LEX LOCALIS-JOURNAL OF LOCAL SELF-GOVERNMENT
ISSN:1581-5374 E-ISSN:1855-363X
VOL. 24, NO. 2(2026)

STRENGTHENING DIGITAL LOCAL SELF-GOVERNMENT THROUGH SECURE IOT INFRASTRUCTURE: A LIGHTWEIGHT COAP-BASED AUTHENTICATION FRAMEWORK FOR MUNICIPAL SERVICES

Gladson Oliver S[1], Kalidass J[2], Suguna T[3] and Aswini C[4]

[1,3,4]Department of Information Technology, Government College of Technology, Coimbatore, 641013, India
[2]Department of Computer Science and Engineering, Government College of Engineering, Tiruchirappalli, 620012, India

Corresponding author: Gladson Oliver S (gladson.s.it@gct.ac.in[1])

ABSTRACT

Local government and sub-national authorities increasingly rely on Internet of Things (IoT) infrastructures to support smart municipal services such as traffic management, waste collection, public lighting and environmental monitoring. The reliability and security of these digital infrastructure are critical for effective local self-government, public service continuity and citizen trust. However, most municipal IoT deployments operate under strict resource constraints, making conventional security mechanisms impractical. This paper proposes a lightweight and secure mutual authentication framework designed for resource-constrained IoT devices used in local government and municipal service environments. By leveraging symmetric cryptographic primitives and the Constrained Application Protocol (CoAP), the proposed approach enables secure device authentication without relying on computationally expensive DTLS handshakes. A state-based authentication mechanism is introduced to enhance resistance against replay, impersonation and denial of service attacks. Security analysis and experimental evaluation demonstrate that the proposed framework achieves strong protection against common attack vectors while maintaining low communication, storage and computational overhead. From a local self-government perspective, the proposed approach provides a practical security foundation for scalable and cost-effective deployment of IoT-based municipal services, supporting digital transformation at the sub-national level.

KEYWORDS: Local self-government, IoT, Mutual authentication, Lightweight protocol, CoAP

I. INTRODUCTION

Local self-government institutions play a central role in delivering essential public services and managing urban and rural environments. In recent years, municipalities and sub-national governments worldwide have increasingly adopted digital technologies to improve service efficiency, transparency and responsiveness. Smart local governance initiatives--often implemented under the broader concept of smart--cities rely heavily on Internet of Things (IoT) infrastructures operated and maintained by local authorities.

IoT enabled municipal services, including intelligent traffic systems, waste management, environmental monitoring, water distribution and public lighting depend on large numbers of interconnected, resource-constrained devices deployed across local jurisdictions. The secure operation of these infrastructures is essential not only for technical liability but also for institutional accountability, service continuity and public trust in local administrations. Security breaches or service disruptions at the local level can have direct social, economic and political consequences for communities and local governments.

Despite the growing reliance on IoT technologies in local governance, many municipal deployments face significant constraints related to cost, energy consumption, computational capacity and network reliability. As a result, traditional security solutions based on heavyweight cryptographic protocols or centralised infrastructures are often unsuitable for local government environments. These challenges highlight the need for lightweight, scalable and practical security mechanisms that can support the digital transformation of local self-government without imposing excessive operational or financial burdens.

The growth of the Internet of Things (IoT) has been remarkable, with billions of devices already connected and projections estimating tens of billions of interconnected devices in the

near future [1]. This expansion is driven by advancements in communication technologies, sensor miniaturization, and enhanced data processing capabilities, enabling large-scale deployment of IoT systems without significantly compromising performance. Industries worldwide—including healthcare, manufacturing, agriculture, and smart cities—are increasingly adopting IoT technologies to improve efficiency, optimize operations, and deliver innovative services.

Furthermore, IoT devices typically operate under strict resource constraints in terms of computation, memory, and energy, while the underlying networks are often lossy and low-power in nature [2,3]. Consequently, traditional authentication mechanisms and computationally intensive cryptographic techniques used in conventional networks are unsuitable for IoT applications [4,5].

Key aspects of IoT security include authentication and authorization, data encryption, secure boot and firmware updates, access control, network security, privacy protection, physical security, and continuous monitoring [6]. Overall, efficient and lightweight security mechanisms are required to address the diverse security requirements and threat models of IoT ecosystems. Conventional authentication methods, such as password-based or heavyweight cryptographic approaches, are inadequate for IoT environments due to resource constraints and increased susceptibility to attacks. Most IoT devices lack sufficient computational power and memory to support complex security protocols, rendering them vulnerable to threats such as spoofing, eavesdropping, and unauthorized access [7].

This paper highlights the significance of authentication in IoT systems, discusses existing challenges, and motivates the need for a new lightweight authentication mechanism. The remainder of this paper is organized as follows: Section II reviews related work on authentication mechanisms in IoT environments. Section III presents the proposed lightweight mutual authentication scheme in detail. Section IV provides a comprehensive security analysis, while Section V evaluates the performance of the proposed scheme through experimental comparison. Finally, Section VI concludes the paper and outlines future research directions.

