An EAP-Based Mutual Authentication Protocol for WLAN-Connected IoT Devices

Awaneesh Kumar Yadav, *Student Member, IEEE*, Manoj Misra, *Member, IEEE*, Pradumn Kumar Pandey, *Member, IEEE*, and Madhusanka Liyanage, *Senior Member, IEEE*

***Abstract*—Several symmetric and asymmetric encryption based authentication protocols have been developed for the wireless local area networks (WLANs). However, recent findings reveal that these protocols are either vulnerable to numerous attacks or computationally expensive. Considering the demerits of these protocols and the necessity to provide enhanced security, a lightweight extensible authentication protocol based authentication protocol for WLAN-connected Internet of Things devices is presented. We conduct an informal and formal security analysis to ensure robustness against the attacks. Furthermore, the empirical performance analysis and comparison show that the proposed protocol outperforms its counterparts, reducing computational, communication, storage costs, and energy consumption by up to 99%, 80%, 91.8%, and 98%, respectively. Simulation results of the protocol using the NS3 and its overhead under unknown attacks demonstrate that the proposed protocol performs better in all scenarios. A prototype implementation of the protocol has also been tested to evaluate its feasibility in real-time applications.**

***Index Terms*—Authentication, extensible authentication protocol (EAP), formal verification, network security, wireless local area network (WLAN).**

I. Introduction

WE ARE rapidly moving toward a smart world where almost everything will be digital. The Internet of Things (IoT) is the next phase of technological revolution which is rapidly evolving toward a smart world where the dependence of connected things on wireless and mobile technology will be inevitable. This is due to the fact that IoT applications such as smart city, health monitoring, smart homes, smart factories, smart grid, hospitality, and tourism in real life are changing the way we go about every societal function. With the recent influx of low-cost wireless local area network (WLAN) capable smart IoT gadgets, our reliance on WLAN technology has grown even more [1], [2]. WLAN is widely regarded as an insecure public network because of open-air broadcasting. Any unknown user can intercept or access WLAN communication between communicating parties. As a result, the security of the WLAN (especially authentication) is a severe concern. To resolve this concern, a robust authentication method is required to prevent illegal network access and ensure that only authorized users have access to the network [3]. Authentication is a method of confirming an entity's identity when accessing a resource [4], [5]. The WLAN security architecture is defined by IEEE 802.11i, which outlines the flexible key hierarchy and key exchange between the IoT Device (D) and the authentication server (AS). IEEE 802.11i employs IEEE 802.1x, a secure and reliable authentication framework for establishing a secure connection between the D and AS with the help of AP. The IEEE 802.1x architecture uses the extensible authentication protocol (EAP) framework for a trustworthy base and message exchange [6], [7].

EAP is a framework for facilitating a variety of WLAN authentication techniques known as EAP methods, and Request for Comments (RFC)-3748 [8] contains a detailed description of the EAP framework. Several authentication methods that employ the EAP architecture have been developed and are commonly used in WLANs. However, all these existing protocols fail to protect from newly identified attacks, such as privileged insider attack, traceable attack, and ephemeral secret leakage. Apart from that, most of the authentication protocols do not support the fast reconnect protocol for quick reauthentication, and all the symmetric-based authentication schemes require the secure channel during the registration phase. Therefore, there is a pressing need to design an authentication mechanism that protects from newly identified attacks, supports fast reconnect, eliminates the secure channel requirement, and is suitable for ultralow cost IoT devices (D).

A. Motivation and Contributions

The increasing use of Ds has necessitated the design of security mechanisms for IoT applications. Generally, Ds that require moderate bandwidth use WLAN for communication. However, security (notably authentication) continues to be a

Manuscript received 15 November 2021; revised 23 March 2022 and 5 July 2022; accepted 20 July 2022. Date of publication 29 July 2022; date of current version 13 December 2022. This work was supported in part by the Academy of Finland in 6Genesis Flagship Project under Grant no. 318927, in part by SFI Connect Center under Grant 13/RC/2077_P2, and in part by the INDIFICORE Project. Paper no. TII-21-5067. (*Corresponding author: Awaneesh Kumar Yadav.*)

Awaneesh Kumar Yadav, Manoj Misra, and Pradumn Kumar Pandey are with the Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee 247667, Uttarakhand, India (e-mail: akumaryadav@cs.iitr.ac.in; manojfec@gmail.com; pg201283006@iitj.ac.in).

Madhusanka Liyanage is with the School of Computer Science, University Collage Dublin, Dublin 4, Ireland, and also with the Centre for Wireless Communications, University of Oulu, 90570 Oulu, Finland (e-mail: madhusanka.liyanage@oulu.fi).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2022.3194956.

Digital Object Identifier 10.1109/TII.2022.3194956

1551-3203 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.

significant impediment to WLAN adoption. Therefore, to secure the communication between the D and AS, several symmetric and asymmetric encryption-based authentication protocols have been proposed, with the majority of them relying on the EAP architecture. Asymmetric encryption-based authentication protocols offer excellent security but come at a high cost, making them unsuitable for ultralow cost Ds [9]–[13]. In order to address the cost issue, several symmetric encryption-based authentication protocols are proposed. However, some recent findings [14], [15] reveal that although these protocols are lightweight but do not ensure the prominent security features such as perfect forward secrecy, identity protection, protection from traceability attack, privileged insider attack protection, ephemeral secret leakage, and many of them do not support fast reconnect for quick reauthentication. To the best of our knowledge, all the symmetric encryption-based authentication protocols [9]–[13] need a secure channel during the registration process. However, this is only achievable in private premises, such as smart homes, smart factories, and smart firm, and finding a secure channel is infeasible in public places, such as smart hospital and smart shops.

This article proposes a symmetric key-based authentication and key agreement protocol and shows that authentication and key agreement method relying solely on symmetric key-based operations can offer the same amount of security features as provided by asymmetric key-based methods and at a much lower cost. Further, the proposed method does not need a secure channel during the registration process, which was only possible with the public key-based protocols till date.

The key contributions of this article are as follows:

1) We design a symmetric key-based authentication method that provides the same level of security as public key-based authentication and key agreement protocols.
2) The proposed method removes the necessity of a secure channel during the registration phase. To the best of our knowledge, this is the first symmetric encryption-based protocol that does not require a secure channel at the time of registration. This feature may be essential for the services, such as smart hospitals, smart shops, and smart transport, where users may have to register in the absence of a secure channel.
3) The informal and formal (i.e., Burrows–Abadi–Needham (BAN) logic and Scyther tool) security analysis are conducted to confirm that the proposed protocol offers all the identified security features and securely generates the secret parameters.
4) The empirical performance analysis and comparison demonstrate that the proposed protocol outperforms its counterparts in terms of computational, communication, storage costs, and energy consumption. Furthermore, we compute the overhead under unknown attacks and do simulations using the NS3 tool, which shows that the proposed protocol performs better in all parameters.
5) A prototype implementation of the proposed protocol is done to show its feasibility in real-time application.

The rest of this article is organized as follows. In Section II, we summarize the existing literature on authentication in WLAN, including the research gaps. Section III discusses the preliminaries and backgrounds used in the article. Section IV presents the proposed protocol for mutual authentication. Furthermore, informal and formal security analysis of the proposed protocol is discussed in Sections V and VI. The performance of the proposed protocol is shown in Section VII. Section VIII shows the prototype implementation. Finally, Section IX concludes the article.

II. Related Works

The EAP framework has been used to create a variety of authentication methods. These protocols can be divided into two groups: 1) EAP protocols based on certificates and 2) EAP protocols based on strong passwords.

A. EAP Protocols Based on Certificates

D and AS both utilize certificates to confirm their legitimacy in certificate-based EAP techniques. To establish a reliable authentication approach, EAP-Transport Layer Security (TLS) [16] method was presented. For authentication, this protocol uses certificates. However, it is computationally costly and necessitates a large number of message exchanges. As a result, resource-constrained Ds cannot use this authentication approach. EAP-Tunnelled-Transport Layer Security (TTLS) [17] was developed in response to the constraints of EAP-TLS. Though it also uses certificates but unlike EAP-TLS and EAP-TTLS, only requires a server-side certificate rather than a client-side certificate. It, however, falls short of the cost-cutting goal. As an alternative N Cam-Winget [18] provided an authentication mechanism. When automatic Protected-Access-Credential (PAC) provisioning is enabled, it provides strong protection but fails to save cost and is unable to hide the credentials from the attacker. Shojaie et al. [19] proposed an authentication approach that improves the security of EAP-TLS while incurring a higher cost than EAP TLS. Kumar and Kumar et al. [14] presented an authentication paper that establishes a connection using a combination of certificates and preassigned replies. Moriarty and Farrell [15] proposed an extended version of EAP-TLS to facilitate the identity protection.

