Title	Secure and User Efficient EAP-based Authentication Protocol for IEEE 802.11 Wireless LANs
Authors(s)	Yadav, Awaneesh Kumar, Misra, Manoj, Liyanage, Madhusanka, Varshney, Gaurav
Publication date	2020-12-13
Publication information	Yadav, Awaneesh Kumar, Manoj Misra, Madhusanka Liyanage, and Gaurav Varshney. “Secure and User Efficient EAP-Based Authentication Protocol for IEEE 802.11 Wireless LANs.” IEEE, 2020.
Conference details	The 17th IEEE International Conference on Mobile Ad-Hoc and Smart Systems (IEEE MASS - 2020), Delhi, India (held online due to Coronavirus outbreak), 10-13 December 2020
Publisher	IEEE
Item record/more information	http://hdl.handle.net/10197/12099
Publisher's statement	© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Publisher's version (DOI)	10.1109/mass50613.2020.00076

Secure and User Efficient EAP-based Authentication Protocol for IEEE 802.11 Wireless LANs

Awaneesh Kumar Yadav*, Manoj Misra†, Madhusanka Liyanage‡, Gaurav Varshney§
*†Dept. of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India
‡School of Computer Science, University College Dublin, Ireland and CWC, University of Oulu, Finland
§Dept. of Computer Science and Engineering, Indian Institute of Technology, Jammu, Jammu , India
Email: [*akumaryadav, †manoj.misra]@cs.iitr.ac.in, ‡madhusanka@ucd.ie, §gaurav.varshney@iitjammu.ac.in

Abstract*—Wireless Local Area Networks (WLANs) have experienced significant growth in the last two decades due to the extensive use of wireless devices. Security (especially authentication) is a staple concern as the wireless medium is accessible to everybody. Extensible Authentication Protocol (EAP) is the widely used authentication framework in WLANs to secure communication. The authentication mechanism designed on EAP is called EAP method. There are numerous EAP based and non-EAP based authentication protocols for WLANs, but there is no protocol that fulfills all the security requirements, as mentioned in RFC-4017 and other additional requirements like perfect forward secrecy, Denial-of-service (DoS) attack protection, and lightweight computation. Hence, it is fair to infer that there is an impelling need to design a protocol that can meet all the security requirements. In this paper, we propose a secure and user efficient EAP-based authentication protocol for IEEE 802.11 WLANs. The proposed protocol has been formally validated by BAN logic and the AVISPA tool [18]. The simulation results depict that the proposed protocol achieves all security requirements, as mentioned in RFC-4017 along with perfect forward secrecy, Denial-of-service (DoS) attack protection, and lightweight computation. The proposed protocol outperforms the existing protocols in terms of computation cost by reducing the computation cost by $\approx$ 99.9956%, 99.991%, 27.27%, 22.705% in comparison to EAP-TLS, EAP-TTLS, EAP-Ehash, EAP-SELUA, respectively.

***Keywords*—AP, AS, AVISPA, BAN, EAP, WLANs.**

I. INTRODUCTION

As the era moves towards the next generation, the demand for wireless devices (smartphones, tablets, bluetooth mice and keyboards, wireless routers, IoT, etc.) is increasing sharply [1]. It is observed that the use of WLANs have risen rapidly in the last two decades because of technological advancement [2]. WLANs are used in different areas like colleges, hospitals, airports, etc. A study conducted by CISCO suggests that the world's average mobile data traffic per user has increased from 40 megabytes to 2000 megabytes from 2012 to 2018 [3]. One of the significant advantages of WLAN is that it provides untethered connectivity to portable devices like smartphones, tablets, laptops, etc. Therefore, the security of the WLAN is a prime concern because they use an insecure public network for communication and data transfer. Wireless networks need to fulfill authentication and confidentiality as the very basic security requirements so that the users can transfer important data over the network with sufficient trust. Authentication is a way of verifying the identity of entities while accessing a resource. In the WLAN authentication, the user and authentication server verify each other by using authentication factors. It is essential for WLAN to authenticate the client and set up a secure channel between the client and the server to share the private information [4].

Development of a secure authentication mechanism that fulfills all the security requirements through which the client and the server can communicate with each other is crucial for WLANs. EAP is a generic authentication framework that supports various authentication schemes called the EAP methods. EAP framework has been defined by Internet Engineering Task force (IETF) [5]. It runs over the data link layer by the support of IEEE 802.1x . The mandatory requirements of the EAP based method are described in RFC-4017 for the WLANs environment. Some additional requirements, such as DoS attack protection, perfect forward secrecy, and lightweight computation, excluded in RFC-4017, are also desirable for WLANs authentication. There are various authentication protocols that fulfill all the requirements of RFC-4017, but they fail to meet the additional requirements, such as DoS attack protection, perfect forward secrecy, and lightweight computation. This makes the existing EAP based authentication protocols unsuitable for the practical application. Hence, there is an impelling need to design a protocol that can meet all the security requirements.

With this view, we design a lightweight EAP-based authentication protocol for the client and the authentication server. Security and performance analysis shows that: (a) it achieves all the essential security requirements (b) it takes less computation and communication costs than other related EAP-based protocols. The proposed protocol efficiently achieves a fragile balance between security and performance with very less computation time.

The rest of the paper is organized as follows. In Section II, we describe the WLAN and prevailing EAP authentication methods followed by critical security assessment. Section III demonstrates the proposed protocol. The formal and informal verification of the proposed protocol is presented in the Section IV. To analyze the performance of the proposed protocol, an experimental study is carried out in Section V. Eventually, the conclusions are drawn in Section VI.