Despite the extensive research on authentication mechanisms for Internet of Things (IoT) environments, many existing schemes rely on computationally expensive cryptographic primitives, public key infrastructures, or DTLS-based handshakes, which are unsuitable for highly resource-constrained devices. Several lightweight approaches proposed in the literature either suffer from high communication overhead, lack comprehensive security guarantees, or require specialized hardware support such as physically unclonable functions. These limitations motivate the need for a practical, lightweight, and secure mutual authentication mechanism tailored for constrained IoT networks.

Novelty and Contributions:

This study contributes to the literature on digital local governance and municipal infrastructure management by addressing the security challenges of IoT-based public service systems. The main contributions of this work are as follows:

1. *Secure IoT infrastructure for local self-government:* The paper presents a lightweight mutual authentication framework that supports secure communication among resource-constrained IoT devices commonly deployed in municipal and local government service environments.
2. *Cost-effective authentication without DTLS for municipal systems*: By eliminating the need for DTLS handshakes, the proposed approach reduces computational and communication overhead, making it suitable for budget and resource-constrained local authorities.

3. *State-based authentication management for scalable municipal deployments*: A server-side Auth-State mechanism is introduced to efficiently manage authentication sessions and mitigate replay and denial-of-service attacks in large scale municipal IoT infrastructures.
4. *Enhanced reliability of smart municipal services*: the integration of nonce-timestamp binding ensures message freshness and operational reliability, supporting continuous delivery of critical local public services.
5. *Empirical validation in constrained environments*: Security analysis and performance evaluation using realistic IoT simulation tools demonstrate the applicability of the proposed framework to real-world local government and municipal service deployments.

II. RELATED WORK

Numerous studies have focused on the design of lightweight mutual authentication mechanisms for IoT devices, aiming to enhance security without imposing excessive computational or communication overhead. This section reviews relevant authentication schemes proposed for IoT environments and discusses their strengths and limitations in relation to the proposed approach.

Zhou et al. [8] proposed a lightweight authentication mechanism for IoT applications based on a cloud-assisted architecture. The authors claimed resistance against several attacks, including insider attacks, guessing attacks, and user identity disclosure. However, reliance on cloud infrastructure may introduce latency and scalability concerns in highly constrained or real-time IoT scenarios. Pelaez et al. [9] presented an enhanced lightweight authentication scheme for cloud-based IoT environments, claiming resistance against insider, disclosure, and offline guessing attacks. Subsequent analysis revealed that the scheme is vulnerable to replay, impersonation, and session key disclosure attacks [10]. Chen et al. [11] proposed an IoT authentication scheme derived from the protocol introduced by Nikooghadam et al. [12]. The protocol in [12] was designed with the intention of overcoming the issues identified in the authentication scheme proposed in [13] by Kumara et al. Several researchers [14–16] later identified multiple vulnerabilities in this protocol, including susceptibility to insider attacks, password guessing attacks, and message modification attacks. Additionally, the protocol does not provide forward secrecy and lacks proper session key update verification mechanisms.

Alzahrani et al. [17] proposed an authentication scheme based on elliptic curve cryptography (ECC) and self-certified public keys to address impersonation and anonymity issues. It was proposed to address the failures of the authentication schemes proposed by Mandal et al. [18] and Islam Biswas [19], which does not ensure the anonymity of the user and is vulnerable to impersonation attacks. Alzahrani et al. developed an authentication mechanism to address various security vulnerabilities and guarantee anonymity among devices in an IoT network. Despite its improvements, subsequent studies have shown that the scheme does not provide comprehensive resistance against all known attack vectors [20].

In 2019, Gope and Sikdar proposed a two-factor lightweight authentication technique for devices in the IoT with the objective of addressing the vulnerability to physical attacks on smart cards that use passwords for two-factor authentication [21]. Gope and sikdar have used physical unclonable functions to address the security issues mentioned above. However, in 2022, Siddiqui et al. [22] reported that the scheme proposed by Gope and Sikdar is vulnerable to session hijacking attacks, impersonation attacks, man-in-the-middle attacks and differential and conventional template attacks.

Authentication mechanisms based on physically unclonable functions (PUFs) have been proposed in [23, 24] to achieve lightweight and energy-efficient mutual authentication. The protocols proposed by these authors are efficient in terms of energy, memory utilization,

communication overhead and computational cost to provide low-cost secure communication and authentication; Although these approaches are efficient, the limited availability and deployment complexity of PUF hardware restrict their practical adoption in current IoT systems [25]. In the work presented in [26], the author proposed a scheme to achieve authentication through a unicast communication channel. This scheme uses a symmetric algorithm for encryption and attempts to reduce the computational overhead on embedded physical IoT devices. In this paper, the authors claim that the DTLS protocols may be configured to implement an authentication scheme that is efficient in terms of energy. However, the author has not evaluated this scheme by configuring the DTLS protocol to implement authentication.