B. EAP Protocols Based on Strong Passwords

In the strong password based EAP approaches, D and AS convince each other that they know a secret without really disclosing it. Cheikhrouhou et al. [20] provided a user authentication strategy that also includes a mechanism for key creation, however their scheme lacks the ability to quickly reconnect. An authentication solution for IEEE 802.11 WLANs was presented by Idrissi et al. [21]. Their approach employs asymmetric public-key encryption and complies with all of the RFC-4017 specifications. Additional security needs, such as Denial-of-Service (DoS) attacks, perfect forward secrecy, and lightweight processing, are not met. In the WLAN context, Fan et al. [9] created a user authentication system. Though, it is lightweight but prone to replay attack. Amit and Om [11]

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.

proposed a technique that claims to relieve the server's burden while also meeting all security requirements. An authentication protocol was proposed by Pandey et al. [10]. The fast reconnect mechanism and key generation are not specified in their protocol. Dey et al. [12] presented an EAP authentication system for WLANs that uses dynamic keys. Yadav et al. [13] proposed an authentication mechanism that ensures perfect forward secrecy and identity protection. To address these challenges, the elliptic curve cryptography (ECC) [3], [22], [23] based authentication is proposed in the literature. However, the scheme provides the protection from several types of attacks excepts ephemeral secret leakage and is computationally high.

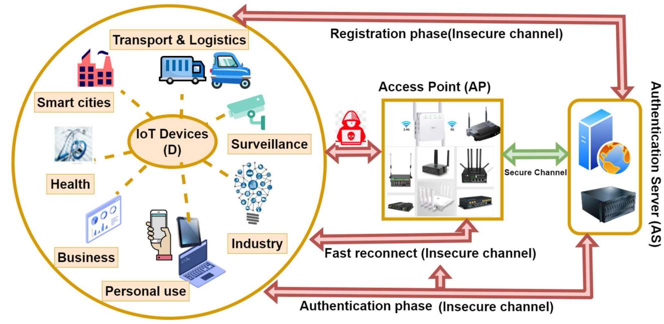


Fig. 1. Network model for IoT-WLAN.

C. Research Gaps in the Existing Authentication Schemes

We observed the following flaws in the existing protocols.

1) Secure channel assumption: None of the existing symmetric encryption based techniques [9]–[13], [15], [20], [21] assume insecure channel between D and AS during the registration phase.
2) Protection from traceable attack: All existing schemes [9]–[14], [16]–[18], [20], [21], fail to provide the protection from traceable attack.
3) Identity protection: The majority of authentication schemes [9]–[13], [16], [20], [21] do not protect identity of D and AS.
4) Prefect forward secrecy: The majority of symmetric authentication protocols [9]–[12], [20], [21] fall short of providing perfect forward secrecy.
5) Privileged insider attack protection: Privileged insider attack prevention is not included in any of the existing symmetric encryption based authentication protocols [9]–[13], [15], [20]–[22].
6) Ephemeral secret leakage: The authentication protocols [3], [10], [11], [13] do not provide protection from ephemeral secret leakage.
7) Fast-reconnect: Majority of the authentication protocols do not support [9], [10], [12]–[14], [19], [22] fast reconnect for quick reauthentication.

III. Preliminaries and Background

The background used in the article is discussed in this section.

A. Network Model

WLAN is a wireless communication network that allows devices to access the network services in a specific range. It is commonly utilized because of its ease of installation. The user can wander throughout the region while staying connected to the WLAN [6]. Fig. 1 represents the network model for IoT-WLAN that involves the following three entities.

1) D that require network access, such as a smartphone, smartwatch, or tablet.
2) Access point (AP) serves as a connection point between the device and the authentication server.
3) AS operates as a backend server in charge of authenticating the device.

When any user or client wants to access the network using the Ds then first it needs to establish a secure connection. To establish a secure connection, authentication is required between the device and authentication server. During authentication, the device and the authentication server verify their authenticity; if they are confirmed to be genuine, the authentication server permits the device to connect to the network via a certain access point within a certain range.

B. Threat Model

We use the widely established "Delev–Yao (DY) [24] and CK-adversary [25] threat model" to test the resilience of the developed protocol. In our threat model, the adversary has the following capabilities.

1) The adversary has complete control over the communication sent over the open wireless channels and can read, delete, or change the messages sent over the wireless channels. Adversary can also insert valid communications.
2) As it is a "computationally infeasible task" to guess multiple values at once, such as identity and password at the same time, the adversary can only guess one value in polynomial time.
3) Adversary has the ability to intercept messages from many sessions and launch a traceability attack.
4) Adversary has the ability to act as a middleman and launch a man-in-the-middle attack.

C. Design Goals

The following are the security goals that the designed authentication technique must meet.

1) Mutual authentication: It specifies that communicating parties (D and AS) must verify each other's validity before transferring any confidential or personal information.
2) Identity protection: To support identity protection, communicating parties' identities should not be sent in plain text via an insecure public channel.
3) Perfect forward secrecy: It assures that even if the attacker has the long-term credentials, he or she will not be able to retrieve the earlier session keys.

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.

4) Replay attack protection: The usage of nonce or timestamp in the protocol is highly suggested to enable replay attack prevention.
5) Protection from ephemeral secret leakage attack: The session key cannot be obtained even if the attacker has the short-term credentials used in the authentication session.
6) Protection from privileged insider attack: If an insider or an attacker gains access to sensitive data of the device, it is hard to get the secret credentials.
7) Protection from traceable attack: It is impossible for an attacker to determine that two different authentication requests are sent by the same device.

IV. Proposed Protocol

This section presents an effective and robust authentication protocol for WLAN communication that overcomes the existing authentication protocols' limitations and security flaws. These are the following three phases in the proposed protocol.

1) Registration phase: During the registration phase, D and AS exchange their secrets using an insecure channel.
2) Mutual authentication phase: With the help of AP, D, and AS confirm their legitimacy and securely procure the session key for data confidentiality and integrity.
3) Fast reconnect phase: When D is detached from an access point due to a network fault and wishes to reconnect with a frequently visited AP, it can quickly reconnect utilizing fast reconnect credentials without having to go through the entire authentication process.

A. Registration Phase

To utilize $AS's$ services, D must first register by entering its identifier and password. This phase is carried out using an insecure public channel, and the steps are outlined as follows:

1) D chooses identity UID, password PW, and random number R_1. Afterwards it computes $D_1 = E_{K'_m}(UID \parallel PW \parallel R_1)$ using public key K'_m of AS and forwards $< D_1 >$ to AS.
2) Upon receiving $< D_1 >$, AS decrypts $D_{K_m}(D_1)$ using private key K_m and checks the database that UID exists. If it exits then AS notifies to D to send another request with a different identifier otherwise AS selects key k, p random number R_2 and compute $Z_{UID} = E_{K_S}(UID \parallel R_2)$, $D_2 = E_{R_1}(k \parallel p \parallel SID \parallel Z_{UID})$. It then sends $< D_2 >$ and stores the $< PW, UID, SID, k, p >$ into his database.
3) When D receives $< D_2 >$ then it decrypts message $D_{R_1}(D_2)$ and saves credentials into his database in encrypted form $J = E_{PW}(k, p, SID, Z_{UID})$ using the PW.

B. Authentication Phase

The mutual authentication procedure between D and AS is carried out during this phase which allows both parties to share a session key that will be used to encrypt future data sent across the network. We assume that the connection between D and AP is unsafe, whereas the connection between AP and AS is secure

TABLE I
NOTATIONS AND MEANINGS

Notations	Meanings
D , UID	D and identity of device
AP, K_S	AP and secret common key used by AS
AS, SID	AS and identity of AS
k, p, L, p_n, & SK	Short-term keys and session key
K_m, K'_m	Private key of AS and public key of AS
Z_{UID}, PW	Masked identity of device and password
H, TK	Hash function and temporary key for fast reconnect
T_1, T_2, T_3, T_4 & r_1, r_2, R_3, R_2	Time stamps and random numbers

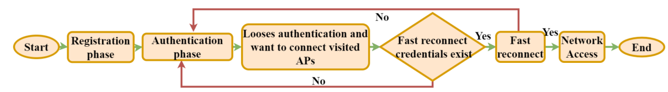


Fig. 2. Flowchart for proposed protocol.

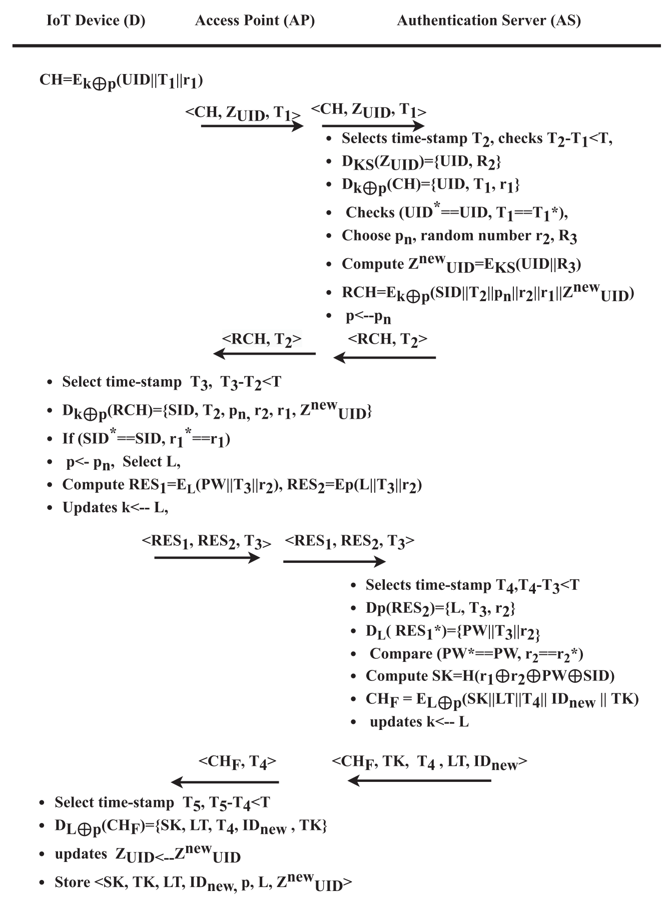


Fig. 3. Proposed protocol.

in our work. We assume that the clocks of D, AP, and AS are synchronized as assumed by many other researchers [3], [4].