II. Background

A. Wireless Local Area Networks (WLANs)

WLAN is a network that allows the client and authentication server to connect and communicate with each other as shown in Fig. 1. The security architecture of WLAN is defined by IEEE 802.1i that describes the flexible key hierarchy and key exchange between the client and the authentication server. IEEE 802.1i specifies the use of IEEE 802.1x that describes the reliable and secure authentication framework to set up a secure connection between the client and the authentication server or secure connection is established between client and authenticator (AP) with the help of authentication server in the IEEE 802.11 WLANs environment. EAP framework provides a flexible and reliable base for IEEE 802.1x architecture so that various authentication mechanisms can be executed over this. It defines the three participants:

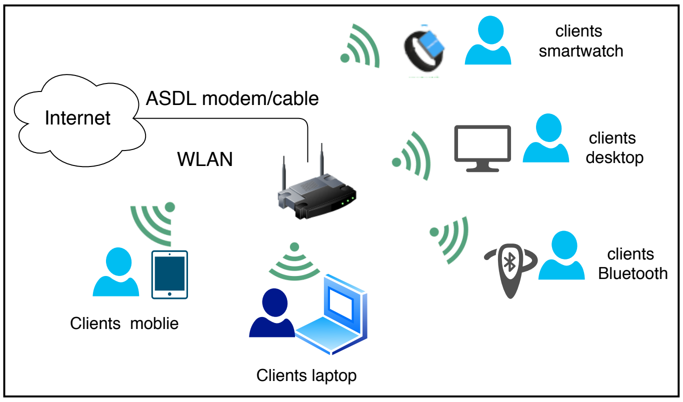

Fig. 1. The typical network structure of a WLAN

- Client (C): device that wishes to attach with the LAN. It is also called supplicant.
- Authenticator (AP): acts like a bridge between client and authentication server or network to communicate and data transfer.
- Authentication server (AS) : a backend server that is responsible for providing authentication services to the client.

B. Extensible Authentication Protocol (EAP)

EAP is very flexible and widely used authentication framework in the WLANs. RFC-3748 defines the full description of EAP framework that runs over the data link layer. Fig 2 illustrates a typical message exchange of the full EAP framework.

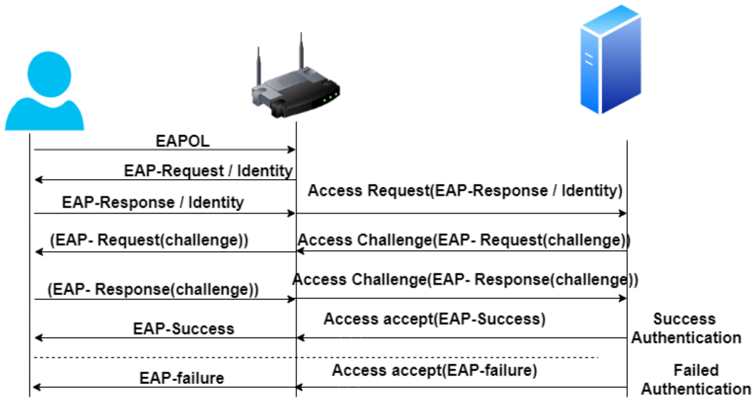

Fig. 2. EAP framework

1) Classification of EAP methods:

- Legacy based EAP methods: Legacy based EAP methods use single-factor authentication (e.g., username and password) to prove the legitimacy. These methods are defined in RFC 3748 and RFC 1994 [6].
- Certificate-based EAP methods: In the certificate-based EAP method, the client and server use a digital certificate to prove legitimacy. These methods are considered to be the foremost secure methods as compared to other EAP methods. But they require a third party for maintenance and revocation of the Certificate [7].
- Strong password-based EAP methods: In these methods, client and server convince each other that they know a secret without transmitting the secret [8] [9].

2) Mandatory Requirements:

- RFC4017 [10] defined by IETF has given some mandatory requirements for EAP methods used in IEEE 802.1x and IEEE802.11i standards. These are
  - mutual authentication support.
  - generation of symmetric keying material.
  - protection against Man-in-the –middle (MITM) attack.
  - resistance to dictionary attack, identity protection and replay attack.
- Additional requirements that are not mentioned in RFC-4017
  - Low communication and computation cost .
  - Perfect forward secrecy, protection against DoS attack.

3) EAP Methods: Several methods have been developed using the EAP framework. Only a few of them fulfill the mandatory EAP framework requirement, defined in RFC-4017. Some standard authentication methods are described below. Table I depicts the notations and abbreviations used in the background.

- EAP Transport Layer Security (EAP-TLS): It facilitates mutual authentication between the client and also the authentication server. It uses the digital certificate signed by both client and authentication server to prove the authenticity. It requires the public key infrastructure (PKI) that needs a third party for maintenance and revocation of the certificate [11].
  - Advantages:
    - Mutual authentication, perfect forwards secrecy.
    - Protection from dictionary attack, MITM, DoS attacks.

TABLE I
NOTATIONS AND ABBREVIATIONS

Notations	Description
AK/EK	Keys derived by preshared key
$RandS$ /$RandC$	Random numbers
SID	Server identity
CID	Client identity
$Algo$	Field that contains value
MIC	Message Integrity code
K	Preshared key
U_{id}	Client Id
S_{id}	Server Id
N_c, N_S	Nonce
PW	Password
PMK	Pairwise master key
F	Hash function
E_k	Symmetric encryption
D_k	Symmetric decryption

- Disadvantages:
  * High administration cost.
  * High number of the message exchange.
- EAP Tunneled Transport Layer Security (EAP-TTLS) [12]: EAP-TTLS is an extended version of EAP-TLS. It uses the combination of a public key and the certificate to prove the authenticity. In the EAP-TLS, the client and server use the digital certificate, but with the EAP-TTLS, only server uses the certificate to establish a tunnel, and the client uses the public key. It creates a secure tunnel for the client, using which the client sends information by employing a mechanism like Challenge-Handshake Authentication Protocol (CHAP) or EAP-MD5.
  - Advantages:
    * Mutual authentication, perfect forwards secrecy.
    * Protection from dictionary attack and DoS attack.
  - Disadvantage:
    * This protocol has weakness due to which several attacks are possible like Man -on-The-side attack and MITM attack [13].
    * High message exchange.
- EAP-Ehash: Omar et al. [14] proposed an authentication protocol for WLAN which is defined in two phases:
  - Registration Phase: within the registration phase, the client and the EAP server exchange the pre-shared key (PSK) and negotiate the cipher suite to prove the legitimacy at the time of authentication .
  - Authentication Phase:
    * Server derives two keys AK, EK with the help of pre-shared key (PSK).

