LEMAP: A Lightweight EAP based Mutual Authentication Protocol for IEEE 802.11 WLAN

Awaneesh Kumar Yadav*, Manoj Misra*, Pradumn Kumar Pandey *, Kuljeet Kaur†, Sahil Garg‡, Madhusanka Liyanage‡

*Dept. of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India.
†Electrical Engineering Department, École de technologie supérieure (ETS), Montreal, QC H3C 1K3, Canada.
‡School of Computer Science, University College Dublin, Ireland and CWC, University of Oulu, Finland.

Email: [*akumaryadav, *manoj.misra, *pradumn.pandey]@cs.iitr.ac.in, [†kuljeet.kaur, ‡sahil.garg]@ieee.org,
†madhusanka@ucd.ie.

Abstract—The growing usage of wireless devices has significantly increased the need for Wireless Local Area Network (WLAN) during the past two decades. However, security (most notably authentication) remains a major roadblock to WLAN adoption. Several authentication protocols exist for verifying a supplicant’s identity who attempts to connect his wireless device to an access point (AP) of an organization’s WLAN. Many of these protocols use the Extensible Authentication Protocol (EAP) framework. These protocols are either vulnerable to attacks such as violation of perfect forward secrecy, replay attack, synchronization attack, privileged insider attack, and identity theft or require high computational and communication costs. In this paper, a lightweight EAP-based authentication protocol for IEEE 802.11 WLAN is proposed that not only addresses the security issues in the existing WLAN authentication protocols but is also cost-effective. The security of the proposed protocol is verified using BAN logic and the Scyther tool. Our analysis shows that none of the existing authentication mechanisms provide a balance between security and the cost. Therefore, we propose a lightweight EAP-based authentication protocol that uses a combination of symmetric encryption and secure hash function to achieve this balance.


I. INTRODUCTION

Wireless Local Area Network (WLAN) has achieved significant popularity in the last two decades. The rationale behind the high demand for WLAN is the development of lightweight wireless devices: smartphones, tablets, printers, bluetooth mics, and social applications such as Gmail, Twitter, Linked-In, Facebook, etc [1]. WLAN communication uses a public channel where any unauthorized user within the radio range of the wireless router or AP may try to connect it. So, to solve this issue, a robust authentication mechanism is required that restricts unauthorized access to the network [2] [3]. Authentication is a mechanism through which communicating parties examine the legitimacy of each other and restrict unauthorized access to the network [4] [5].

IEEE 802.11i defines the WLAN security architecture, which outlines the flexible key hierarchy and key exchange between the Supplicant (STA) (i.e., refers to the software that is installed on the client’s device) and Authentication Server (S) (i.e., functions as a backend server that authenticates and provides the authentication services to the supplicant). IEEE 802.11i makes use of IEEE 802.1x, which establishes a safe and reliable authentication framework for establishing a secure connection between the STA and S. The EAP framework is used in the IEEE 802.1x design for dependable base and message exchange [6]. Several authentication mechanisms have been proposed for IEEE 802.11 WLANs. Most of them use the EAP framework because it is flexible and easy to use. The detailed description of the EAP framework and the mandatory security requirements for EAP framework-based authentication protocols are given in RFC-3748 [7], and RFC-4017 [8]. However, it is a well-established fact that additional requirements like protection from privileged insider attack and lightweight computation are also needed in designing robust solutions using EAP.

Several authentication protocols exist in the literature based on symmetric and/or asymmetric encryption. Most authentication protocols based on symmetric encryption are lightweight but prone to various attacks such as privileged insider attacks, violation of perfect forward secrecy, synchronization attack, identity theft. In comparison, asymmetric encryption-based authentication protocols offer better security but require high costs and are susceptible to Man-In-the-Middle (MITM) if not implemented correctly [9]. Hence, we can infer that the existing protocols do not provide the fragile balance between the security and the cost. This motivated us to design an authentication mechanism that addresses the issues in existing protocols. We present an authentication mechanism that addresses the security issues in state-of-the-art authentication solutions and offers additional security features like privileged insider attack protection and lightweight computation.