Table I summarizes the symbols and abbreviations used in the article, Fig. 2 describes the flowchart of the proposed protocol, Algorithms 1 and 2 describes the pseudo-code of the proposed authentication protocol and Fig. 3 provides a detailed description of the authentication process.

1) $D \rightarrow AP$: When D wants to access network services then it decrypts the $D_{PW}(J)$ to extract the stored credentials (k, p, SID, Z_{UID}). Afterwards, it selects timestamp T_1, random number r_1, and computes the $CH = E_{k \oplus p}(UID \parallel T_1 \parallel r_1)$ and forwards $< CH, T_1, Z_{UID} >$ to AP.

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.

2) $AP \rightarrow AS$: AP forwards this message $< CH, T_1, Z_{UID} >$ to AS.
3) $AS \rightarrow AP$: Upon receiving the message $< CH, T_1, Z_{UID} >$, AS gets timestamp T_2 to verify the freshness of received message by checking the freshness condition $(T_2 - T_1 < T)$. Afterwards AS decrypts $D_{K_S}(Z_{UID})$ using secret key K_S to get identifier UID and based on that it extracts store credentials (k, p, PW, SID) into his database. It then decrypts $D_{k\oplus p}(CH)$ to obtain (UID, T_1, r_1) and compare $(UID == UID*, T_1 == T_1)$, if it matches, it then choose new key p_n, random number r_2, R_3 and compute $Z_{UID}^{\text{new}} = E_{K_S}(UID \parallel R_3)$, $RCH = E_{k\oplus p}$ $(SID \parallel T_2 \parallel p_n \parallel r_2 \parallel r_1 \parallel Z_{UID}^{\text{new}})$. After computing RCH, it updates p by p_n and forwards $< RCH, T_2 >$ to AP.
4) $AP \rightarrow D$: AP forwards this message $< RCH, T_2 >$ to D.
5) $D \rightarrow AP$: When D receives $< RCH, T_2 >$, get timestamp T_3 and verify the freshness of the received message by checking $(T_3 - T_2 < T)$. If it matches then it decrypts $D_{k\oplus p}(RCH)$ to obtain credentials $(SID, T_2, p_n, r_2, r_1, Z_{UID}^{\text{new}})$ and compare $(SID == SID*, r_1 == r_1*)$, if it matches, then D believes that AS is authentic and selects new key L, updates p by p_n to compute $RES_1 = E_L(PW \parallel T_3 \parallel r_2)$, $RES_2 = E_p(L \parallel T_3 \parallel r_2)$. After computing RES_1, RES_2, D updated k by L and forwards $< RES_1, RES_2, T_3 >$ to AP.
6) $AP \rightarrow AS$: AP forwards this message $< RES_1, RES_2, T_3 >$ to AS.
7) $AS \rightarrow AP$: After receiving the messages $< RES_1, RES_2, T_3 >$ from the AP, AS gets timestamp T_4 and verifies the freshness of the received message by checking $(T_4 - T_3 < T)$. If it matches, it then decrypts [i.e., $D_p(RES_2)$] to obtain the L. After getting L, it decrypts $D_L(RES_1) = (PW, T_3, r_2)$ and compare $(PW == PW*, r_2 == r_2*)$, if it matches then it believes that D is authentic and selects a temporary new identity ID_{new} for D (i.e., D will use this identity during the fast reconnect authentication process), and lease time (LT) for the session key. (i.e., defines the temporary key expiry time and also ensures that this is unique for every D) and compute $SK = H(r_1 \oplus r_2 \oplus PW \oplus SID)$, $CH_F = E_{L\oplus p}(SK \parallel LT \parallel T_4 \parallel ID_{\text{new}} \parallel TK)$. After computing CH_F, AS updates k by L, store $< L, p >$ and forwards $< CH_F, TK, LT, ID_{\text{new}}, T_4 >$ to the AP.
8) $AP \rightarrow D$: When AP receives the message $< CH_F, TK, LT, T_4, ID_{\text{new}} >$, it saves $(TK, LT, ID_{\text{new}})$ into its database and passes the $< CH_F, T_4 >$ to D.
9) After receiving the message $< CH_F, T_4 >$ from AP, D selects the timestamp T_5 to verify the freshness condition $(T5 - T4 < T)$, if it matches, it then decrypts the message $D_{L\oplus p}(CH_F)$ to obtain the credentials (LT, ID_{new}) and saves the credentials $(SK, LT, ID_{\text{new}}, TK, p, L, Z_{UID}^{\text{new}})$ for further communication.

C. Proposed Protocol for Fast Reconnect

When D is disconnected from AP due to some network issue and wants to rejoin again with frequently visited AP, it can easily connect with the network using fast reconnect credentials without requiring the full authentication process (i.e., mutual authentication protocol between D and AS will not be invoked again).

1) $D \rightarrow AP$: D selects timestamp T_1 and random number r_1 to compute the $R_{\text{auth}} = E_{TK}(T_1 \parallel r_1 \parallel ID_{\text{new}} \parallel LT)$,. After computing the R_{auth}, it forwards $< R_{\text{auth}}, T_1, LT >$ to AP.
2) $AP \rightarrow D$: Upon receiving message $< R_{\text{auth}}, T_1, LT >$ from D, AP gets timestamp T_2, and a random number r_2. It then verifies the freshness condition $(T >= T_2 - T_1)$, if the freshness condition holds then it searches the pair (ID_{new}, TK) into its database based on the received LT. After that it decrypts the $D_{TK}(R_{\text{auth}})$ to obtain the credentials $(T_1, r_1, LT, ID_{\text{new}})$. These credentials are compared with the stored credentials $(T_1 = T_1^*, ID_{\text{new}} = ID_{\text{new}}^*, LT = LT^*)$, if they match then AP believes that D is authentic and computes the $R`_{\text{auth}} = E_{TK}(T_2 \parallel r_2 \parallel TSK)$, and temporary session key $TSK = H(ID_{\text{new}} \parallel r_1)$.
After computing the $R`_{\text{auth}}$, $AP$ forwards the $(R`_{\text{auth}}, T_2)$ to D.
3) $D \rightarrow AP$: After receiving message $< R`_{\text{auth}}, T_2 >$ from the access point, $D$ selects the timestamp $T_3$ to verify the freshness of the message by checking the freshness condition $T >= T_3 - T_2$, if it holds then it decrypts the $D_{TK}(R`_{\text{auth}})$ to obtain the credentials (T_2, r_2, TSK). It then computes the temporary session key $TSK = H(ID_{\text{new}} \parallel r_1)$, and compares the obtained credentials with its own credentials $(T_2 = T_2^*, TSK = TSK^*)$. If they match then it believes that AP is authentic and forwards the successful message to AP.

V. Informal Analysis

In this part, we examine the security of the proposed mutual authentication protocol informally, demonstrating that it has additional security characteristics.

Proposition 1: The proposed protocol facilitates the mutual authentication.

Proof: When D receives message $< RCH, T_2 >$ from the AS, D decrypts message (RCH) and verifies the credentials ($SID^*== SID$ and $r_1^*{==}r_1$). If credentials are matched, D believes that AS is authentic. Otherwise, it terminates the authentication process. On the other hand, when AS receives message $< RES_1, RES_2, T_3 >$, it decrypts message and compares the credentials $(r_2^* == r_2, PW^* == PW)$; if they match, AS believes that D is authentic. Thus, our proposed protocol facilitates mutual authentication. ■

Proposition 2: The proposed protocol for mutual authentication is resilient against identity protection and traceable attack.

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Executed by IoT device D.

Input: Value stored at D

Output: $SK = r_1 \oplus r_2 \oplus PW \oplus SID$

Stpe-1. Select random number r_1, get the timestamp T_1;
  compute $CH = E_{k \oplus p}(UID \parallel T_1 \parallel r_1)$;
  send (CH, Z_{UID}, T_1) to AP;
Stpe-2. /**Wait for the message from* AP */
  receive (RCH, T_2) from AP;
  /* *message received, go ahead**/
  get current timestamp T_3;
If $((T_3 - T_2 < T)\&\&(SID == SID^*)\&\&(r_1 == r_1^*))$ **then**
  $p \leftarrow p_n$;
  Select L;
  $RES_1 = E_L(PW \parallel T_3 \parallel r_2)$;
  $RES_2 = E_p(L \parallel T_3 \parallel r_2)$;
  send (RES_1, RES_2, T_3) to AP;
else
  ABORT;
Stpe-3. /* *Wait for the message from* AP*/.
  receive (CH_F, T_4) from AP;
  /* *message received, go ahead**/
  get current timestamp T_5
  If $(((T_5 - T_4 < T)$ **then**
    $SK = H(r_1 \oplus r_2 \oplus PW \oplus SID)$;
    $Z_{UID} = Z_{UID}^{new}$;
Return SK;

Algorithm 2: Executed by Authentication Server AS.