To address computational and storage constraints, several studies have explored symmetric key–based authentication schemes for IoT and smart home environments [27–30]. In [27], the authors proposed an efficient authentication mechanism for securing the smart home application environment to provide secure remote access by applying symmetric key encryption algorithms. In this protocol, the authors claimed that their system is secure against stolen card attacks and synchronization. However, the authors in [28] demonstrated that the scheme presented in [27] does not provide forward secrecy and does not provide security against smart card loss attacks and password guess attacks. In [29], Wazid et al. proposed a user authentication protocol using symmetric key-based algorithms to achieve secure future communications. They claimed that their system is secure against various known and possible attacks. However, subsequent analyses have shown that many of these schemes suffer from issues such as lack of forward secrecy, vulnerability to desynchronization attacks, and excessive overhead for highly constrained devices [31].

In summary, although numerous authentication schemes have been proposed for IoT environments, many of them either incur high computational overhead, rely on complex cryptographic primitives, or fail to provide comprehensive resistance against common attacks. These limitations motivate the development of a lightweight, symmetric key–based mutual authentication scheme that leverages the CoAP protocol without relying on DTLS, as proposed in this work.

III. PROPOSED LIGHTWEIGHT MUTUAL AUTHENTICATION USING AES

This section presents a lightweight mutual authentication technique for IoT client and server devices based on symmetric key encryption. The Advanced Encryption Standard (AES) algorithm is employed for encrypting and decrypting messages exchanged between communicating devices. The server maintains all information required for the execution of the authentication process in a two-dimensional table referred to as the Auth-State table. The structure and fields of the Auth-State table are presented in Table 1. The three phases of the proposed authentication mechanism are listed and explained below.

Table 1: Structure of the 'Auth-State' Table

Device ID	*ID1*	*ID2*	...	...	*IDn*
State code	000	000	…	…	000
Challenge+TS	0	0	…	…	0
Secret key	K_1	K_2	…	…	K_n

Initialization Phase

The initialization phase is a one-time configuration process executed for each new device joining the network. Each device joining the network is assigned a unique device identity (ID_i) by the device manufacturer or the network administrator. It is assumed that a secret key (K_i) corresponding to the device identity ID_i is pre-shared between the client device and the server. For each device ID_i, a new entry is created in the Auth-State table consisting of

four fields: device identity, state code, challenge + timestamp, and secret key. Each table entry is uniquely identified by the corresponding device identity ID_i.

Server Authentication Phase

The server authentication phase is initiated when the client sends an authentication request message to the server. The client generates a 128-bit random nonce Rc_1, which is exclusively OR-ed with its device identity ID_i to produce a 128-bit value T_1. The value T_1 is concatenated with Rc_1 to form a 256-bit value, which is encrypted using the shared secret key K_i to generate the ciphertext C_1.

$$T_1 = Rc_1 \oplus ID_i$$
$$C_1 = AES\_e\{K_i, T_1 \| Rc_1\}$$
$$M_1 = ID_i \| C_1$$

The message M_1 is set as the payload of the CoAP confirmable message, and their option values are set. In the proposed authentication scheme, two additional options, beyond the core options defined in the CoAP specification, are introduced in the CoAP message header.

Option 1: Auth: The Auth option is used to distinguish authentication messages from other CoAP message types. This option is included as a critical and unsafe-to-forward option in the CoAP header. Critical options must be processed by the receiver; otherwise, the receiver responds with a Bad Option error (response code 4.02).

Option 2: Auth type: Another option introduced in the CoAP header is the Auth-type field. The Auth-type field enables the server to distinguish between an initial authentication request and a response to a server-generated challenge, thereby facilitating mutual authentication. Based on the received authentication message with Auth-type = 0, the server may observe four possible states in the Auth-State table, as summarized in Table 2.

Table 2: 'Auth-State' Table States and Their Interpretation.
(While receiving an authentication message with option Auth-type=0)

'Auth-State' state	**State code**	**Challenge+TS**	**Interpretation**
State 0	0	0	No message has been received from the client with identity ID_i after successful completion of last session
State 1	0	Non-Zero	The session has already been created and the received message is not from the genuine client.
State 2	Non-Zero	0	The message is already received from the client with identity ID_i had transmission error or the message was modified in between transmission or an attempt for attack.
State 3	Non-Zero	Non-Zero	The authentication request message is already processed and the reply has already been sent to the client device with identity ID_i. The message might be lost in its transmission or captured by adversary.

If the Auth-State table is in State 0 upon receiving message M_1, the server retrieves the corresponding secret key K_i and decrypts the ciphertext C_1. The encrypted cipher text C_1 is extracted from the message M_1 and decrypted via the key K_i to derive the plain text P_1. In P_1, the most significant 128 bits (P_1m) are exclusively ORed with the least significant 128 bits (P_{1l}) to derive the value ID_x. The derived value ID_x is compared with the unencrypted device identity ID_i to verify message integrity. If the values do not match, the message is considered tampered or corrupted, and the server responds with an unauthorized error (response code 4.01).

$$P_1 = AES\_d\{K_i, C_1\}$$
$$ID_x = P_{1m} \oplus P_{1l}$$

$I_1 = ID_x \oplus ID_i$

If the value of I_1 is zero, ID_x and ID_i match that the message has not been tampered with or modified. Upon successful verification of M_1, the server generates a fresh 128-bit random nonce Rs_1 and appends a timestamp. The nonce and timestamp are stored in the Challenge + TS field of the corresponding entry in the Auth-State table. The server computes $T_2 = Rs_1 \oplus Rc_1$, concatenates it with Rs_1, encrypts the result using K_i, and sends the resulting ciphertext C_2 to the client as message M_2.

