communications in industrial IoT environment", IEEE Internet of Things Journal, Vol.6, No.1, pp.288-296, 2017.
16. Wallrabenstein, J.R., 2016, August. Practical and secure IoT device authentication using physical unclonable functions. In 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud) (pp. 99-106). IEEE.
17. Li, N., Liu, D. and Nepal, S, "Lightweight mutual authentication for IoT and its applications", IEEE Transactions on Sustainable Computing, Vol.2, No.4, pp.359-370, 2017.
18. Pereira, P.P., Eliasson, J. and Delsing, J, "An authentication and access control framework for CoAP-based Internet of Things", In IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society, pp. 5293-5299, 2014.
19. Rabiah, A.B., Ramakrishnan, K.K., Liri, E. and Kar, K., "A Lightweight Authentication and Key Exchange Protocol for IoT", 2018.
20. Jan, M.A., Nanda, P., He, X., Tan, Z. and Liu, R.P, "A robust authentication scheme for observing resources in the internet of things environment", In 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 205-211, 2014.
21. Seungyong Yoon, Jeongnyeo Kim, "Mutual Authentication Scheme for Lightweight IoT Devices", SECURWARE 2017 The Eleventh International Conference on Emerging Security Information, Systems and Technologies, 2017
22. Liu, Jingwei, Ailian Ren, Lihuan Zhang, Rong Sun, Xiaojiang Du, and Mohsen Guizani, "A Novel Secure Authentication Scheme for Heterogeneous Internet of Thing", arXiv preprint arXiv:1902.03562, 2019.
23. Li, C.T., Wu, T.Y., Chen, C.L., Lee, C.C. and Chen, C.M. An efficient user authentication and user anonymity scheme with provably security for IoT-based medical care system. Sensors, Vol.17, No.7, p.1482, 2017.
24. Dhillon, P.K. and Kalra, S, "Multi-factor user authentication scheme for IoT-based healthcare services", Journal of Reliable Intelligent Environments, Vol.4, No.3, pp.141-160, 2018.
25. Feng, Y., Wang, W., Weng, Y. and Zhang, H, "A replay-attack resistant authentication scheme for the internet of things", In 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Vol. 1, pp. 541-547, 2017.

26. Mahmood, Z., Ning, H. and Ghafoor, A, Lightweight two-level session key management for end user authentication in Internet of Things. In 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and

AUTHORS PROFILE

Mrs. Pritam S. Salankar PhD scolar at UPES, Dehradoon.M.E. in Electronics (Specialization in computer Technology).Research area : Light weight encryption algorithm for Internet of Things

Dr. Vinay Avasthi B.Sc (PCM) HPU Shimla – HP MCA MDU Rohtak – Haryana Phil (Computer Science) MKU Madurai- TN PhD (Computer Science) UPES Dehradu UK,Associate Professor at School of Computer Science UPES Dehardun. He has around 40 publication in a repotted journals / Conference proceedings. His research interest includes Software Engineering, Cloud Security, IOT, Software reusability, Machine Learning, Smart computing. . He is members of CSI, ACM & IEEE.,

Dr. Ashutosh Pasricha Oil Field Services & Equipment sales Director at Schlumberger,South Delhi, India Ph.D from IIT, Delhi in 1999Dissertation Topic: ,Watershed Modelling using Geographical Information M.Tech. (Water Resources Engg.) from REC, Kurukshetra in 1991 B.Tech (Civil) from REC, Kurukshetra in 1989
Certificate Course on Miller Heimen Strategic Selling Training from Certified Instructor Professiona; Affiliation Indian Association of Hydrologists LM 940
Indian Water Resources Society LM 944508
Institution of Engineers (India) AM 71466/9
Indian Society for Technical Education LM 14749
Specialties:
1. Geographical Information System & Remote Sensing Technologies
2.Infrastructure Solution Architecting
3.High End Virtual Reality & Collaboration Technologies/Solutions
4.Communication solutions & design expertise
5. Application of various technologies in Water Resources sector
6.Real time solutions and its implementation
7. Project Management