Input: Value stored at AS

Output: $SK = r_1 \oplus r_2 \oplus PW \oplus SID$

Stpe-1. /**Wait for the message from* D*/
  receive (CH, Z_{UID}, T_1);
  /* *message received, proceed* */;
  get current timestamp T_2;
If $(((T_2 - T_1 < T)\&\&(UID == UID^*)\&\&(T_1 == T_1^*)) = True$ **then**
  compute $RCH = E_{K \oplus p}(SID \parallel T_2 \parallel p_n \parallel r_2 \parallel r_1 \parallel Z_{UID}^{new})$;
  $p \leftarrow p_n$;
  send (RCH, T_2) to AP;
else
  ABORT;
  Stpe-2. /* *Wait for the message from* D*/
    receive (RES_1, RES_2, T_3);
    /* *message received, go ahead**/
    get timestamp T_4;
  If $(((T_4 - T_3 < \Delta T)\&\&(PW == PW^*)\&\&(r_2 == r_2^*))$ **then**
    compute $SK = H(r_1 \oplus r_2 \oplus PW \oplus SID)$;
    compute $CH_F = E_{L \oplus p}(SK \parallel LT \parallel T_4 \parallel ID_{new} \parallel TK)$;
    $k \leftarrow L$;
    sends (CH_F, T_4);
  else
    ABORT and goto Step-1;
Return SK;

Proof: The D and AS identities are always exchanged in masked form in the proposed protocol. Identity of D is exchanged in masked encrypted form as Z_{UID}. Server's identity is also exchanged in encrypted form in RCH. As a result, collecting the exchanged messages will not provide any information regarding the identity of the communicating parties. Further $D's$ encrypted masked identity is changed after each successful authentication session to Z_{UID}^{new}. So, even if attacker captures the messages from multiple sessions he cannot link the messages of one session to another session.

Proposition 3: The proposed protocol is resilient against ephemeral secret leakage (ESL) attack. ■

Proof: If the attacker gets access to the ephemeral secrets r_1 and r_2, he will be unable to compute the session key $SK = H(r_1 \oplus r_2 \oplus PW \oplus SID)$ since the attacker can only do so if he has access to long-term credentials (PW, SID). The session key is calculated using both long- and short-term credentials, and obtaining both is computationally impossible. As a result, our proposed protocol is resilient against ephemeral secret leakage (ESL) attack.

Proposition 4: The proposed protocol ensures perfect forwards secrecy.

Proof: If an attacker obtains long-term credentials (UID, PW, SID), he will be unable to obtain session key $SK = H(r_1 \oplus r_2 \oplus PW \oplus SID)$ because an attacker can only obtain r_1 and r_2 if he has (k, p), which have already been replaced by new keys (L, p_n) after successful authentication. As a result, even if an attacker has (UID, PW, L, p_n, SID), getting (SK) is difficult. As a result, our proposed protocol ensures perfect forward secrecy. ■

Proposition 5: The proposed protocol is resilient against the replay attack.

Proof: The proposed protocol utilizes timestamps to verify the freshness of the exchanged messages [i.e., $(CH, T_1, Z_{UID}), (RCH, T_2), (RES_1, RES_2, T_3), (CH_F, T_4)$] by checking the freshness condition $(T_i - T_{i-1} < T)$. If it matches, then it believes that message is fresh otherwise abort the process. Hence, this implies that our proposed protocol is resilient against the replay attack.

Proposition 6: The proposed protocol is resilient against privileged insider attack.

Proof: If an attacker gains access to $D's$ database or eavesdrops stored credentials, he will be unable to retrieve the secrets since they are encrypted with PW, which is kept in a secure location. So, the proposed protocol is resilient against privileged insider attack. ■

Proposition 7: The proposed protocol is resilient against jamming/desynchronization attack.

Proof: D or AS may abort and reinitiate the authentication process at any step either because the freshness condition is not met or because credentials in the received message do not match

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.

TABLE II
BAN NOTATIONS AND FORMULAS

Symbol	Description
$D \mid\equiv V$, $D \triangleleft V$ $D \mid\sim V$, $D \Rightarrow V$ $\#(V)$, $\langle V \rangle_K$, $\{V\}_K$, $D \mid\equiv D \overset{K}{\longleftrightarrow} AS$	D believes V, D receives V D once sent V, D has full control over V V is fresh, V is combined with K V is encrypted with K, D believes that K is shared between D and AS.
$\frac{D \mid\equiv D \overset{K}{\longleftrightarrow} AS, D \triangleleft \{V\}_K}{D \mid\equiv AS \mid\sim V}$	Message meaning (MM) rule
$\frac{D \mid\equiv \#(V), D \mid\equiv AS \mid\sim V}{D \mid\equiv AS \mid\equiv V}$	Timestamp verification (TV) rule
$\frac{D \mid\equiv AS \Rightarrow V, D \mid\equiv AS \mid\equiv V}{D \mid\equiv V}$	The jurisdiction rule (JR)

with stored credentials. This may lead to a situation where after abort keys k and p with D and AS do not match. To handle this scenario D and AS make a copy of k and p before starting the authentication process. D and AS discard these copies only when they are sure that the authentication process has been successfully completed. As stated earlier, the channel between D and AP is not secure but reliable. D, after successfully receiving CH_F, sends an Acknowledgement to AP. It then waits for a time T. If no retransmission is received, it discards the copies of k, and p. AP then informs AS, and AS also discards copies of k and p. In all other cases, D and AS revert back to copies of k and p saved before starting the authentication process. ■

VI. FORMAL ANALYSIS

In this section, we use BAN logic [26] and Scyther tool [27] to perform a formal analysis of the proposed protocol.

A. Security Verification Using BAN Logic

D and AS are the communicating agents, V is the statement, and K is the key. The notations used to define the BAN logic and assumptions for the proposed protocol are shown in Table II.

1) For the protocol's initial state, the following assumptions apply.
   $R_1 : D \mid\equiv D \overset{k}{\leftrightarrow} AS,\ R_2 : D \mid\equiv D \overset{p}{\leftrightarrow} AS.$
   $R_3 : D \mid\equiv \#(T_2),\ R_4 : D \mid\equiv \#(T_4).$
   $R_5 : D \mid\equiv D \overset{L}{\leftrightarrow} AS,\ R_6 : D \mid\equiv AS \Rightarrow p_n.$
   $R_7 : AS \mid\equiv D \overset{k}{\leftrightarrow} AS.$
   $R_8 : AS \mid\equiv D \overset{p}{\leftrightarrow} AS,\ R_9 : AS \mid\equiv \#(T_1).$
   $R_{10} : AS \mid\equiv \#(T_3),\ R_{11} : AS \mid\equiv D \Rightarrow L.$
   $R_{12} : D \mid\equiv AS \Rightarrow (D \overset{SK}{\leftrightarrow} AS),$ $R_{13} : AS \mid\equiv D \overset{p_n}{\leftrightarrow} AS.$
   $R_{14} : AS \mid\equiv D \Rightarrow (D \overset{SK}{\leftrightarrow} AS), R_{15} : AS \mid\equiv D \overset{SID}{\leftrightarrow} AS.$
2) The proposed protocol's security goals are: $AS \mid\equiv (D \overset{SK}{\leftrightarrow} AS), D \mid\equiv (D \overset{SK}{\leftrightarrow} AS)$
3) Idealized form of the proposed protocol is as follows.
   M_1: $D \rightarrow AS$: $(UID \,\|T_1 \,\|\, r_1)_{k \oplus p}$.
   M_2: $AS \rightarrow D$:$(SID \,\|T_2 \,\|r_1 \,\|\, r_2 \,\|\, D \overset{p_n}{\leftrightarrow} AS)_{k \oplus p}$.
   $M_{3.1}$: $D \rightarrow AS$: $(PW \,\|\, T_3 \,\|\, r_2)_L$.
   $M_{3.2}$: $D \rightarrow AS$:$(r_2 \,\|\, T_3 \,\|\, D \overset{L}{\leftrightarrow} AS)_{p_n}$.
   M_4: $AS \rightarrow D$: $(LT \,\|\, ID_{\text{new}} \,\|\, TK \,\|\, T_4 \,\|\, AS \overset{SK}{\leftrightarrow} D)_{L \oplus p}$.