$T_2 = Rs_1 \oplus Rc_1$

$M_2 = C_2 = AES\_e\{K_i, T_2 \| Rs_1\}$

C_2 is the response message M_2 sent from the server for the received request M_1. The message M_2 of size 256 bits is sent to the client. Upon receiving M_2, the client decrypts the ciphertext using the shared key K_i and verifies the server-generated nonce. In Plain text P_2, the most significant 128 bits (P_2m) are exclusively ORed with the least significant 128 bits (P_{2l}) to derive the value Rc_{1x}. The value Rc_{1x} is compared with the random number Rc_1 sent to the server in the authentication request and challenge message M_1.

$P_2 = AES\_d\{K_i, C_2\}$

$Rc_{1x} = P_{2m} \oplus P_{2l}$

$I_2 = Rc_{1x} \oplus Rc_1$

If the value of I_2 is zero, Rc_{1x} and Rc_1 are matched. If the verification fails, the authentication process is immediately terminated.

Client Authentication Phase

After authenticating the server, the client generates a fresh 128-bit nonce R_c2. The client computes $T_3 = Rc_2 \oplus Rs_1$, concatenates it with Rc_2, encrypts the result using K_i, and sends the resulting message M_3 to the server.

$T_3 = Rc_2 \oplus Rs_1$

$C_3 = AES\_e\{K_i, T_3 \| Rc_2\}$

$M_3 = ID_i \| C_3$

The message M_3 is set as the payload of the CoAP confirmable type message, and the option field 'Auth' is included in the header option field with an empty value to indicate that the message is an authentication message. The 'Auth-type' option is also included in the header field of the message with the value '1'. The message M_3 is sent to the server. There are four possible states in the 'Auth-state' table, which are given in Table 3 with their interpretations.

Table 3: 'Auth-State' Table States and Interpretation. (While receiving the authentication message with option Auth-type=1)

'Auth-State' state	**State code**	**Challenge+TS**	**Interpretation**
State 0	0	0	Authentication request message with 'Auth-type' = 0 has not yet been received. The current message may be a replay attack attempt by adversary.

State 1	0	Non-Zero	Already session has been created and the received message is not from the genuine client.
State 2	Non - Zero	0	The previous message from the client with 'Auth-type' = 0 was not processed successfully. The current message is not from genuine client.
State 3	Non - Zero	Non-Zero	The authentication request message is already processed and the reply has already been sent to the client device with identity ID_i. The server is waiting for the second message with 'Auth-type' = 1 from client.

If the server is in State 3 and receives a message with Auth-type = 1, it proceeds to verify the client. It reads the 'Challenge+TS' field corresponding to the ID_i from the 'Auth-State' table and extracts the time stamp from the value retrieved from this field. The extracted time stamp (TS) is compared with the current server time stamp (T_s), and the difference (T_d) is calculated. If the timestamp difference exceeds 138 s, the message is rejected as a replay attack. The 'state code' field of the associated entry with device identity IDi in the 'Auth-State' table is incremented by one. The server uses the secret key K_i associated with the client to perform decryption on the received cipher text, and the plain text P_3 is derived. In P_3, the most significant 128 bits (P_3m) are exclusively ORed with the least significant 128 bits (P_{3l}) to derive the value Rs_{1x}. The value Rs_{1x} is compared with the random number Rs_1, which is sent to the client in response to the authentication request and challenge message M_1. The previously sent Rs_1 value is fetched from the Challenge+TS field.

$$P_3 = AES\_d\{K_i, C_3\}$$
$$Rs_{1x} = P_{3m} \oplus P_{3l}$$
$$I_3 = Rs_{1x} \oplus Rs_1$$

If the value of I_3 is zero, Rs_{1x} and Rs_1 are matched. Upon successful verification, the client device is authenticated, and the server sends an acknowledgment message. To establish communication, the server will generate a session object in the server to represent the client, and a common session key will be created by both the client and the server. The session key is 128 bits in length and is created from the value known only to the server and the client. Rs1 and Rc2 are used for creating the session key.

IV. SECURITY ANALYSIS OF THE PROPOSED AUTHENTICATION SCHEME

The authentication scheme presented in the previous section is designed to resist a wide range of attacks commonly observed in IoT application networks. This section evaluates the resilience of the proposed scheme against major security threats and discusses its robustness under different attack scenarios.