$$AK = F(PSK, RandS) \quad (1)$$

$$EK = F(PSK, RandS, SID, CID) \quad (2)$$

    * After exchanging the identity and cipher suite, the server sends a challenge message ($Challenge$, $ServerID$, $Rands$, $Algo$, $E_K(MIC)$) to the client.

$$MIC = F(AK, Challenge, SID, RandS, Algo) \quad (3)$$

    * Since the client incorporates a pre-sheared key, it also calculates AK, EK then calculates MIC and compares it with received if it is equal, then the client authenticates the server and also sends some response message ($Randc$, $Algo$, $(HASH)_{EK}$) to the server.

$$HASH = F(AK, Challenge, RandC, Algo) \quad (4)$$

    * After receiving the message, the server verifies the message and sends a successful authentication message to the client.
  - Advantages:
    * Provides mutual authentication.
    * It takes less message exchange.
  - Disadvantages:
    * Replay attack: In this protocol, the client authenticates the server before it sends a challenge. Before authenticating the server, the client must verify the MIC, which contains the server's challenge but not his. So the challenge-response mechanism is not implemented correctly on the client-side.
    * Perfect forward secrecy: Perfect forward secrecy demands that even if an attacker knows the long term keys, he should not be able to calculate the previous session keys. In EAP-Ehash session keys are derived with the assistance of pre-shared key (PSK). Therefore if attacker gets the PSK, then he / she can easily steal the session key.

.
- EAP-SELUA: Amit et al. [15] proposed a secure and efficient authentication protocol which has two phases:
  - Registration Phase: In the registration phase, the client and server exchange their credentials. The client saves (ID, Sid, K) and sever saves (ID, K, PW) into database.
  - Authentication Phase:
    * After receiving the identity request, the client sends his identity to the server.
    * server receives the identity message from the client and sends Uid to the client.

$$Uid = (ID, MAC, Address, Time, Date). \quad (5)$$

    * After receiving the message Uid, client sends the response message ($t1$, $t2$) that is encrypted with pre-shared key K.

$$t1 = (E_K(Uid)) \quad (6)$$

$$t2 = E_K(Uid, Sid, C, N_c) \quad (7)$$

    * The server decrypts the message and verifies the parameters. If matched, then it also calculates an

access-challenge ($Respc$, PMK, N_C) and sends it to the client.

$$PMK = H(Uid, Sid, C, N_s, N_c) \quad (8)$$

$$Respc = E_K(Nc, Sid, Uid||C, N_s) \quad (9)$$

* After receiving the message client decrypts and verifies the message, if equal then it belives that server is authentic and respond to the server.

$$RespS = H(NS) \oplus E_K(Uid||PW||C) \quad (10)$$

* After receiving the response, the server decrypts the message, and if it is equal, then it believes that the client is authentic. So it sends the EAP-success message.

- Advantages:
  * Mutual authentication
  * Protection from dictionary attack
- Disadvantages:
  * Perfect forward secrecy: It violates the perfect forward secrecy that states an attacker can not steal the session key even if long term key has been compromised. In this protocol if long term key K is compromised, then the attacker can get the session key.
  * Replay attack: The client's message $t1 = E_K(Uid)$) to the server, does not include a timestamp or nonce. So it is difficult for the server to check the message freshness that gives the attacker a chance to send the repeated message.

III. Proposed Protocol

The proposed protocol involves three crucial participants: a client (C), an access point (AP), and an authentication server (AS). We assume that the connection between AP and AS is secure and reliable. The protocol consists of two phases, namely registration phase and authentication phase. In the registration phase, the client C and authentication server AS share the credentials via a secure medium. In the authentication phase, we use the combination of the symmetric encryption algorithm and hash function instead of an asymmetric algorithm that reduces the exponential computation and communication overhead (message exchange) and achieves all security requirements.

A. Registration phase

In the registration phase, client and server exchange their credentials through a secure medium. Fig. 3 demonstrates the registration phase of the proposed protocol.

Where k- long term key, p- one-time-key, UID- user identity, PW- password, SID- server identity.

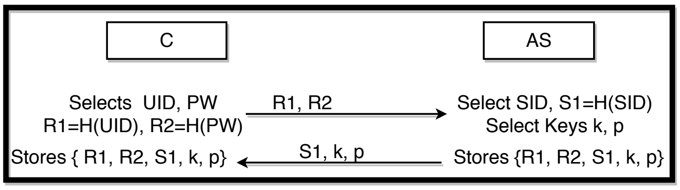

Fig. 3. Registration Phase

B. Authentication Phase

In this phase, the client and server communicate with each other to prove their authenticity. The description of the messages exchange is given below.
The symbols and abbreviations used in the paper are summarized in Table II, and the full description of the authentication phases is given in Fig. 4.

TABLE II
NOTATIONS AND ABBREVIATIONS

Notations	Description
C	Client
AP	Authenticator
AS	Authentication server
k	Long term key
p, p_n	One-time key selected by server
L	One time key selected by client
SK	Session key
$\oplus$	Xor
$\|\|$	Concatenation
UID	Client Id
SID	Server Id
PW	Password
H	One-way hash function
E_k	symmetric encryption with k
D_k	Symmetric decryption with k
T_i	Time stamp

- $C \rightarrow AP$: The client C sends a connection request to the authenticator AP by sending EAP over LAN (EAPOL).
- $AP \rightarrow C$: After receiving the connection request from C, AP sends an identity request message to C.
- $C \rightarrow AP$: Upon receiving the identity request message from AP, C sends the message CH (given in (11)) to AP, that includes $R1$ and $T1$ encrypted by pre-shared keys k and p.