1) Validation and Derivation of Security Goals:

1) Based on R_7 and R_8, we apply MM rule on M_1.
$$S_1 : AS \mid\equiv D \mid\sim M_1.$$
2) Based on R_9, and S_1, we apply TV rule on M_1
$$S_2 : AS \mid\equiv D \mid\equiv (UID, r_1).$$
3) Based on R_1 and R_2, we apply MM rule on M_2.
$$S_3 : D \mid\equiv AS \mid\sim M_2.$$
4) Based on R_3 and S_3, we apply TV rule on M_2.
$$S_4 : D \mid\equiv AS \mid\equiv (SID, p_n, r_2, r_1).$$
5) Based on R_6, and S_4, we apply JR rule on M_2.
$$S_5 : D \mid\equiv D \overset{p_n}{\leftrightarrow} AS.$$
6) Based on R_{13}, we apply MM rule on $M_{3.2}$.
$$S_6 : AS \mid\equiv D \mid\sim M_{3.2}.$$
7) Based on R_{10}, and S_6, we apply TV rule on $M_{3.2}$.
$$S_7 : AS \mid\equiv D \mid\equiv (r_2, D \overset{L}{\leftrightarrow} AS).$$
8) Based on R_{11}, and S_7, we apply JR rule.
$$S_8 : AS \mid\equiv D \leftrightarrow LAS.$$
9) Based on S_8, we apply MM rule on $M_{3.1}$.
$$S_9 : AS \mid\equiv D \mid\sim M_{3.1}.$$
10) Based on R_{10}, and S_9, we apply TV rule on $M_{3.1}$.
$$S_{10} : AS \mid\equiv D \mid\equiv (PW, r_2).$$
11) Based on R_{15}, S_2, S_{10} and $SK = (r_1 \oplus r_2 \oplus PW \oplus SID)$, we can infer S_{11}.
$$S_{11} : AS \mid\equiv D \mid\equiv D \overset{SK}{\leftrightarrow} AS.$$
12) Based on D_{14}, we apply JR rule on S_{11}.
$$S_{12} : AS \mid\equiv D \overset{SK}{\leftrightarrow} AS - Goal_1.$$
13) Based on R_5 and S_5, we apply MM rule on M_4.
$$S_{13} : D \mid\equiv AS \mid\sim M_4.$$
14) Based on R_4, we apply TV rule on M_4.
$$S_{14} : D \mid\equiv AS \mid\equiv ((D \overset{SK}{\leftrightarrow} AS), LT, ID_{\text{new}}, TK).$$
15) Based on R_{12} and S_{14}, the JR rule.
$$S_{15} : D \mid\equiv (D \overset{SK}{\leftrightarrow} AS)\ Goal_2.$$

B. Security Verification Using Scyther Tool

Scyther is a formal verification tool used to prove or disprove the security of protocols. The security protocols are modeled with the security protocol description language (.spdl). The proposed protocol's security properties are verified using the scyther tool. As shown in Fig. 4, the validation result clearly shows that our proposed protocol addresses all of the security claims such as Alive (i.e., guarantees that the communicating

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.

Scyther results : verify

Claim				Status	Comments
MA	D	MA,D2	Secret r_2	Ok	No attacks within bounds.
		MA,D3	Secret p_n	Ok	No attacks within bounds.
		MA,D4	Alive	Ok	No attacks within bounds.
		MA,D5	Weakagree	Ok	No attacks within bounds.
		MA,D6	Niagree	Ok	No attacks within bounds.
		MA,D7	Nisynch	Ok	No attacks within bounds.
		MA,D8	Commit AS,r_1,r_2,PW,SID	Ok	No attacks within bounds.
	AS	MA,AS2	Secret r_1	Ok	No attacks within bounds.
		MA,AS3	Secret L	Ok	No attacks within bounds.
		MA,AS4	Alive	Ok	No attacks within bounds.
		MA,AS5	Weakagree	Ok	No attacks within bounds.
		MA,AS6	Niagree	Ok	No attacks within bounds.
		MA,AS7	Nisynch	Ok	No attacks within bounds.
		MA,AS8	Commit D,r_1,r_2,PW,SID	Ok	No attacks within bounds.

Done.

Fig. 4. Scyther tool result for mutual authentication.

parties carry out all events), Weakagree (i.e., guarantees that the protocol is not vulnerable to impersonation attacks), Nisynch (i.e., guarantees that the sender sends all messages and that the recipient receives them), and Secret specified by scyther tool. As a result, we may conclude that the Scyther tool found no vulnerabilities in the proposed protocol.

VII. Performance Analysis

This section outlines the comparison of security features and experimental analysis to compute the cost of proposed protocol in terms of computational, communication, storage costs, and energy consumption to examine the efficacy of the proposed protocol. In addition to this, we also demonstrate the overhead under unknown attacks and simulate the proposed protocol using the NS3 tool.

A. Experimental Analysis Using the MIRACL

This subsection demonstrates the experimental analysis performed using the MIRACL library [4] to compute the cost of the cryptographic operations employed in the proposed protocol. MIRACL is a standard C $\setminus$ C++ based programming library used by cryptography researchers to compute the cost of cryptographic operations. The cryptographic symbols used in proposed protocol T_H, T_{AES}, T_{RSA}, T_{DH}, and T_{PM} are defined as the time required for one way hash function (256 b), (AES-128 b) encryption/ decryption, (RSA-2048 b) encryption/decryption, Diffie–Helmen (DH), and ECC multiplication (ECC-256 b), respectively.

We computed the cost of cryptographic operations on two different platforms: 1) a desktop which is used as server and 2) Raspberry Pi used as an D.

Platform-1. A desktop as the AS: A desktop was used as a server having the configuration: Intel(R) Core(TM) i7-3770 with 3.40 GHz clock, 8 GB RAM running Linux Ubuntu 18.04.6 LTS.

TABLE III
Results Obtained Through Experimental Analysis Using the MIRACL

Primitives	T_H	T_{PM}	T_{RSA}	T_{AES}	T_{DH}
Desktop (ms)	0.0032	0.495	4.69	.0036	1.0041
Raspberry PI 4 (ms)	0.0315	1.54	8.14	0.041	3.042

TABLE IV
Comparison of Computation Cost for Protocols

P	Device side	Server side	Total cryptographic operations	Total time (ms)
[3]	$5T_H + 3T_{PM}$	$3T_H + 3T_{PM}$	$8T_H + 6T_{PM}$	6.2
[19]	$T_{RSA} + 2T_{AES} + T_H$	$T_{RSA} + T_{AES} + T_H$	$2T_{RSA} + 3T_{AES} + 2T_H$	12.9
[14]	$T_{RSA} + T_{AES} + T_{DH}$	$T_{RSA} + T_{DH}$	$2T_{RSA} + T_{AES} + 2T_{DH}$	16.9
[15]	$T_{DH} + T_{RSA}$	$T_{DH} + T_{RSA}$	$2T_{DH} + 2T_{RSA}$	16.8
[11]	$3T_{AES} + 3T_H + T_{MIC}$	$T_{AES} + 4T_H + T_{MIC}$	$4T_{AES} + 7T_H + 2T_{MIC}$	0.25
[10]	$2T_{AES}$	$2T_{AES}$	$4T_{AES}$	0.09
[12]	$2T_{AES}$	$2T_{AES}$	$4T_{AES}$	0.09
[13]	$2T_H + 3T_{AES}$	$2T_H + 2T_{AES}$	$4T_H + 5T_{AES}$	0.2
[22]	$T_{PM} + 5T_H$	$2T_{PM} + 6T_H$	$3T_{PM} + 11T_H$	2.7
[23]	$3T_{PM} + T_H$	$3T_{PM}$	$6T_{PM} + T_H$	6.10
Ours	$3T_{AES} + T_H$	$3T_{AES} + T_H$	$6T_{AES} + 2T_H$	0.16

Platform-2. A Raspberry Pi as D: A Raspberry Pi was used as the D (UE) having the configuration: Model: 4B, CPU: ARM Cortex-A7, Cores: 4, and RAM: 8 GB.

Each cryptographic primitive was run 100 times to see how well it works. Based on the longest and shortest run times, we estimated the average run time in milliseconds, which is shown in Table III.

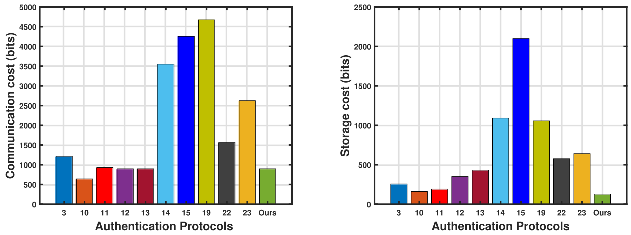
B. Computational Cost

In this section, we calculate and compare the cost of cryptographic operations in the proposed protocol and its counterparts. We use the computing time for cryptographic operations given in Table III, to evaluate the proposed protocol's performance. The computational cost required for the proposed protocol is $(6T_{AES} + 2T_H) \approx 0.16$ ms. The proposed protocol is least costly because it uses the combination of symmetric encryption and hash function. However, the combination of symmetric encryption and the hash function is less costly as compared to the combination of asymmetric encryption [4], [9]. Therefore, it is quite clear from the output of Table IV that the proposed protocol not only takes lesser cost compared to the protocols [3], [14], [15], [19], [22], [23] that use the asymmetric encryption but also the protocols [11], [13] that use the combination of symmetric and hash. However, the proposed protocol has slightly higher cost compared to symmetric encryption based protocols [10], [12] but provides more security features such as perfect forward secrecy, traceability, protection from ephemeral secret leakage, protection from privileged insider attack, and fast reconnect which is lacking in [10], [12].