Threat Model

System Entities

- **Client device (C):** Resource-constrained IoT node initiating authentication.
- **Server (S):** IoT gateway / server maintaining the *Auth-State* table and shared secrets.
- **Adversary (A):** Network attacker with standard Dolev–Yao capabilities over the communication channel.

Assumptions

1. **Pre-shared secret key:** Each device C shares a unique symmetric key $\mathbf{K_i}$ with S, provisioned securely during the initialization phase.
2. **Cryptographic primitive security:** AES is secure (IND-CPA/PRP assumption) under proper key management; A cannot feasibly recover K_i from observed ciphertexts.
3. **Nonce generation:** Nonces $\mathbf{Rc_1}$, $\mathbf{Rc_2}$, $\mathbf{Rs_1}$ are uniformly random 128-bit values generated by a cryptographically secure PRNG.

4. **Server time availability:** S maintains a reliable clock used to validate message freshness using timestamps (bounded tolerance window).
5. **Secure storage:** K_i is securely stored on C and S. Physical compromise of a node is out-of-scope unless explicitly considered.
6. **CoAP message delivery:** The network may drop, delay, duplicate, or reorder packets (LLN characteristics). The scheme's state handling addresses this at the authentication layer.

Adversary capabilities (Dolev-Yao Network model)

The adversary A can:

- Eavesdrop, intercept, and record all messages (M_1, M_2, M_3).
- Replay previously captured messages.
- Modify, inject, delete, delay, and reorder messages.
- Attempt impersonation of C or S without knowing K_i.
- Launch resource-exhaustion/DoS attempts by flooding authentication requests.

A Cannot:

- Break AES, compute K_i, or decrypt valid ciphertexts without K_i.
- Predict fresh nonces with non-negligible probability.
- Modify encrypted payloads to another valid ciphertext without detection (except with negligible probability).

Security Goals

- **G1 (Mutual authentication):** C authenticates S and S authenticates C.
- **G2 (Message integrity in authentication):** Any modification of authentication messages is detected.
- **G3 (Freshness):** Replayed messages are detected and rejected.
- **G4 (Resistance to impersonation):** A cannot successfully impersonate C or S.
- **G5 (DoS resilience at protocol level):** Authentication attempts should not allocate heavy resources prior to verification.

The following Table 4 summarizes the threat model and table 5 summarizes the security analysis of the proposed scheme.

Table 4: Threat Model and Assumptions

Category	**Description**
Network model	Low-power and lossy network (LLN) using CoAP; packet loss, delay, duplication, and reordering may occur
Adversary model	Dolev–Yao: eavesdrop, intercept, replay, modify, inject, delete, delay, reorder
Cryptographic assumption	AES secure; adversary cannot derive K_i or decrypt ciphertext without K_i
Key provisioning	Unique pre-shared key K_i provisioned securely during initialization
Randomness assumption	128-bit nonces Rc_1, Rc_2, Rs_1 are unpredictable and non-repeating with negligible probability
Time assumption	Server maintains reliable clock; timestamp window used to validate freshness ($\leq$ 138 s)

Table 5: Security Analysis of the proposed Scheme

Attack / Property	**Adversary capability**	**Defense mechanism in proposed scheme**	**Result**
Replay attack	Replays captured M_1 or M_3	Timestamp validation bound to server challenge stored in *Auth-State*;	Replay rejected

		freshness check (≤ 138 s) before accepting M_3	
MITM modification	Modifies $M_1/M_2/M_3$ in transit	Integrity check via XOR consistency after decryption (e.g., $ID_x = P_1m \oplus P_1l$); verification of Rc1 and Rs1 in responses	Modification detected; session aborted
Client impersonation	Sends forged M_1/M_3 with ID_i	Cannot generate valid ciphertext without K_i; server challenge in M_2 must be answered correctly in M_3	Impersonation fails
Server impersonation	Sends forged M_2 to client	Client verifies Rc1 embedded in decrypted payload of M_2; mismatch terminates authentication	Impersonation fails
Eavesdropping	Reads ID_i and ciphertext	ID_i alone insufficient; nonces and responses protected by AES under K_i	No impersonation from observation
Device identity theft	Learns ID_i from plaintext	Encrypted payload implicitly binds to ID_i and fresh nonce; attacker cannot produce valid C_1/C_3 without K_i	Identity theft ineffective
DoS / flooding	Floods server with invalid requests	Early rejection using CoAP options (*Auth*, *Auth-type*), ID_i table lookup, and state-based throttling/alerting	Reduced impact
Desynchronization	Sends out-of-order or repeated messages	*Auth-State* state codes distinguish valid phases and reject unexpected phase messages	Resilient
Session key exposure	Attempts to derive session key	Session key derived from fresh nonces (R_s1, R_c2) unknown to attacker; ciphertext prevents extraction	Not feasible

V. PERFORMANCE ANALYSIS OF THE PROPOSED SCHEME

The proposed authentication scheme was evaluated through implementation on the Cooja network simulation platform. The simulator supports the emulation of various IoT devices and enables detailed performance evaluation. Cooja provides built-in tools for measuring key performance parameters such as execution time, power consumption, and memory usage. For experimental evaluation, the Tmote Sky platform based on the MSP430 microcontroller was used.