$$CH = E_{k \oplus p}(R1 \parallel T1) \quad (11)$$

- $AP \rightarrow AS$: AP forwards this message (11) to the AS.
- $AS \rightarrow C$: Upon receiving the message (11), AS first decrypts the message (i.e., $D_{k\oplus p}(CH)$) to check the freshness condition (given in (12)), after that it checks whether $R1^{'}$=$R1$ ($R1^{'}$ is received from the client and $R1$ is saved in server's database that is sent by the client at the registration phase).

$$T2 - T1 < T \quad (12)$$

$$R1' = R1 \quad (13)$$

where $T, T2, T1$ denotes timeout value, message received time by AS, and message send time by C respectively. If the shared credentials are matched then AS selects p_n (where p_n is a one-time-key randomly selected by the server using steiner triple system (STS) [17] (i.e., STS is a combinatorial block design model used for key distribution stretegy)) and calculates the re-challenge RCH(given in (14)).

$$RCH = H(T1) \oplus E_{k \oplus p}(S1 \parallel T2 \parallel p_n \parallel R1), \quad (14)$$

RCH is sent to the C through AP. AS now replaces p by p_n.

- $C \rightarrow AS$: C decrypts the received message (i.e., $D_{k \oplus p}(H(T1) \oplus RCH)$) to check the freshness of the message by checking the freshness condition (given in (15)). If the message is fresh, it checks whether ($S1^{'}$= $S1$ & $R1^{'}$=$R1$)

$$T3 - T2 < T \quad (15)$$

If the later condition is true, the authenticity of the server is established (i.e., client believes that the server is authentic). C updates the one-time key p by $p \leftarrow p_n$ and deletes the old p. The client selects a key L (L is randomly selected by the client using STS to interchange the key k) and sends the hashed password with time stamp.The hashed password with time stamp is encrypted by L (given in (16)) and L is additionally encrypted with updated one-time-key p (given in (17)).

$$RES1 = H(T2) \oplus E_L(R2| \parallel T3 \parallel S1) \quad (16)$$

$$RES2 = E_p(L \parallel T3 \parallel R1) \quad (17)$$

- $AS \rightarrow AP$: After receiving the messages (16), (17) from the C, AS decrypts (i.e., $D_p(RES2)$) and checks the freshness of the message. If the freshness condition (given in (18)) holds, the decryption of $RES1$ (i.e., $D_L(H(T2) \oplus RES1)$) takes place with subject to the following condition: check whether ($R2^{'}$ = $R2$ & $S1^{'}$=$S1$); If all the conditions hold, the authenticity of the client is established (i.e., server believes that the client is authentic).

$$T4 - T3 < T \quad (18)$$

So AS replaces k by L and selects a new p_n after that it calculates the Session key (SK) (i.e, $SK = (T4 \oplus T3) \parallel p_n$), encrypts it and sends it to the AP.

- $AP \rightarrow C$: AP passes this message SK (i.e., $E_L(T4 \oplus T3) \parallel p_n$) to C .
- After receiving the message $E_L((T4 \oplus T3)\| \ p_n)$ from AP, C decrypts the message and verifies the freshness of the message by checking the freshness condition (given in (19)). If the condition holds, C saves the session key $((T4 \oplus T3)\| \ p_n)$ and updates the p by $p \leftarrow p_n$ for further communication.

$$T5 - T4 < T \quad (19)$$

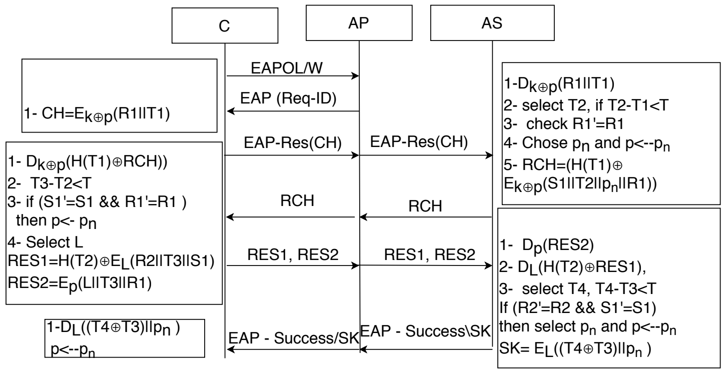

Fig. 4. Proposed Protocol

Algorithm 1: Mutual authentication and session key generation

---

Input: ($R1$, $R2$, $S1$, k, p) is exchanged between client and server during registration phase.
Output: A session key.
Procedure:

1. $C \rightarrow AP$: Initially, the client sends a connection request to AP.
2. $AP \rightarrow C$: AP demands Id to the Client.
3.$C \rightarrow AP$: The client sends a challenge (CH) to AP.
4. $AP \rightarrow AS$: AP forwards this message to AS.
5. $AS \rightarrow AP$: AS computes a response challenge (RCH) in reply to the challenge and sends it to C.
6. $AP \rightarrow C$: AP sends this message to Client.
7. $C \rightarrow AP$: Client decrypts the RCH after that it encrypts the RES1 with chosen key L and encrypts RES2 by key p and sends them to AP.
8. $AP \rightarrow AS$: AP forwards the message to AS.
9. $AS \rightarrow AP$: Server decrypts $RES2$ and $RES1$ and generates the session key and sends it to AP.
10. $AP \rightarrow C$: AP forwards this message to Client.
11. C: Client saves the session key for further communication.
12. Returns $E_L((T4 \oplus T3)\| \ p_n))$

IV. Security Analysis

A. Informal security analysis

Our proposed protocol achieves the following goals.

1) Mutual Authentication: In step 2, the C receives the message (14) from the AS; C decrypts the message (14) and verifies the credentials. If credentials are correct, then the client authenticates the server. On the other hand, the AS receives the message (16), (17) in the 3rd message exchange. AS then decrypts the message, and if it receives the correct response, the server authenticates the client. Thus, our proposed protocol provides mutual authentication.