C. Communication Cost

This section calculates and compares the number of bits sent in the channel for the proposed protocol and its counterparts. Based on earlier research (see [4]), we assess the cost of communication that is, identity, random number, each requiring 160 b. AES encryption/decryption, hashed output, public-key encryption/decryption using RSA, need 128 b, 256 b, 2048 b, respectively. The

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 5. Comparison of (a) communication and (b) storage cost of authentication protocols.

communication bits required for proposed protocol is $(CH, T_1, Z_{UID}), (RCH, T_2), (RES_1, RES_2, T_3), (CH_F, T_4)$ $\approx$ 896 b. The proposed protocol takes lesser communication cost because the exchanged messages are encrypted with AES. While AES with a 128-b key has the same level of security as RSA with a 2048-b key [4]. Therefore, the output of Fig. 5(a) clearly indicates that proposed protocol has lesser communication cost not only the protocols [3], [14], [15], [19], [23] that use the RSA or ECC but also the protocols [11], [22] that use the AES for exchanged messages. However, the proposed protocol has slightly higher cost compared to [10] and equivalent to [12], [13] but provides more security features such as perfect forward secrecy, traceability, protection from ephemeral secret leakage, protection from privileged insider attack, and fast reconnect not provided by [10], [12], [13].

D. Storage Cost

This section calculates the amount of memory required on the mobile device to hold the permanent protocol data. For the storage cost evaluation, we consider the cost of the cryptographic operations as indicated in [4]. In the proposed protocol, J is stored, which requires 128 b. The storage cost evaluation shows that our proposed protocol requires less storage cost compared to [3], [10]–[15], [19], [22], and [23] as shown in Fig. 5(b).

E. Energy Consumption

In this section, we compute the energy required for the proposed protocol and compare it with other related protocols. We compute the energy consumption as in [28]. The energy usage of a "StrongARM" CPU running at 133 MHz doing various tasks is summarized as the energy required for transmitting a bit is 0.00066 mj, energy required for AES symmetric enc/dec is 0.00217 mj, energy required for Hashed output is 0.000108 mj, and energy required for public key enc/dec RSA is 15.3 mj. The energy consumption needed for the proposed protocol is (896*.00066+6*.000217+2*.000108 = 0.59 mj). The proposed protocol uses the AES and hash function which requires lesser energy consumption as compared to RSA or ECC [28]. Therefore, we can infer from the output of energy consumption shown in Table V that the proposed protocol consumes lesser energy not only the protocols [3], [14], [15], [19], [22], [23] that use the RSA and ECC but also the protocols [11]–[13] that use the AES and hash function. However, the proposed protocol has slightly higher energy consumption as compared to [10] but provides more security features than [10].

TABLE V
COMPARISON OF ENERGY CONSUMPTION FOR PROTOCOLS

P	Exchanged message	Energy consumption (mj)
[3]	$(1216 \times 0.00066 + 6 \times 9.1 + 8 \times 0.000108)$	55.4
[19]	$(4672 \times 0.00066 + 2 \times 15.1 + 3 \times 0.00207 + 2 \times 0.000108)$	33.28
[14]	$(3552 \times 0.00066 + 2 \times 15.1 + 1 \times 0.00207 + 2 \times 5.3)$	43.1
[15]	$(4256 \times 0.00066 + 2 \times 15.1 + 2 \times 5.3)$	43.60
[11]	$(928 \times 0.00066 + 4 \times 0.00207 + 7 \times 0.000108 + 2 \times 0.00708)$	0.7
[10]	$(640 \times 0.00066 + 4 \times 0.00207)$	0.5
[12]	$(896 \times 0.00066 + 4 \times 0.00207)$	0.6
[13]	$(768 \times 0.00066 + 5 \times 0.00207 + 4 \times 0.000108)$	0.55
[22]	$(1568 \times 0.00066 + 3 \times 9.1 + 11 \times 0.000108)$	28.33
[23]	$(2624 \times 0.00066 + 6 \times 9.1 + 1 \times 0.000108)$	56.33
Ours	$(896 \times 0.00066 + 6 \times 0.000207 + 2 \times 0.000108)$	0.59

TABLE VI
COMPARISON OF SECURITY FEATURE AND FUNCTIONALITY ANALYSIS FOR AUTHENTICATION PROTOCOLS

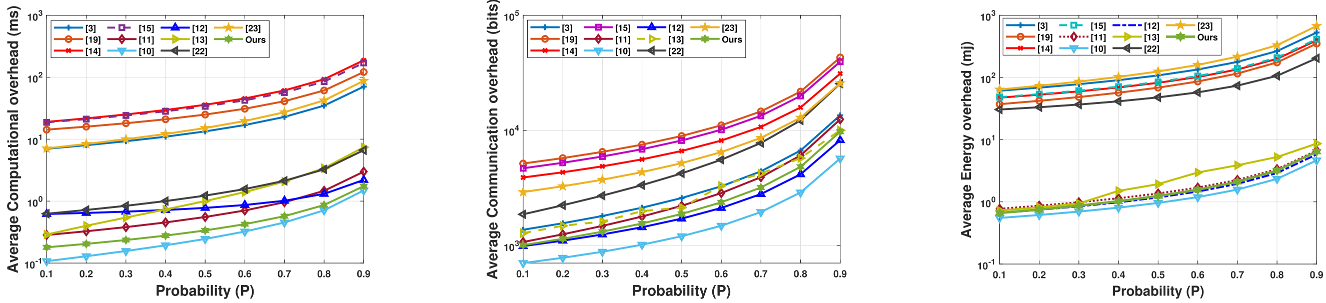
Protocol	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11
[3]	√	√	√	√	×	√	√	√	×	√	×
[19]	√	√	√	×	×	√	√	√	×	×	×
[14]	√	√	√	×	√	×	√	√	×	×	×
[15]	√	√	√	√	√	×	×	√	√	√	×
[11]	√	×	√	×	×	×	√	√	√	×	×
[10]	√	×	×	×	×	×	√	√	×	×	×
[12]	√	×	√	×	×	×	√	√	×	×	×
[13]	√	√	√	×	×	×	√	√	×	×	×
[22]	√	√	√	√	√	×	√	√	×	×	×
[23]	√	√	√	√	√	√	√	√	√	√	×
Ours	√	√	√	√	√	√	√	√	√	√	√

[0]NOTE: G1: Mutual authentication; G2: Perfect forward secrecy; G3: Identity protection; G4: Tractability; G5: Ephemeral secret leakage; G6: Privileged insider attack; G7: Desynchronization attack; G8: Replay attack; G9: Fast reconnect; G10: Illuminate secure channel requirements; G11: Prototype implementation; √- Secure against attack, ×-Insecure against attacks.

F. Security Analysis

This subsection compares the proposed protocol and its counterparts in terms of security features and functionality (mutual authentication, perfect forward secrecy, identity protection, traceability, ephemeral secret leakage, privileged insider attack, desynchronization attack, replay attack, etc.). Table VI summarizes the findings. For the symmetric key based authentication protocols [10]–[12], the security analysis in [13] shows that [10]–[12] are prone to various attacks shown in Table VI. Apart from that, the findings of [14] reveal that existing symmetric key based authentication protocols [10]–[13] do not offer the robust security that makes them inappropriate for practical implementation over WLAN. On the other hand, asymmetric key-based authentication protocols [3], [14], [15], [19], [22], [23] offer better security as compared to symmetric key based authentication protocols but are expensive in terms of computational, communication cost, and energy consumption. Table IV, Fig. 5(a), (b), and Table V show this. Therefore, asymmetric key based authentication protocols are unsuitable for ultralow cost Ds. The outcome of Table VI indicates that as compared to [3], [10]–[15], [19], [22], [23] the proposed protocol provides robust security not only against identified attacks but also offers additional security features such as protection from the privileged insider attack, ephemeral secret leakage, and traceability attack. Reason for this is that, after each successful authentication, the secrets, such as identity of the device, keys,

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c)

Fig. 6. Comparison overhead under unknown attacks. (a) Computational. (b) Communication. (c) Energy consumption of authentication protocols.

and random numbers used in the message exchange, are updated. Therefore, accessing long-term secrets or the device where all the secrets are stored will not provide the attacker any insight into the session key or earlier secret information transmitted between the communicating entities. It is worth noting that not only symmetric encryption protocols [10]–[13] are vulnerable to various attacks, but the asymmetric encryption protocols also [3], [14], [15], [19], [22], [23] fail to offer some security features as illustrated in Table VI. Therefore, we can infer that the proposed protocol has a clear edge over its counterparts in terms of security.