Communication Cost

Communication cost is measured as the total number of bytes transmitted over the network during the mutual authentication process. If the size and number of messages transmitted over the network between the authenticating parties remain low, it improves the performance of the network and constrained nodes. The proposed authentication scheme requires the exchange of four CoAP messages (M_1, M_2, M_3, and acknowledgment) to complete mutual authentication. The total communication cost is derived by adding the sizes of the payloads of all these messages. Figure 1 presents a comparative analysis of the communication costs incurred by different authentication schemes.

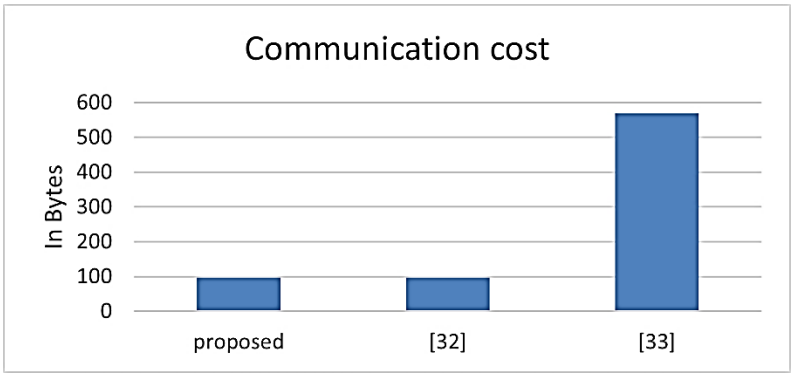

Figure 1: Comparison of the Communication Costs

The scheme proposed by Jan et al. exhibits a communication cost comparable to the proposed scheme, as both utilize CoAP without DTLS handshaking [32]. The authentication scheme proposed by Trabalza et al. imposes a minimum communication overhead of 570 bytes [33].

Handshake Duration

Handshake duration is defined as the time elapsed between the transmission of the initial authentication request and the receipt of the final acknowledgment. In this scheme, handshake duration includes the processing time of four messages by the client, the processing time of four messages by the server and the transmission time of the messages. The measured handshake duration of the proposed scheme is compared with existing schemes proposed by Jan et al. [32] and Trabalza et al. [33], as shown in Figure 2.

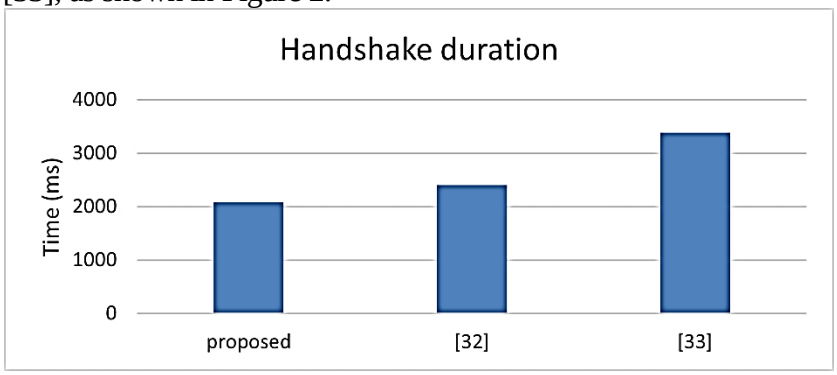

Figure 2: Comparison of Handshake Duration

The proposed scheme demonstrates a shorter handshake duration compared to DTLS-based authentication approaches. The authentication scheme by Jan et al.[32] uses the CoAP protocol for exchanging authentication messages, and security is implemented without using the DTLS protocol. The server-side time complexity of the proposed scheme is O(n), where n represents the number of simultaneous authentication requests. On the other hand, the authentication scheme by the Trabalza et al. [33] used the CoAP protocol in combination with DTLS to accomplish secure authentication between the client and the server. The use of complex cipher suites and operations results in a very long handshake duration in this scheme.

Storage Cost

Storage cost is measured as the total number of bytes required to store authentication-related information on the IoT device. Given the limited memory capacity of IoT devices, minimizing storage overhead is essential. In the case of the authentication process, many of the protocols demand the storage of secret keys, device identities and certain information specific to the authentication technique used.

The proposed authentication scheme is designed to minimize storage requirements while maintaining security. The storage cost is comparable to that of other existing

authentication protocols. The cost incurred here is compared with that of the authentication schemes that use the CoAP protocol without the DTLS and the CoAP protocol in combination with the DTLS protocol. Figure 3 compares the storage requirements of the proposed scheme with existing CoAP- and DTLS-based authentication mechanisms.