2) Dictionary attack: A dictionary attack is not possible within the proposed protocol because the server and client store the credentials within the hashed form ($R1$, $R2$, S). The client shares the password when it is confirmed that the server is authentic. It uses a secret key L to encrypt the password because if anyhow the previous key is hacked, the attacker can not steal the password. So, it is difficult for an attacker to guess a valid password through the intercepted message.

3) Man-in-the-middle attack: In the proposed protocol, an attacker can not modify, intercept, and send the modified message to the client or server. We are using two factors that make the protocol more strong to avoid the MITM attack, that are

- Server regularly generates new key p_n and after each message exchange client updates the key by $p_n \rightarrow p$.
- The client chose a key L to send a password $RES1 = H(T2) \oplus E_L(R2| \parallel T3 \parallel S1)$ and that key is also used for session key encryption $E_L((T4 \oplus T3) \| \; p_n)$ by the server so that only authentic user could decrypt the message.

4) Perfect forward secrecy: It means that an attacker can not steal the session key even if long term key has been compromised. In order to maintain the perfect forward secrecy in the proposed protocol, we have used a brand new key L and an updated key p. If anyhow attacker gets k and p, he can not calculate the session key because the server does not use the pre-shared key k and p to generate the session key.

5) Replay attack : In the proposed protocol, the timestamp (T_i) is used to check the message's freshness. It calculates the time difference between message send (T_{i-1}) and message received (T_i) time ($T \geqslant T_i - T_{i-1}$). If the condition holds, the message is fresh; otherwise, it simply discards the message.

6) Identity protection : In order to protect the identity, client's and server's ids are used in hashed form, and these ids are exchanged with the assistance of a strong key encryption mechanism.

7) Denial-of-service (DoS) attack: To avoid the DoS, we perform two actions: (a) we use the timestamp for every message exchange to verify the freshness of the proposed protocol. (b) we have efficiently utilized the one-time key p (i.e., one time key p is updated in every session). So it is not easy for an attacker to send the repeated message. Subsequently, the proposed protocol is secure from Denial-of-service attack protection.

B. formal security analysis using BAN logic

We have used the BAN logic to verify the proposed protocol [17]. The Ban logic rules, assumptions, idealized forms, and security goals, are described below. Table III depicts the notation used to describe the BAN logic assumption and rules in the proposed protocol.

TABLE III
BAN NOTATIONS

Symbol	Description
$A \mid\equiv M$	The principal A believes that message M is true
$A \triangleleft M$	The principal A receives a message M
$A \mid\sim M$	The principal A transmits the message M
$A \Rightarrow M$	The principal A controls M
$\#(M)$	Message M is fresh and not used previously
$\langle M \rangle_k$	Message M is combined with secret key k
$\{M\}_k$	Message M is encrypted with secret key k
$A \mid\equiv A \overset{k}{\leftrightarrow} B$	The secret key (k) is used by the principal A and B.

- BAN Logic Rules:-
  - Message meaning rule:- If A believes that k is shared between $A\&B$, A sees that M is encrypted with k then A believes that B has sent the M,

$$\frac{A \mid\equiv A \overset{k}{\leftrightarrow} B, A \triangleleft \{M\}_k}{A \mid\equiv B \mid\sim M}$$

  - Nonce verification rule:- If A believes that M is fresh and A believes that B has sent M then A believes that B believes M,

$$\frac{A \mid\equiv \#(M), A \mid\equiv B \mid\sim M}{A \mid\equiv B \mid\equiv M}$$

  - The jurisdiction rule:- If A believes that B has jurisdiction over M and A believes that B believes M then A believes M,

$$\frac{A \mid\equiv B \Rightarrow M, A \mid\equiv B \mid\equiv M}{A \mid\equiv M}$$

  - The belief rule:- If A believes at M and A believes at N then A believes (M, N)

$$\frac{A \mid\equiv M, A \mid\equiv N}{A \mid\equiv (M, N)}$$

- Following assumptions hold for the initial state of the protocol

$$L1 : C \mid\equiv C \overset{k}{\leftrightarrow} AS$$

$$L2 : C \mid\equiv C \overset{p}{\leftrightarrow} AS$$

$$L3 : C \mid\equiv \#(T2)$$

$$L4 : C \mid\equiv \#(T4)$$

$$L5 : C \mid\equiv C \overset{L}{\leftrightarrow} AS$$

$$L6 : C \mid\equiv AS \Rightarrow p_n$$

$$L7 : AS \mid\equiv C \overset{k}{\leftrightarrow} AS$$

$$L8 : AS \mid\equiv C \overset{p}{\leftrightarrow} AS$$

$$L9 : AS \mid\equiv \#(T1)$$

$$L10 : AS \mid\equiv \#(T3)$$

$$L11 : AS \mid\equiv C \Rightarrow L$$

$$L12 : C \mid\equiv AS \Rightarrow (C \overset{SK}{\longleftrightarrow} AS)$$

$$L13 : AS \mid\equiv C \stackrel{p_n}{\longleftrightarrow} AS$$

- Goals of authentication: Our proposed protocol for client and server is considered complete if it achieves the following goals:

$$C \mid\equiv AS \mid\equiv (C \stackrel{SK}{\longleftrightarrow} AS)$$

$$C \mid\equiv (C \stackrel{SK}{\longleftrightarrow} AS)$$

- The protocol is idealized as:
M_1: $C \rightarrow AS$: $(R1 \parallel T1)_{k\oplus p}$,
M_2: $AS \rightarrow C$:$(R1 \parallel T2 \parallel S1 \parallel AS \stackrel{p_n}{\longleftrightarrow} C)_{k\oplus p}$,
$M_{3.1}$: $C \rightarrow AS$: $(R2 \parallel T3 \parallel S1)_L$,
$M_{3.2}$: $C \rightarrow AS$:$(R1 \parallel T3 \parallel C \stackrel{L}{\leftrightarrow} AS)_{p_n}$,
M_4: $AS \rightarrow C$: $(AS \stackrel{SK}{\longleftrightarrow} C)_L$.