G. Performance Under Unknown Attack

This section looks into how well the proposed protocol and its competitors function in the face of unforeseen attacks. In the preceding subsections, we demonstrate that the proposed protocol is resilient against all identified known attacks. We anticipate there will be certain unidentified attacks that are difficult to predict when they occur. Therefore, we assessed the performance of the proposed protocol in the face of the unknown attack by calculating the likelihood impact of an unknown attack, similar to [29] and [30]. The performance under the unknown attack of the proposed protocol and its counterparts is computed using

$$C_A = \frac{C_S \times (1-P) + C_f \times P}{(1-P)}. \tag{1}$$

The terms C_A, C_S, C_f, and P used in (1) represents average cost, total cost of the successful authentication, cost when protocol halts [i.e., (2)] in the step k, and probability of attack in step k (i.e., independent of steps in which attack happens). The chance of an unknown attack occurring in step k is $1/L$, where L is the total number of signaling messages in a single protocol execution. The outcome of Fig. 6(a), (b), and (c) demonstrate that the proposed protocol outperforms its counterparts when an unknown attack happens. This is due to the fact that the proposed protocol has less computational, communication, and energy consumption. However, the proposed protocol has slightly greater overhead than [10], [12] since they require less communication, computational, and energy. While the security study of [13] reveals that [10] and [12] lack the prominent security characteristics, making them inappropriate for use in real-time applications. As a result of the findings, we may conclude that the proposed protocol performs better not in presence of known attacks, but also when unknown attack occurs

$$C_f = \sum_{k=1}^{L} C_k \times \frac{1}{L}. \tag{2}$$

TABLE VII
PARAMETERS USED IN NETWORK SIMULATION

Parameter	Value
Operating System	Ubuntu 18.04.6 LTS
Simulation Time	1800s
Network coverage area	100 m × 100 m
Number of Access point & Authentication server	1& 1
Number of devices in scenario 1, 2, and 3	10 , 20, and 30
Routing protocol	OLSR
MAC protocol	IEEE 802.11
Distance between the devices and Access point	20 m to 50 m

H. Practical Simulation Using NS3

This section demonstrates the results of the experimental analysis carried out using the network simulator tool NS3 [4]. We measure two network performance parameters (i.e., throughput and packet delivery ratio), to demonstrate the applicability of the authentication phase. Table VII depicts the parameters used in the simulation of authentication protocols.

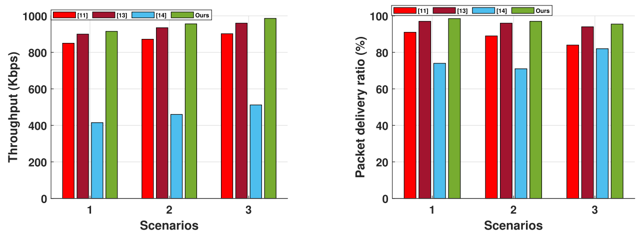
The authentication phase takes place between the D, AP, and AS which contains four messages: (CH, Z_{UID}, T_1), (RCH, T_2), (RES_1, RES_2, T_3), and (CH_F, T_4), of size 288, 160, 288, and 160 b long, respectively. In order to show the efficiency, we also simulate the existing authentication protocols [11], [13], and [14].

1) Impact on Throughput: The throughput is computed based on the number of bits transmitted per unit using

$$Throughput = \sum \frac{P_{M_i} \times N_i}{t} \tag{3}$$

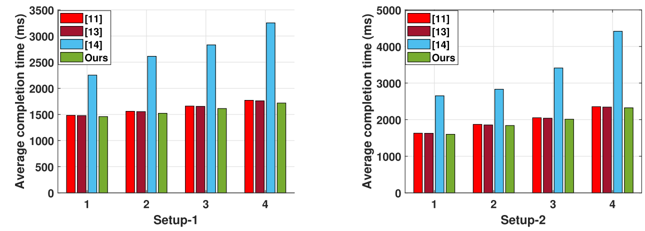
whereas P_{M_i} denotes the number of received packets of i type, N_i denotes the length of the packet of type i, and t denotes the total time. The outcome of the throughput for scenarios 1, 2, and 3 are 915, 956, and 986 kbps, respectively. The comparison result shown in Fig. 7(a) indicates that the proposed protocol has the highest throughput compared to [11], [13], and [14]. This is due

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 7. Comparison of network performance. (a) Throughput. (b) PDR of protocols.



(a) (b)

Fig. 9. Comparison of completion time. (a) Setup-1. (b) Setup-2 of authentication protocols.

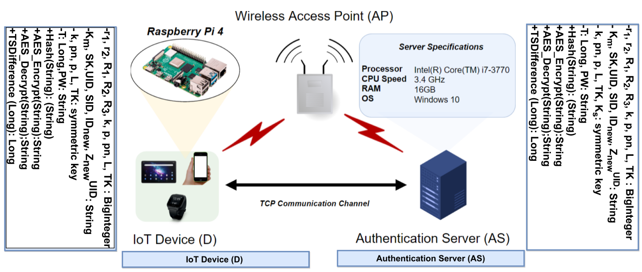


Fig. 8. Prototype test bed implementation of the protocol.

to the fact that the message size and cryptographic operations of the proposed protocol are less than [11], [13], and [14]. Apart from that, it can be observed from the throughput results that when the number of nodes exceeds, throughput exceeds as well. The main reason behind this is that the number of authentications between the server and the devices has increased.

2) Packet Delivery Ratio (PDR): The PDR is the parameter used to track network congestion using

$$PDR = \frac{R_p}{S_p} \tag{4}$$

where R_p represents the number of the received packet and S_p represents the number of sent packets by the sender. The PDRs for scenarios 1, 2, and 3 are 98.3%, 97%, and 95%, respectively. The comparison result shown in Fig. 7(b) shows that the proposed protocol has the highest PDR as compared to [11], [13], and [14]. The outcome of the PDR shows that as the number of nodes increases, the PDR decreases due to congestion.

VIII. Prototype Implementation

To determine the feasibility of the proposed protocol, a prototype test bed was created for real-time implementation. We set up a Transmission Control Protocol (TCP)-based communication channel between the communicating entities, as shown in Fig. 8. A wireless AP was used to route the connection. The channel was built on a Java platform and used a socket-based inter-process communication (IPC) method. To complete the message flow, two primary classes were formalized, and the IPC scenario was established using the sequence shown in Fig. 1. It was assumed that preshared symmetric key would be distributed to all parties involved in the communication. In Fig. 8, the derived parameters and procedures are listed. We tested our protocol for two different setup as explained below. The goal was to compare and contrast the performance of a resource-limited D with those of a resource-rich device.

Setup-1. A laptop as IoT device and a desktop as the server: We use a laptop as the D baring the configurations: Intel(R) Core(TM) i7-3770 with 3.40 GHz clock, 8 GB RAM running on Windows 10, and the desktop as the server which specifications are indicated in Fig. 8.

Setup-2. A Raspberry Pi as the D and desktop as the server: A Raspberry Pi (Model : 4B, CPU: ARM Cortex-A7, Cores : 4, and RAM : 8 GB) was deployed as the D. The same wireless AP was deployed for both scenarios. This scenario reflects a real resource-constrained IoT environment.

To demonstrate the efficacy in terms of average completion time, we implement the proposed protocol and more contemporary protocols [11], [13], [14] in a test-bed environment. These protocols are implemented for different key sizes {i.e. [Size combination (SC1)=(AES-128, Hash-160)], [SC2=(AES-128, Hash-256)], [SC3=(AES-256, Hash-160)], [SC4=(AES-256, Hash-256)]}. While, we used RSA encryption with a 2048-b key size for [14]. The comparison outcome of Fig. 9(a) and (b) shows that proposed protocol has less average completion time compared to [11], [13], and [14]. The rationale behind this is that the proposed protocol takes less communication and computational cost compared to [11], [13], and [14]. Apart from that, [14] is based on asymmetric encryption, that is why it requires very high completion time while [11] and [13] are based on symmetric that requires the high completion time compared to the proposed protocol and are vulnerable to several types of attacks shown in Table VI. Therefore, we can conclude from the implementation results that the proposed protocol performs better compared to asymmetric encryption based protocol [14] as well as symmetric encryption based protocols [11], [13].

IX. Conclusion

In this article, we presented a mutual authentication protocol for the WLAN communication. To achieve a balance between security criteria and at the same time being lightweight, the proposed protocol employs a combination of symmetric encryption and hash functions. We provided an informal and formal

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.

(using BAN logic and Scyther tool) analysis of the proposed protocol, which demonstrates that it is secure against the attacks. Moreover, we evaluated the proposed protocol's performance in terms of computational, communication, storage, and energy consumption, demonstrating that the proposed protocol is less expensive than the existing protocols. In addition to this, we computed the overhead under unknown attacks indicating that proposed protocol takes less overhead compared to its counterparts. Furthermore, we showed the practical simulation of the proposed protocol using the NS3 tool to confirm it applicability in practical scenarios. A prototype implementation has been done to show that it can be easily implemented in real-time applications. As a result, we may conclude that the proposed protocol is safe, efficient, suitable for IoT applications, and provides the balance between the security and cost.

References

[1] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3680–3689, Jun. 2019.

[2] M. Liyanage, A. Braeken, P. Kumar, and M. Ylianttila, *IoT Security: Advances in Authentication*. Wiley, 2020.