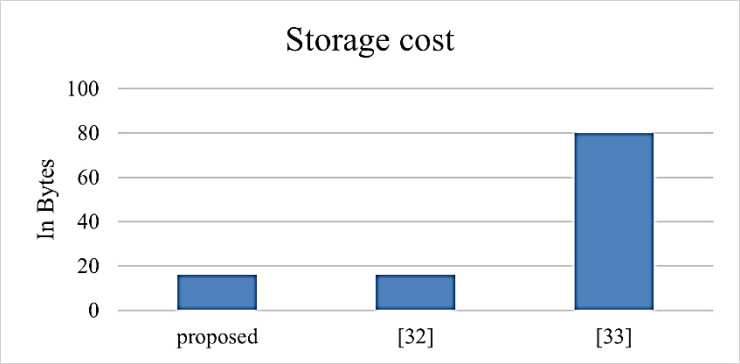

Figure 3: Comparison of Storage Cost

The authentication method proposed by Jan et al. uses the CoAP protocol for enforcing security without the DTLS protocol and incurred the same storage cost as the proposed technique, which also uses the CoAP protocol for secure authentication.

VI. CONCLUSION AND FUTURE WORK

This paper presented a lightweight mutual authentication framework designed to support secure communication in resource-constrained IoT infrastructures operated by local governments and municipal authorities. By relying on symmetric cryptography and CoAP-based communication, the proposed approach avoids the complexity and overhead of conventional security protocols, making it suitable for practical deployment in local self-government environments.

From a governance perspective, the proposed framework contributes to strengthening the digital capacity of local administrations by enabling reliable and secure operation of IoT-based public services. The state-based authentication mechanism and low-overhead design support scalability and resilience in municipal infrastructures while reducing operational and financial burdens on local authorities.

While the current implementation is best suited for relatively static municipal IoT deployments, future work will focus on extending the framework to dynamic local governance scenarios, including secure device onboarding, inter-municipal interoperability and formal verification of security properties. Overall, the proposed approach provides a practical technological foundation for advancing smart local governance and digital self-government.

Declarations

Funding: No funding was received to assist with this research work.

Conflicts of interest/Competing interests: The authors have no relevant financial or non-financial interests to disclose.

Data Availability: No specific dataset was generated or used for this research work.

Ethics Approval: Not Applicable

Consent to Participate: Not Applicable

Consent to Publish: Not Applicable

References:

[1] Satyajit Sinha, “Iot analytics: Global IoT market forecast (in billions of connected IoT devices)”, 2023. online available: https://iot-analytics.com/wp/wp-content/uploads/2023/05/Global-IoT-market-forecast-in-billions-of-connected-IoT-devices.png

[2] Adat V, Gupta BB (2018) Security in Internet of Things: issues, challenges, taxonomy, and architec- ture. Telecommun Syst 67(3):423–441
[3] Hammoudi S, Aliouat Z, Harous S (2018) Challenges and research directions for Internet of Things. Telecommun Syst 67(2):367–385
[4] Samaila MG, Neto M, Fernandes DA, Freire MM, Inácio PR (2017) Security challenges of the Internet of Things. In: Batalla J, Mastorakis G, Mavromoustakis C, Pallis E (eds) Beyond the Inter- net of Things. Springer, Cham, pp 53–82
[5] Sudha MN, Rajendiran M, Specht M et al (2021) A low-area design of two-factor authentication using DIES and SBI for IoT security. J Supercomput. https://doi. org/10. 1007/s11227- 021- 04022-w
[6] Ahmed SRHIR, Tomader MAZRI and Mohammed BENBRAHIM, "Security in the IoT: State-of-the-Art, Issues, Solutions, and Challenges" International Journal of Advanced Computer Science and Applications (IJACSA), 14(5), 2023.http://dx.doi.org/10.14569/IJACSA.2023.0140507
[7] Phillip Williams, Indira Kaylan Dutta, Hisham Daoud, Magdy Bayoumi, "A survey on security in internet of things with a focus on the impact of emerging technologies", Internet of Things, Volume 19, 2022, 100564, ISSN 2542-6605, https://doi.org/10.1016/j.iot.2022.100564.
[8] Zhou, L.; Li, X.; Yeh, K.H.; Su, C.; Chiu, W. Lightweight IoT-based authentication scheme in cloud computing circumstance. future generation computer systems. Future Gener. Comput. Syst. 2019, 91, 244–251.
[9] Pelaez, R.M.; Cruz, H.T.; Michel, J.R.; Garcia, V.; Mena, L.J.; Felix, V.G.; Brust, A.O. An enhanced lightweight IoT-based authentication scheme in cloud computing circumstances. *Sensors* 2019, *19*, 2098.
[10] Yu,S .;Park, K.;Park, Y. A secure lightweight three-factor authentication scheme for IoT in cloud computing environment. Sensors 2019, 19, 3598.
[11] Shouqi, C.; Wanrong, L.; Liling, C.; Xin, H.; Zhiyong, J. An improved authentication protocol using smart cards for the Internet of Things. IEEE Access 2019, 7, 157284–157292
[12] Nikooghadam,M.;Jahantigh,R.;Arshad,H. A lightweight authentication and key agreement protocol preserving user anonymity. Multimed. Tools Appl. 2017, 76, 13401–13423.
[13] Kumari, S.; Chaudhry, S.A.; Wu, F.; Li, X.; Farash, M.S.; Khan, M.K. An improved smart card based authentication scheme for session initiation protocol. Peer Netw. Appl. 2017, 10, 92–105.
[14] Sharma, G.; Kalra, S. A lightweight multifactor secure smart card based remote user authentication scheme for cloud-IoT applications. J. Inf. Secur. Appl. 2018, 42, 95–106.
[15] Chandrakar, P.; Om, H. An extended ECC-based anonymity-preserving 3-factor remote authentication scheme usable in TMIS. Int. J. Commun. Syst. 2018, 31, e3540.
[16] Limbasiya, T.; Soni, M.; Mishra, S.K. Advanced formal authentication protocol using smart cards for network applicants. Comput. Electr. Eng. 2018, 66, 50–63.
[17]Alzahrani, B.A.; Chaudhry, S.A.; Barnawi, A.; Al-Barakati, A.; Shon, T. An Anonymous Device to Device Authentication Protocol Using ECC and Self Certified Public Keys Usable in Internet of Things Based Autonomous Devices. *Electronics* **2020**, *9*, 520. https://doi.org/10.3390/electronics9030520
[18] Mandal, S.; Mohanty, S.; Majhi, B. Cryptanalysis and enhancement of an anonymous self-certified key exchange protocol. Wirel. Pers. Commun. 2018, 99, 863–891.
[19] Islam, S.H.; Biswas, G. Design of two-party authenticated key agreement protocol based on ECC and self-certified public keys. Wirel. Pers. Commun. 2015, 82, 2727–2750.