1) Proof and derivation of security goals: We have analyzed the idealized form of the propose protocol:

- Based on the assumptions $L7$ and $L8$, and we apply message meaning rule on M_1

$$R1 : \frac{AS \mid\equiv C \stackrel{k\oplus p}{\longleftrightarrow} AS, AS \triangleleft \{M_1\}_{k\oplus p}}{AS \mid\equiv C \mid\sim M_1}$$

- Based on the assumption $L9$, and we apply timestamp verification rule on M_1, we get

$$R2 : \frac{AS \mid\equiv \#(T1), AS \mid\equiv C \mid\sim M_1}{AS \mid\equiv C \mid\equiv R1}$$

- Based on the assumptions $L1$ and $L2$, and we apply message meaning rule on M_2

$$R3 : \frac{C \mid\equiv C \stackrel{k\oplus p}{\longleftrightarrow} AS, C \triangleleft \{M_2\}_{k\oplus p}}{C \mid\equiv AS \mid\sim M_2}$$

- Based on the assumption $L3$, and we apply timestamp verification rule on M_2

$$R4 : \frac{C \mid\equiv \#(T2), C \mid\equiv AS \mid\sim M_2}{C \mid\equiv AS \mid\equiv R1},$$

$$R5 : \frac{C \mid\equiv \#(T2), C \mid\equiv AS \mid\sim M_2}{C \mid\equiv AS \mid\equiv S1},$$

$$R6 : \frac{C \mid\equiv \#(T2), C \mid\equiv AS \mid\sim M_2}{C \mid\equiv AS \mid\equiv AS \stackrel{p_n}{\longleftrightarrow} C}$$

- Based on the assumption $L6$, and we apply Jurisdiction rule on M_2

$$R7 : \frac{C \mid\equiv AS \Rightarrow p_n, C \mid\equiv AS \mid\equiv AS \stackrel{p_n}{\longleftrightarrow} C}{C \mid\equiv C \stackrel{p_n}{\longleftrightarrow} AS},$$

- Based on the assumption $L13$, and we apply message meaning rule on Message $M_{3.2}$

$$R8 : \frac{AS \mid\equiv C \stackrel{p_n}{\longleftrightarrow} AS, AS \triangleleft \{M_{3.2}\}_{p_n}}{AS \mid\equiv C \mid\sim M_{3.2}}$$

- Based on the assumption $L10$, and we apply timestamp verification rule on $M_{3.2}$

$$R9 : \frac{AS \mid\equiv \#(T3), AS \mid\equiv C \mid\sim M_{3.2}}{AS \mid\equiv C \mid\equiv R1},$$

$$R10 : \frac{AS \mid\equiv \#(T3), AS \mid\equiv C \mid\sim M_{3.2}}{AS \mid\equiv C \mid\equiv C \stackrel{L}{\leftrightarrow} AS.}$$

- Based on the assumption $L11$, we apply jurisdiction rule $M_{3.2}$

$$R11 : \frac{AS \mid\equiv C \Rightarrow L, AS \mid\equiv C \mid\equiv C \stackrel{L}{\leftrightarrow} AS}{AS \mid\equiv C \stackrel{L}{\leftrightarrow} AS}$$

- Based on the $R11$, we apply message meaning rule on message $M_{3.1}$

$$R12 : \frac{AS \mid\equiv C \stackrel{L}{\leftrightarrow} AS, AS \triangleleft \{M_{3.1}\}_L}{AS \mid\equiv C \mid\sim M_{3.1}}$$

- Based on the assumption $L10$, and we apply timestamp verification rule on message $M_{3.1}$

$$R13 : \frac{AS \mid\equiv \#(T3), AS \mid\equiv C \mid\sim M_{3.1}}{AS \mid\equiv C \mid\equiv R2},$$

$$R14 : \frac{AS \mid\equiv \#(T3), AS \mid\equiv C \mid\sim M_{3.1}}{AS \mid\equiv C \mid\equiv S1,}$$

- Based on the assumption $L5$, we apply Message meaning rule on M_4

$$R15 : \frac{C \mid\equiv C \stackrel{L}{\leftrightarrow} AS, C \triangleleft \{M_4\}_L}{C \mid\equiv AS \mid\sim M_4}$$

- Based on the assumption $L4$, we apply timestamp verification rule on M_4

$$R16 : \frac{C \mid\equiv \#(T4), C \mid\equiv AS \mid\sim M_4}{C \mid\equiv AS \mid\equiv (C \stackrel{SK}{\longleftrightarrow} AS)}$$

$$Goal_1 : C \mid\equiv AS \mid\equiv (C \stackrel{SK}{\longleftrightarrow} AS)$$

- Based on the assumption L12, the jurisdiction rule

$$R17 : \frac{C \mid\equiv AS \Rightarrow (C \stackrel{SK}{\longleftrightarrow} AS), C \mid\equiv AS \mid\equiv (C \stackrel{SK}{\longleftrightarrow} AS)}{C \mid\equiv (C \stackrel{SK}{\longleftrightarrow} AS)}$$

$$Goal_2 : C \mid\equiv (C \stackrel{SK}{\longleftrightarrow} AS)$$

This concludes the proof of our security goals.

C. formal security analysis using AVISPA

We have performed formal verification by using the AVISPA tool to verify the proposed protocol. We have used the Constraint-Logic (CL-AtSe) backend server of the AVISPA [18]. The result of the tool depicts that the proposed protocol protects from various attacks, as shown in Fig. 5. All the simulations are performed on Intel(R) Core(TM) i5-3210M under the Window 10 in 64-bit mode with 4GB RAM.