A. Contributions

- Our analysis shows that none of the existing authentication mechanisms provide a balance between security and cost. Therefore, we propose a lightweight EAP-based authentication protocol that uses a combination of symmetric encryption and secure hash function to achieve this balance.
The formal validation of the proposed protocol is carried out through BAN logic and the Scyther tool. The validation outcome shows that the proposed protocol addresses all the identified security issues in the existing authentication solutions. Moreover, the proposed protocol provides extra security, such as protection from privileged insider attack.

Extensive analysis of the proposed protocol is performed in terms of computation and communication cost. The outcome of the analysis indicates that the proposed protocol is lightweight compared to state-of-the-art solutions.

B. Organization

In Section II, we summarise the existing literature on authentication in WLAN, including the research gaps. Section III presents the proposed protocol for mutual authentication. Further, formal security analysis of the proposed protocol is discussed in Section IV. The performance of the proposed protocols is demonstrated in Section V followed by the conclusion in Section VI.

II. RELATED WORK

This section covers the most up-to-date WLAN authentication standards. There are three types of EAP-based authentication methods currently available: a) strong password-based authentication, b) certificate-based authentication and c) hybrid that is a combination of strong password and certificate-based authentication.

In strong password-based authentication protocols, supplicant (STA) and authentication server (S) assure each other that they knew a secret without transmitting it. EAP based authentication protocols [9]–[16] belonging to this category use symmetric encryption to encrypt and decrypt the exchanged messages. These protocols are lightweight but prone to various attacks like replay attack, identity theft, privileged insider attack, and violation of perfect forward secrecy. Therefore, existing strong password-based authentication protocols fail to provide a balance between security and cost [17].

Certificate-based EAP authentication methods [17] [18] [19] use asymmetric encryption to encrypt and decrypt the exchanged messages. The analysis shows that they provide better security as compared to strong password-based EAP authentication protocols but require high cost and delay. This is because they use a combination of RSA and Diffie-Hellman, which is costly compared to using the combination of Advanced Encryption Standard (AES) and hash function [11].

There are various authentication protocols [20] [21] that use the combination of the certificate and strong password to achieve a balance between cost and security. In this type of protocols, S uses the certificate to prove its legitimacy, while STA uses the strong password-based approach to prove its authenticity. It is observed that they provide better security as compared to the strong password-based approaches and require lesser cost as compared to certificate-based authentication protocol but are susceptible to MITM attack if not implemented correctly [9].

A. Research Gaps

We found the following research gaps after the analysis of existing EAP based authentication schemes:

1) Lack of identity protection: The identities of the STA and S must always be exchanged in masked form, according to identity protection. The majority of the EAP based authentication protocols [9], [10], [12], [13], [18] do not provide the identity protection.

2) Lack of protection from privileged insider attack: It requires that STA must keep secret credentials in disguised form in the database so that no insider may pry into the information. None of the existing schemes [9]–[16] based on pre-shared key provide the protection from the privileged insider attack.

3) Perfect forward secrecy: Even if long-term credentials are compromised, obtaining the previous session key should not be possible. Majority of the authentication protocols [9], [10], [12], [13], [18] fail to preserve the perfect forward secrecy.

4) Cost: It is seen that strong password-based EAP protocols fail to address the security requirements like replay attack protection, privileged insider attack protection, identity protection, and perfect forward secrecy, while the certificate-based authentication protocols require high cost and delay.