[20] Sieun Ju, Yohan Park, "Provably Secure Lightweight Mutual Authentication and Key Agreement Scheme for Cloud-Based IoT Environments", 2023, *Sensors* 2023, *23*(24), 9766; https://doi.org/10.3390/s23249766
[21] Gope, P.; Sikdar, B. Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. IEEE internet Things J. 2018, 6, 580–589.
[22] Siddiqui, Z.; Gao, J.; Khan, M.K. An improved lightweight PUF–PKI digital certificate authentication scheme for the Internet of Things. IEEE internet Things J. 2022, 9, 19744–19756.
[23] Aman, Muhammad & Chua, Kee & Sikdar, Biplab. (2017). A Light-Weight Mutual Authentication Protocol for IoT Systems. 1-6. GLOBECOM 2017-2017 IEEE Global Communications Conference. Piscataway: IEEE Press, 10.1109/GLOCOM.2017.8253991.
[24] A. Jain and A. M. Joshi, "Device Authentication in IoT using Reconfigurable PUF," *2019 2nd IEEE Middle East and North Africa COMMunications Conference (MENACOMM)*, Manama, Bahrain, 2019, pp. 1-4, doi: 10.1109/MENACOMM46666.2019.8988545.
[25] B. B. Ehui, Y. Han, H. Guo and J. Liu, "A Lightweight Mutual Authentication Protocol for IoT," in *Journal of Communications and Information Networks*, vol. 7, no. 2, pp. 181-191, June 2022, doi: 10.23919/JCIN.2022.9815201.
[26] A. Bhattacharyya, A. Ukil, T. Bose, A. Pal, Lightweight mutual authentication for coap (wip), draft-bhattacharyya-core-coap-lite-auth-00 (2014).
[27] Vaidya, B.; Park, J.H.; Yeo, S.S.; Rodrigues, J.J. Robust one-time password authentication scheme using smart card for home network environment. Comput. Commun. 2011, 34, 326–336.
[28] Kim, H.J.; Kim, H.S. AUTH HOTP-HOTP based authentication scheme over home network environment. In International Conference on Computational Science and Its Applications; Springer: Berlin/Heidelberg, Germany, 2011; pp. 622–637.
[29] Wazid, M.; Das, A.K.; Odelu, V.; Kumar, N.; Susilo, W. Secure remote user authenticated key establishment protocol for smart home environment. IEEE Trans. Dependable Secur. Comput. 2017, 17, 391–406.
[30] Lyu,Q.; Zheng,N.; Liu,H.; Gao,C.; Chen,S.; Liu,J. Remotely access "my" smart home in private: An anti-tracking authentication and key agreement scheme. IEEE Access 2019, 7, 41835–41851.
[31] Oh J, Yu S, Lee J, Son S, Kim M, Park Y. A Secure and Lightweight Authentication Protocol for IoT-Based Smart Homes. *Sensors*. 2021; 21(4):1488. https://doi.org/10.3390/s21041488
[32] M. A. Jan, F. Khan, M. Alam and M. Usman, "A payload-based mutual authentication scheme for internet of things," Future Generation Computer Systems, vol. 92, pp. 1028–1039, 2019.
[33] D. Trabalza, S. Raza and T. Voigt, "Indigo: Secure coap for smartphones," in Wireless Sensor Networks for Developing Countries, 1st edition. , Springer, Cham, vol. 366, pp. 108–119, 2013.