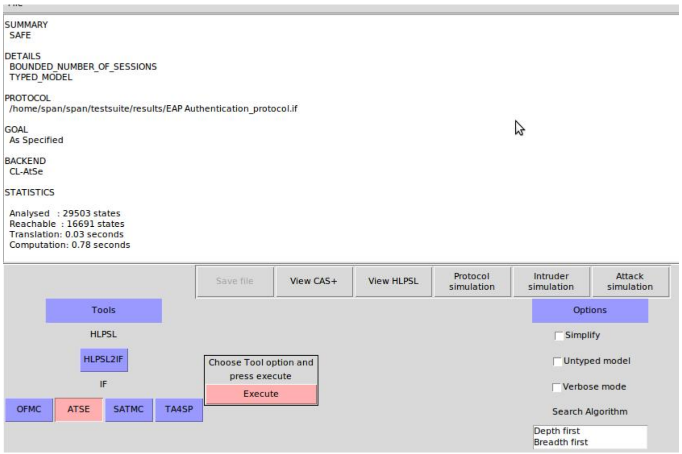

Fig. 5. Security analysis of the proposed protocol using AVISPA (CL-AtSe) model [18]

V. Performance Analysis

This section reports a set of experiments to demonstrate the performance of the proposed protocol. In order to assess the relative performance of the proposed protocol, we compare it against four existing protocols. In this set of experiments, first, we analyze the performance of our proposed protocol in terms of some crucial security features, i.e., mutual authentication, perfect forward secrecy, dictionary attack protection, replay attack protection, identity protection, MITM, DoS attack protection. The results obtained are reported in Table IV. From the results shown in Table IV, it is clear that unlike its counterparts, the proposed protocol achieves every crucial security features.

TABLE IV
Comparison of security features/ $\surd$-Yes, $\times$-No

Features /Methods	[11]	[12]	[14]	[15]	Ours
Mutual authentication	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Perfect forword secrecy	$\surd$	$\surd$	$\times$	$\times$	$\surd$
Dictionary attack protection	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Replay attack protection	$\surd$	$\surd$	$\times$	$\times$	$\surd$
Identity protection	$\surd$	$\surd$	$\surd$	$\times$	$\surd$
MITM attack protection	$\surd$	$\times$	$\times$	$\surd$	$\surd$
DoS attack protection	$\surd$	$\surd$	$\times$	$\times$	$\surd$
No.of message exchange	8	8	6	6	4

To make the algorithm more robust, we have adopted the Advanced Encryption Standard (AES) and hash function for encryption and decryption operation in the proposed protocol. We have used the CryptoPP library to simulate the proposed protocol that is tested on Intel(R) Core(TM) i5-3210M under the Window 10 in 64-bit mode with a CPU frequency of 2.50 GHz. Table V depicts the notation and cost estimation used in the proposed protocol [19].

In Table VI, we have shown the computation time of our proposed protocol. It is obvious from the results shown in Table VI that the proposed protocol requires less computation time in comparison to other related protocols. The rationale behind this is that in the proposed protocol, the combination of the AES and hash function is used, which has a clear edge over asymmetric algorithms in terms of computational cost, i.e., the combination of AES and hash function yields better (less) cost compared to that of asymmetric algorithms. The numerical values shown in Table V, depict that the proposed protocol reduces the computation cost by $\approx$ 99.9956%, 99.991%, 27.27%, 22.705% with respect to [11], [12], [14], [15], respectively.

TABLE V
The notation utilized in the computation time estimation

Symbol	Discription	Cost
T_H	The execution Time for calculating a message digest	=3.61*10^2 cpu cycles
T_{DH}	The execution Time for Diffie-Hellman Key agreement	=17.24*10^6 cpu cycles
T_{AES}	The execution Time for 128 bit AES encryption/decryption	=4.48*10^2 cpu cycles
T_{RSA_s}	The execution Time for signing an RSA signature	=31.17*10^6 cpu cycle
T_{RSA_v}	The execution time for signing an RSA verifying	=.66*10^6 cpu cycles
T_{MIC}	The execution time for VMAC message	= .3044*10^2 cpu cycles

TABLE VI
Comparison of computation time between the proposed protocol and other related protocols

Protocol	Total computation	time (ms)	Cost Reduction
[11]	$2T_{DH}$ + T_{RSA_s}+T_{RSA_v} =66.97*10^6 cpu cycles	36.42 ms	99.9956%
[12]	$2T_{DH}$ + T_{RSA_v} =35.14*10^6 cpu cycles	19.2 ms	99.991%
[14]	$6T_H$ +4 T_{AES} =39.58*10^2 cpu cycles	2.2*10^{-3} ms	27.27%
[15]	$2T_H$ +$6T_{AES}$ + T_{MIC} =38.06 *10^6 cpu cycles	2.07*10^{-3} ms	22.705%
Ours	$2T_H$+$5T_{AES}$ =29.62.97*10^2 cpu cycles	1.60*10^{-3} ms	

Fig. 6 shows the comparison of message exchange between the proposed protocol with other related protocols. The number of messages required by the proposed protocol is four, approximately half of the number of messages required by its counterparts.

Fig. 7 demonstrates the computation time of the proposed protocol with respect to other related protocols. We avoid the asymmetric encryption that increases the cost because it requires the certificates at the client or server-side to verify each other.

VI. Conclusion

Providing a secure and user efficient authentication protocol for secure communication between client and server is still a crucial issue for WLANs. In this paper, we first analyzed the standard EAP based authentication protocol followed by critical security assessment. We informally proved that although existing protocols fulfill the mandatory requirement mentioned in RFC-4017, they fail to meet the other additional

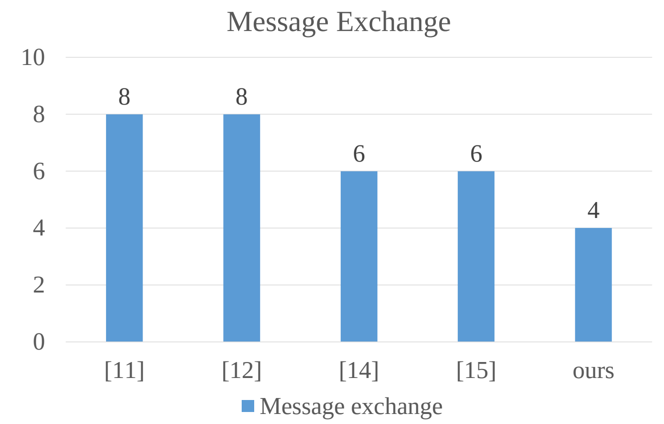

Fig. 6. Comparison of message exchange between our proposed protocol and other releted protocols