III. PROPOSED PROTOCOL

This section discusses our proposed authentication protocol. The authentication process takes place between the three entities, namely: a supplicant (STA), an access point (AP), and an authentication server (S). In the WLAN communication, we assume that the connection between the AP and S is secure while the connection between STA and AP is considered insecure [11]. We also assume that the clocks on the STA and S are synchronized. The proposed protocol involves two phases: i) registration phase and ii) authentication phase. The proposed protocol uses the combination of symmetric encryption and a Secure Hash Algorithm (SHA) that reduces the cost of the authentication compared to the asymmetric encryption-based authentication protocol. Table I represents the notations used in our paper.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Meanings</th>
</tr>
</thead>
<tbody>
<tr>
<td>STA , $M_{id}$</td>
<td>Supplicant and masked identity of supplicant</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>S</td>
<td>Authentication-server</td>
</tr>
<tr>
<td>$K_s$, $R_s^l$</td>
<td>short-term keys</td>
</tr>
<tr>
<td>$K_{AS}$</td>
<td>Private key of Server</td>
</tr>
<tr>
<td>$\oplus$</td>
<td>XOR, Concatenation</td>
</tr>
<tr>
<td>$S_{id}$</td>
<td>Supplicant's identity, Server’s identity</td>
</tr>
<tr>
<td>PW</td>
<td>Password</td>
</tr>
<tr>
<td>H</td>
<td>One-way hash function</td>
</tr>
<tr>
<td>$E_k$, $D_k$</td>
<td>Encryption and Decryption with symmetric key k</td>
</tr>
<tr>
<td>$T_1$, $T_2$, $T_3$, $T_4$</td>
<td>Time stamps</td>
</tr>
<tr>
<td>$r_1$, $r_2$, $r_3$, $r_4$, $R_s$</td>
<td>random numbers</td>
</tr>
<tr>
<td>$SK$</td>
<td>session key for mutual- authentication</td>
</tr>
</tbody>
</table>
A. Threat Model

In our proposed work, we adopt the Dolev-Yao [22] and Canetti-Krawczyk [23] threat models.
1) Attacker (A) can launch both active and passive attacks.
2) A has complete access of the communication network and can access, modify, and delete transmitted messages.
3) A can also inject fake messages on the network and can impersonate as a legitimate entity while communicating with S.

B. Registration Phase

The registration phase is carried out using a secure channel in which STA and S exchange their secret credentials. The registration phase for the proposed protocol is shown in Fig. 1.

**Step-1:** STA chooses and sends the identity STAid and password PW to S.

**Step-2:** After receiving the STAid and PW, S selects two random numbers (Rs, Rp). It computes the $K'_AS = (KAS ⊕ Rp)$ and $M_{id} = H(STA_{id} ⊕ Rs)$ (given in Eqs (1) - (4)) and sends $< K_s, P_1, M_{id}, S_{id} >$ to STA. Afterwords S stores the $< M_{id}, PW, STA_{id}, Rp >$ into its database. However, private key $K_{AS}$, and database having STA’s data are stored at different places as in [24].

\[
K'_AS = (KAS ⊕ Rp) \tag{1} \\
M_{id} = H(STA_{id} ⊕ Rs) \tag{2} \\
P_1 = STA_{id} ⊕ Rs ⊕ K'_AS \tag{3} \\
K_s = H(Rs) \tag{4}
\]

**Step-3:** After receiving the credentials $< P_1, K_s, M_{id}, S_{id} >$

\[
Q = E_{PW}(M_{id} || S_{id} || P_1 || K_s) \tag{5}
\]

C. Authentication Phase

The authentication process occurs between the STA and S in which they verify each other’s legitimacy using the pre-shared secrets with the help of AP. Fig. 2 shows the complete mutual authentication process. Details of the steps shown in Fig. 2 are given below:

**Step-1:** To access the network, STA decrypts the stored credential ($D_{PW}(Q)$), gets the current timestamp $T_1$, and selects a random number $r_1$. Then it computes $E_1$ (given in Eq (6)) and forwards $< E_1, T_1, M_{id}, P_1 >$ to AP.

\[
E_1 = E_{K_s}(T_1 || r_1 || PW || M_{id} || P_1) \tag{6}
\]