[3] L. D. Tsobdjou, S. Pierre, and A. Quintero, "A new mutual authentication and key agreement protocol for mobile client–server environment," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 1275–1286, Jun. 2021.

[4] J. Lee, G. Kim, A. K. Das, and Y. Park, "Secure and efficient honey list-based authentication protocol for vehicular ad hoc networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2412–2425, Jul.–Sep. 2021.

[5] T. Bolton, T. Dargahi, S. Belguith, M. S. Al-Rakhami, and A. H. Sodhro, "On the security and privacy challenges of virtual assistants," *Sensors*, vol. 21, no. 7, 2021, Art. no. 2312.

[6] H. A. Omar, K. Abboud, N. Cheng, K. R. Malekshan, A. T. Gamage, and W. Zhuang, "A survey on high efficiency wireless local area networks: Next generation WiFi," *IEEE Commun. Surv. Tut.*, vol. 18, no. 4, pp. 2315–2344, Fourthquarter 2016.

[7] C. Nykvist, M. Larsson, A. H. Sodhro, and A. Gurtov, "A lightweight portable intrusion detection communication system for auditing applications," *Int. J. Commun. Syst.*, vol. 33, no. 7, 2020, Art. no. e4327.

[8] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible authentication protocol (EAP)," Tech. Rep., 2004.

[9] C.-I. Fan, Y.-H. Lin, and R.-H. Hsu, "Complete EAP method: User efficient and forward secure authentication protocol for IEEE 802.11 wireless LANs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 672–680, Apr. 2013.

[10] A. Pandey, P. K. Pant, and R. Tripathi, "A system and method for authentication in wireless local area networks (WLANs)," *Proc. Nat. Acad. Sci., India Sect. A: Phys. Sci.*, vol. 86, no. 2, pp. 149–156, 2016.

[11] A. Kumar and H. Om, "A secure, efficient and lightweight user authentication scheme for wireless LAN," in *Proc. Int. Conf. Emerg. Trends Eng., Technol. Sci.*, 2016, pp. 1–9.

[12] B. Dey, S. Vishnu, and O. S. Swarnkar, "An efficient dynamic key based EAP authentication framework for future IEEE 802.1 x wireless LANs," in *Proc. 2nd Int. Conf. Digit. Signal Process.*, 2018, pp. 125–131.

[13] A. K. Yadav, M. Misra, M. Liyanage, and G. Varshney, "Secure and user efficient EAP-based authentication protocol for IEEE 802.11 wireless LANs," in *Proc. IEEE 17th Int. Conf. Mobile Ad Hoc Sensor Syst.*, 2020, pp. 576–584.

[14] P. Kumar and D. Kumar, "A secure n-secret based client authentication protocol for 802.11 WLANs," *Telecommun. Syst.*, vol. 75, no. 3, pp. 259–271, 2020.

[15] K. Moriarty and S. Farrell, "RFC 8996 Deprecating TLS 1.0 and TLS 1.1," *Internet Eng. Task Force, (IETF)*, 2021.

[16] D. Simon, B. Aboba, and R. Hurst, "The EAP-TLS authentication protocol," Tech. Rep., 2008.

[17] P. Funk and S. Blake-Wilson, "Extensible authentication protocol tunneled transport layer security authenticated protocol version 0 (eap-ttlsv0)," Tech. Rep., RFC 5281, Aug. 2008.

[18] N. Cam-Winget, D. McGrew, J. Salowey, and H. Zhou, "The flexible authentication via secure tunneling extensible authentication protocol method (EAP-fast)," Tech. Rep., 2007.

[19] B. Shojaie, I. Saberi, and M. Salleh, "Enhancing EAP-TLS authentication protocol for IEEE 802.11 i," *Wireless Netw.*, vol. 23, no. 5, pp. 1491–1508, 2017.

[20] O. Cheikhrouhou, M. Laurent, A. B. Abdallah, and M. B. Jemaa, "An EAP-EHash authentication method adapted to resource constrained terminals," *Annals Telecommun.-Annales des Télécommunications*, vol. 65, no. 5, pp. 271–284, 2010.

[21] Y. E. H. El Idrissi, N. Zahid, and M. Jedra, "A new EAP authentication method for IEEE 802.11 wireless," *IJCSNS*, vol. 11, no. 6, pp. 1–11, 2011.

[22] B. Baruah and S. Dhal, "An efficient authentication scheme for secure communication between industrial IoT devices," in *Proc. IEEE 11th Int. Conf. Comput. Commun. Netw. Technol.*, 2020, pp. 1–7.

[23] Y. Zhang, R. H. Deng, E. Bertino, and D. Zheng, "Robust and universal seamless handover authentication in 5G hetnets," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 858–874, Mar./Apr. 2021.

[24] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[25] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, Berlin, Heidelberg: Springer, 2002, pp. 337–351.

[26] M. Abadi and M. R. Tuttle, "A logic of authentication," in *Proc. ACM Trans. Comput. Syst.* Citeseer, 1990, pp. 18–36.

[27] C. J. F. Cremers, *Scyther: Semantics and Verification of Security Protocols*. Eindhoven, Netherlands: Eindhoven Univ. Technol., 2006.

[28] Z. Xu, X. Li, J. Xu, W. Liang, and K.-K. R. Choo, "A secure and computationally efficient authentication and key agreement scheme for internet of vehicles," *Comput. Elect. Eng.*, vol. 95, 2021, Art. no. 107409.

[29] J. Cao, M. Ma, Y. Fu, H. Li, and Y. Zhang, "CPPHA: Capability-based privacy-protection handover authentication mechanism for SDN-based 5G hetnets," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1182–1195, May/Jun. 2021.

[30] R. Ma, J. Cao, D. Feng, H. Li, and S. He, "FTGPHA: Fixed-trajectory group pre-handover authentication mechanism for mobile relays in 5G high-speed rail networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2126–2140, Feb. 2020.

Awaneesh Kumar Yadav (Student Member, IEEE) received the M.Tech. degree in computer science from the Department of Computer Science and Engineering, National Institute of Technology Rourkela, Rourkela, India, in 2019. He is currently working toward the Ph.D. degree in computer science with the Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, India.

His research interests include network security, 5G security, IoT, and cloud security.

Manoj Misra (Member, IEEE) received the Ph.D. degree in computer science from the University of New Castle upon Tyne, Newcastle upon Tyne, U.K., in 1997.

He was previously an Engineer with CMC Limited Noida, an Assistant Engineer with Hindustan Aeronautic Limited at Kanpur India, and an Assistant Professor with HBTI Kanpur. He is currently a Professor with the Department of Computer Science and Engineering, IIT Roorkee, Roorkee, India. His research interests include distributed computing, performance evaluation, computer networks, network security, and cyber frauds.

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.

Pradumn Kumar Pandey (Member, IEEE) received the B.Tech. and the Ph.D. degrees in computer science and engineering from IIT Jodhpur, Jheepasani, India, in 2012 and 2018, respectively.

From May to September 2018, he was an Institute Post-Doctoral Fellow with the Department of Computer Science and Engineering, IIT Kharagpur, Kharagpur, India. From October 2018 to October 2019, he worked as a DST INSPIRE Faculty Member with the Department of Computer Science and Engineering, IIT Roorkee, Roorkee, India, where he has been working as an Assistant Professor since November 2019. His research interests include modeling of complex networks, information diffusion on real networks, social security on online social networks, and network representation learning.

Madhusanka Liyanage (Senior Member, IEEE) received the D.Sc (Tech) degree in communication engineering from the University of Oulu, Oulu, Finland, in 2016.

From 2011 to 2012, he worked as a Research Scientist with the I3S Laboratory and Inria, Sophia Antipolis, France. He is currently an Assistant Professor/Ad Astra Fellow and the Director of Graduate Research with the School of Computer Science, University College Dublin, Dublin, Ireland. He is also acting as an adjunct Processor with the Center for Wireless Communications, University of Oulu and Department of Electrical and Information Engineering, University of Ruhuna, Matara, Sri Lanka. He is also an Expert Consultant with the European Union Agency for Cybersecurity (ENISA). His research interests are 5G/6G, SDN, IoT, blockchain, MEC, mobile, and virtual network security.

Dr. Liyanage was a recipient of the prestigious Marie Skłodowska–Curie Actions Individual Fellowship during 2018–2020. In 2020, he received the “2020 IEEE ComSoc Outstanding Young Researcher” award by IEEE ComSoc EMEA. He was also the recipient of the Irish Research Council (IRC) Research Ally Prize as part of the IRC Researcher of the Year 2021 awards for the positive impact he have made as a supervisor. In 2021, Liyanage was elevated as Funded Investigator of Science Foundation Ireland CONNECT Research Centre, Dublin, Ireland. He was ranked among the World’s Top 2% Scientists (2020) in the List prepared by Elsevier BV, Stanford University, Stanford, CA, USA. Moreover, he is an expert reviewer at different funding agencies in France, Qatar, UAE, Sri Lanka, and Kazakhstan.

Authorized licensed use limited to: Indian Institute Of Technology (IIT) Mandi. Downloaded on September 19,2025 at 11:38:43 UTC from IEEE Xplore. Restrictions apply.