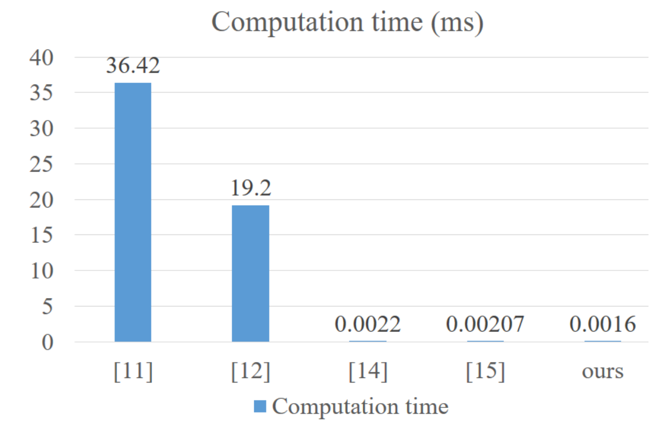

Fig. 7. Comparison of computation time between our proposed protocol and other related protocols

requirements like DoS attack protection, perfect forward secrecy, and light-weight computation. That makes the existing EAP based authentication protocol unsuitable for practical application. To make the protocol secure and user efficient, it is necessary to achieve all the essential security requirements mentioned in RFC-4017 and other additional requirements. We proposed a secure and user efficient authentication protocol for client and server that delivers all the essential security requirements mentioned in RFC-4017 along with other additional requirements. We informally and formally (BAN logic, AVISPA) proved that the proposed protocol achieves all the mandatory requirements, as mention in RFC-4017 and other additional requirements. We computed the performance of the proposed protocol, which demonstrates that it requires less computation and communication costs with respect to other related protocols. The proposed protocol efficiently achieves a fragile balance between security and performance with minimum computation time.

An immediate extension of this work is to extend the existing protocol to develop a fast reconnect protocol so that secure handover could be achievable when a client moves from one service domain to another.

References

[1] Qu Qiao, Bo Li, Mao Yang, Zhongjiang Yan, Annan Yang, Der-Jiunn Deng, and Kwang-Cheng Chen, "Survey and performance evaluation of the upcoming next generation wlans standard-ieee 802.11 ax," *Mobile Networks and Applications*, no.5, pp. 1461-1474, 2019.

[2] Y. Lin, L. Shao, Z. Zhu, Q. Wang, and R. K. Sabhikhi, "Wireless network cloud: Architecture and system requirements," *IBM Journal of Research and Development*, vol. 54, no.1, pp. 1-12, 2010.

[3] Cisco Visual Networking Index, "Global Mobile Data Traffic Forecast Update," *White Paper*, pp. 2015-2020, 2015.

[4] A. K. Yadav, B. Mahapatra, S. Kumar and A. K. Turuk, "Multi-layer Authentication and Key Agreement Protocol for Secured Data Transmission in Cloud-RAN," *15th IEEE India Council International Conference (INDICON)*, pp. 1-6, 2018.

[5] Aboba Bernard, Larry Blunk, John Vollbrecht, James Carlson, and Henrik Levkowetz, "Extensible authentication protocol (EAP)," *RFC-3748*, 2004.

[6] W. Simpson, "PPP challenge handshake authentication protocol (CHAP)," *RFC-1994*, 1996.

[7] N. Cam-Winget, D. McGrew, J. Salowey, and H. Zhou, "Dynamic Provisioning Using Flexible Authentication via Secure Tunneling Extensible Authentication Protocol (EAP-FAST)," *RFC-5422*, 2009.

[8] David Q. Liu and Mark Coslow, "Extensible authentication protocols for IEEE standards 802.11 and 802.16," *In Proceedings of the international conference on mobile technology, applications, and systems (Mobility)*, pp. 1-9, 2008.

[9] Chun-I Fan, Yi-Hui Lin, and Ruei-Hau Hsu, "Complete EAP method: User efficient and forward secure authentication protocol for IEEE 802.11 wireless LANs." *IEEE transactions on parallel and distributed systems* pp. 672-680, 2012.

[10] D. Stanley, J. Walker, B. Aboba, "Extensible authentication protocol (EAP) method requirements for wireless LANs," *RFC-4017*, 2005.

[11] E. Rescorla, T. Dierks, "The transport layer security (TLS) protocol version 1.3," *RFC-8446*, 2018.

[12] P. Funk, and S. Blake-Wilson, "Extensible authentication protocol tunneled transport layer security authenticated protocol version 0 (EAP-TTLSv0)," *RFC-5281*, 2008.

[13] Asokan Nadarajah, Valtteri Niemi, and Kaisa Nyberg, "Man-in-the-middle in tunnelled authentication protocols," *In International Workshop on Security Protocols Springer, Berlin, Heidelberg*, pp. 28-41, 2003.

[14] O. Cheikhrouhou, M. Laurent, A. B. Abdallah, M. B. Jemaa, "An EAP-EHash authentication method adapted to resource constrained terminals," *Annals of telecommunications-annales des télécommunications*, pp.271-284, 2010.

[15] Amit Kumar and Hari Om, "A secure, efficient and lightweight user authentication scheme for wireless LAN," *In 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pp. 1-9, 2016.

[16] A. K. Yadav, B. Mahapatra, and A. K. Turuk, "A Secure Key Management and Authentication Protocol for Virtualized-BBU in C-RAN Architecture", *International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, pp. 1-6, 2018.

[17] Martin Abadi, and Mark R. Tuttle, "A logic of authentication," *ACM Transactions on Computer Systems*, 1990.

[18] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J.Cullar, P. H. Drielsma, P. C. Ham, O. Kouchnarenko, M. Mantovani, and S. Mdersheim, "The AVISPA tool for the automated validation of internet security protocols and applications," *International conference on computer aided verification*, pp. 281-285, 2005.

[19] https://www.cryptopp.com/benchmarks-p4.html