**Step-2:** AP forwards $< E_1, T_1, M_{id}, P_1 >$ to S.

<table>
<thead>
<tr>
<th>Supplicant (STA)</th>
<th>Access Point (AP)</th>
<th>Authentication Server(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&lt; P_1, K_s, M_{id}, S_{id} &gt;$</td>
<td>$&lt; E_1, T_1, M_{id}, P_1 &gt;$</td>
<td>$&lt; E_1, T_1, M_{id}, P_1 &gt;$</td>
</tr>
</tbody>
</table>

Fig. 2: Proposed Protocol For Mutual Authentication

**Step-3:** After receiving the message $< E_1, T_1, M_{id}, P_1 >$, S gets the current time $T_2$ and selects two random numbers ($R_L, r_2$). It then checks the freshness condition (given in Eq (7)), if it meets then S extracts the credentials based on the received $M_{id}$ and computes the $K'_{AS} = (KAS ⊕ Rp)$, $R_s = (P_1 ⊕ STA_{id} ⊕ K'_{AS})$ and $K_s = H(Rs)$. After getting the credentials, it decrypts the message and compares the credentials ($T_1 = T'_1$, $M_{id} = M'_{id}$, $PW = PW'$). If they match then it computes $R'_L$ (given in Eq (9)), $E_2$ (given in Eq (10), $P_2$ (given in Eq (8)) and forwards $< E_2, T_2 >$ to the AP.

\[
E_2 = E_{K_s}(T_2 || r_1 || PW || M_{id} || P_1) \tag{9}
\]

\[
P_2 = STA_{id} ⊕ K'_{AS} \oplus R_L \tag{8}
\]

\[
R'_L = H(R_L) \tag{9}
\]

\[
E_2 = E_{K_s}(T_2 || r_1 || R'_L || S_{id} || P_2) \tag{10}
\]
**Step-4:** After receiving the message \(< E_2, T_2 \rangle \) from \( S \), \( AP \) passes it to \( STA \).

**Step-5:** On receiving the message \(< E_2, T_2 \rangle \) from \( AP \), \( STA \) checks the freshness condition (given in Eq (11)). If it is satisfied, \( STA \) decrypts \( D_{K_s}(E_2) \) and compares the credentials \((T_2 = T_2^*, r_1 = r_1^*, S_id = S_id^*)\). If the credentials match, then it computes \( E_3 \) (given in Eq (12)) and forwards the \(< E_3, T_3 \rangle \) to \( AP \).

\[ T_3 - T_2 < T \text{ (11)} \]
\[ E_3 = E_{K_s}(T_3 \parallel r_3 \parallel r_2) \text{ (12)} \]

**Step-6:** \( AP \) forwards the message \(< E_3, T_3 \rangle \) to the \( STA \).

**Step-7:** Upon receiving the message \(< E_3, T_3 \rangle \) from the \( AP \), \( S \) gets the current timestamp \( T_4 \) and checks the freshness condition (given in Eq (13)) if it is satisfied then \( S \) decrypts the \( D_{K_s}(E_3) \) and compares the credentials \((T_4 = T_4^*, r_2 = r_2^*)\). If credentials match then it computes the session key \( SK \) (given in (14)) and forwards the successful acknowledge to the \( AP \).

\[ T_4 - T_3 < T \text{ (13)} \]
\[ SK = H(r_1 \oplus r_2) \text{ (14)} \]

**Step-8:** Upon receiving the acknowledgement from the \( S \), \( AP \) forwards it to the \( STA \).

**Step-9:** After receiving the acknowledgment, \( STA \) computes the session key (given in Eq (15)) and updates \( K_s \leftarrow R_L^1 \), \( P_1 \leftarrow P_2 \) and starts conversation using the session key.

\[ SK = H(r_1 \oplus r_2) \text{ (15)} \]

### IV. FORMAL SECURITY ANALYSIS

This section presents the formal proof of the proposed protocol using the Burrows, Abadi, and Needham logic (BAN) Logic [25] and Sctyr tool [26] same as in [2] [4] [27].

#### A. Security Verification using BAN logic

\( STA \) and \( S \) are the principals involved in communication, \( G \) denotes the statement and \( K \) is the shared key between the principals \( STA \) and \( S \). Table II shows the BAN logic notation and formulas.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( STA \vdash G )</td>
<td>( STA ) believes the statement ( G )</td>
</tr>
<tr>
<td>( STA \vdash \neg G )</td>
<td>( STA ) receives the statement ( G )</td>
</tr>
<tr>
<td>( STA \Rightarrow G )</td>
<td>( STA ) once sent the statement ( G )</td>
</tr>
<tr>
<td>( K )</td>
<td>( STA ) has full control over the statement ( G )</td>
</tr>
<tr>
<td>( #(G) )</td>
<td>( G ) is fresh</td>
</tr>
<tr>
<td>( {G}_K )</td>
<td>( G ) is combined with ( K )</td>
</tr>
<tr>
<td>( {G}_K )</td>
<td>( G ) is encrypted with ( K )</td>
</tr>
<tr>
<td>( STA \vdash STA \Rightarrow K )</td>
<td>( STA ) believes that ( K ) is shared between ( STA ) and ( S ).</td>
</tr>
<tr>
<td>( \text{TVR} )</td>
<td>( \text{TVR} )</td>
</tr>
<tr>
<td>( \text{MMR} )</td>
<td>( \text{MMR} )</td>
</tr>
<tr>
<td>( \text{JR} )</td>
<td>( \text{JR} )</td>
</tr>
<tr>
<td>( \text{M} )</td>
<td>( \text{M} )</td>
</tr>
<tr>
<td>( \text{I} )</td>
<td>( \text{I} )</td>
</tr>
</tbody>
</table>

| Table II: BAN notations and formulas        |

1) Initial assumptions of the proposed protocol

\[ J_1 : STA \vdash S \Rightarrow STA \vdash K \rightarrow S \]
\[ J_2 : S \vdash \#(T_2) \]
\[ J_3 : STA \vdash \#(T_3) \]
\[ J_4 : S \vdash STA \vdash K \rightarrow S \]
\[ J_5 : S \vdash \#(T_1) \]
\[ J_6 : S \vdash \#(T_3) \]
\[ J_7 : S \vdash STA \Rightarrow STA \vdash K \]

2) Security goals of the proposed protocol

**Goal-1:** \( S \vdash STA \vdash S \vdash (SK) \rightarrow S \)

**Goal-2:** \( S \vdash (SK) \rightarrow S \)

**Goal-3:** \( S \vdash STA \vdash S \vdash (SK) \rightarrow S \)

**Goal-4:** \( S \vdash (SK) \rightarrow S \)

3) Idealized form of the proposed protocol

\[ E_1 : STA \\rightarrow \vdash S : (T_1 \parallel PW \parallel M_id \parallel r_1 \parallel P_1)_K \]
\[ E_2 : S \\rightarrow \vdash STA : (T_2 \parallel r_2 \parallel S_id \parallel r_1 \parallel P_2 \parallel R_L)_K \]
\[ E_3 : STA \vdash \rightarrow S : (T_3 \parallel r_2 \parallel r_3)_K \]

4) Proof and derivation of security goals:

**Step-1:** We apply MMR rule and assumption \( J_4 \) on \( E_1 \).

\[ I_1 : S \vdash STA \vdash \neg E_1 \]

**Step-2:** By applying the TVR rule and assumption \( J_5 \) on \( I_1 \), we conclude

\[ I_2 : S \vdash STA \vdash \neg E_2 \]

**Step-3:** On applying MMR rule on \( E_3 \) with assumption \( J_6 \) we conclude

\[ I_3 : S \vdash STA \vdash \neg E_3 \]

**Step-4:** By applying the TVR rule on \( E_3 \) with \( J_6 \) and \( I_3 \).

\[ I_4 : S \vdash STA \vdash \neg (r_2 \oplus r_3) \]

**Step-5:** From the \( I_2 \), \( I_4 \) and as \( SK = H(r_1 \oplus r_2) \), we can conclude

\[ I_5 : S \vdash STA \vdash (SK) \rightarrow S \quad \text{Goal-1} \]

**Step-6:** We apply JR and \( J_7 \) on \( I_5 \).

\[ I_6 : S \vdash STA \vdash (SK) \rightarrow S \quad \text{Goal-2} \]

**Step-7:** We apply MMR rule and assumption \( J_1 \) on \( E_2 \).

\[ I_7 : STA \vdash S \vdash \neg E_2 \]

**Step-8:** By applying the TVR on \( E_2 \) based on \( J_2 \) and \( I_7 \).

\[ I_8 : STA \vdash S \vdash \neg (r_1, r_2, S_id, R_L, P_2) \]

**Step-9:** From the \( I_8 \) and as \( SK = H(r_1 \oplus r_2) \), we can conclude

\[ I_9 : STA \vdash S \vdash (SK) \rightarrow S \quad \text{Goal-3} \]
Step-10: We apply $JR$ based on $J_3$ and $I_9$

$$I_{10} : \text{STA} \models \text{STA} \xrightarrow{SK} S \text{ Goal-4}$$

Thus, Our proposed protocol achieves all the goals which indicate that STA and S mutually authenticate each other and securely generate the session key.

B. Security Verification using Scyther tool

Scyther is a formal verification tool that may be used to verify or refute the security of protocols [26]. It uses the Security Protocol Description Language (.spdl) to model the security protocols.

![Fig. 3: Scyther tool result for Mutual authentication](image)

The security characteristics of the proposed protocol are validated through the scyther tool. The validation outcome clearly indicates that our proposed protocol addresses all the security claims such as Alive (i.e., assures that the communicating parties carry out all events), Weakagre (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 shown in Fig. 3. Hence, we can deduce that the Scyther tool did not discover any attacks on the proposed protocol.

V. PERFORMANCE ANALYSIS

This section compares the proposed protocol with its counterparts in terms of security, communication, and computation costs.

A. Security features analysis

We compare the security of the proposed protocol with the existing protocols on the basis of Mutual authentication, Identity protection, protection from Replay, MITM, DoS, Privilege insider attacks, and Perfect forward secrecy. The comparison results show that the proposed protocol satisfies all security requirements of RFC-4017 and facilitates additional security requirements such as protection from privileged insider attack, as shown in Table III.

### TABLE III: Comparison of security feature and functionality analysis for mutual authentication protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
</tr>
</thead>
<tbody>
<tr>
<td>[9]</td>
<td>√</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[11]</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[12]</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[14]</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[16]</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[17]</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[18]</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

NOTE: F1: mutual authentication; F2: identity protection; F3: protection from dictionary attack; F4: protection from replay attack; F5: protection from MITM attack; F6: DoS attack protection; F7: perfect forward secrecy; F8: protection from privileged insider attack; F9: provable security; / ✓-provides the security, ×-fail to provide the security

B. Overhead analysis

In this section, we do the relative assessment of the proposed protocol with its counterparts in terms of computation and communication costs. The proposed protocol uses a combination of AES and SHA, which requires lesser cost than using RSA with DH. We use the cost (execution time) of cryptographic operations symmetric encryption/decryption ($T_{AES}$), Hash function ($T_H$), RSA signature ($T_{RSA_s}$), RSA verification ($T_{RSA_v}$), Diffie-Hellman ($T_{DH}$) as 0.0046, 0.0023, 3.8500, 0.1925 and 3.85 (ms), respectively as given in [4]. Additionally, for the communication cost, we consider the cost of communication based on previous studies as in [3] that is identity, timestamp, and random number, each requiring 32 bits. AES encryption/decryption, hashed output, public-key encryption/decryption using RSA, need 128 bits, 160 bits, 1024 bits, respectively.

The outcome of Table IV clearly indicates that the proposed protocol takes significantly less computation and communication cost compared to the state-of-the-art. It can be observed form the outcome of Table IV that the proposed protocol reduces the computation cost by 99% 37.25%, 23%, 16%, 26%, 16%, 99.87%, 99%, 99% with respect to the [2] [9], [11], [12], [14], [16], [18], [17], [19] respectively and the communication cost 16%, 16%, 19.23%, 8.6%, 8.6%, 85%, 79.41%, 74.39% with respect to [2] [9], [11], [12], [14], [18], [17], [19] respectively. Hence, we can infer that the proposed protocol is lightweight compared to all existing authentication protocols.

VI. CONCLUSION

In this paper, we present a lightweight EAP-based authentication mechanism for IEEE 802.11 WLAN. We do the detailed security analysis (using BAN logic and Scyther tool) of the proposed protocol, which shows that the proposed protocol
can solve the security issues in existing protocols. We also do the overhead analysis of the proposed protocol in terms of communication and computation cost, which shows that the proposed protocol requires less overhead than the state-of-the-art. The experimental analysis, security verification and overhead analysis clearly indicate that the proposed protocol is better than existing protocols in terms of cost and security. In the future, we would like to do the practical implementation of the proposed protocol with Commercial off-the-shelf (COTS) devices.

**TABLE IV: Comparison of computation cost for mutual authentication protocols/ Γ - Cost reduction.**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Computation cost (ms)</th>
<th>Communication cost total bits</th>
<th>1</th>
<th>1- Γ</th>
</tr>
</thead>
<tbody>
<tr>
<td>[12]</td>
<td>$6T_H + T_{RING} + 3T_{PM}$</td>
<td>$3T_H + T_{RING} + 3T_{PM} + T_{AES}$</td>
<td>15.3868</td>
<td>99%</td>
</tr>
<tr>
<td>[9]</td>
<td>$6T_H + T_{AES}$</td>
<td>$6T_H + T_{AES}$</td>
<td>0.0368</td>
<td>37.25%</td>
</tr>
<tr>
<td>[11]</td>
<td>$2T_H + 2T_{AES}$</td>
<td>$3T_H + 2T_{AES}$</td>
<td>0.0299</td>
<td>23%</td>
</tr>
<tr>
<td>[10]</td>
<td>$4T_H$</td>
<td>$6T_H + T_{AES}$</td>
<td>0.0276</td>
<td>16%</td>
</tr>
<tr>
<td>[14]</td>
<td>$3T_{AES} + 3T_H + T_{MIC}$</td>
<td>$T_{AES} + 2T_{MIC} + T_{MIC}$</td>
<td>0.0311</td>
<td>26%</td>
</tr>
<tr>
<td>[16]</td>
<td>$2T_{AES} + T_H$</td>
<td>$2T_{AES} + T_H$</td>
<td>0.0276</td>
<td>16%</td>
</tr>
<tr>
<td>[17]</td>
<td>$T_{RSA_A} + T_{DH}$</td>
<td>$T_{RSA_A} + T_{DH}$</td>
<td>17.9</td>
<td>99.8%</td>
</tr>
<tr>
<td>[18]</td>
<td>$T_{DH} + T_{RSA_A} + T_{RSA_A}$</td>
<td>$T_{DH} + T_{RSA_A} + T_{RSA_A}$</td>
<td>15.785</td>
<td>99.87%</td>
</tr>
<tr>
<td>[19]</td>
<td>$T_{RSA_A} + 2T_{AES} + T_H$</td>
<td>$T_{RSA_A} + 2T_{AES} + T_H$</td>
<td>7.7184</td>
<td>99%</td>
</tr>
<tr>
<td>Ours</td>
<td>$2T_{AES} + T_H$</td>
<td>$2T_{AES} + T_H$</td>
<td>0.023</td>
<td>99%</td>
</tr>
</tbody>
</table>

**REFERENCES**


