Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

8th International Conference
on Electrical Engineering
ICEENG 2012

A Moderate Weight EAP Authentication Method (EAP-MEAP) for Wireless Local Area Network

By

AHMED EL- NAGAR* AHMED ABD EL-HAFEZ** ADEL EL-HNAWY***

Abstract

IEEE 802.11standard for Wireless Local Area Networks (WLANs) is facing more and more problems linked to security threats, which expose legitimate users to increased risk. Therefore, the security is always a major concern for WLAN development and one of the major challenges in WLAN security issue is authentication.

Extensible Authentication Protocol (EAP) has been widely used for that important aspect. EAP is a framework of authentication process that uses several methods to perform that process. In this paper, the flows of the existing EAP methods will be analyzed and illustrated.

Then, a new EAP method Extensible Authentication Protocol -Moderate Weight Extensible Authentication Protocol "EAP-MEAP" will be proposed. This method combines between the simplicity of deployment and management of password methods and the robustness of certificated ones. EAP-MEAP can be used widely in IEEE802.11 for WLANs (Wi- Fi and its application domains) as solution to the presented flaws. A security assessment to the proposed protocol will be presented.

Finally, the checked and verified results of the EAP- MEAP security properties using the specialized model checker AVISPA, which provides formal proofs of the security protocols.

Key words:

Wireless network; Security protocol; Access control; EAP; HLPSL.

---

* Egyptian Armed Forces

** Military Technical College Staff, Cairo, Egypt
*** College of Engineering Staff, Ain Shams University, Cairo, Egypt

1. Introduction

During the last three decades, the use of wireless communication technologies has been growing. This is due to the new applications domain such as mobile internet services, which introduced in the multiple high technologies solutions such as laptops, smart phones and tablets. Most of these solutions use unsecured wireless public networks to communicate among the mobile clients where, sensitive information, like user name, password, or data that require high security levels needs to exchange among them.

Securing communication in WLANs is a complex problem [1, 2, 3, 4, 5] due to communicating mobile clients needs a way for both mutually prove their identity between them and verify the contents of their data traffic that manipulated between them is free of tampering or sniffing.

There are three goals must be met to have a successful security strategy in wireless networks; mutual authentication, private communication (privacy) and data integrity.

The first generation of wireless technologies had a bad reputation, due to their poorly designed security strategy by using WEP (Wired Equivalent Privacy) protocol.

To overcome all serious weaknesses found in WEP, IEEE has developed the 802.11i standard, which offers a strong security strategy by using the WPA (Wi–Fi Protected Access) protocol. Unlike in the WEP, the encryption key is based on the TKIP (Temporal Key Integrity Protocol).

To reinforce the security strategy and to give more flexibility to the wireless network users, the IEEE invented the 802.1x standard. This new release provides an intelligent authentication mechanism based on EAP [6].The success of the EAP is the distinction between the EAP protocol and the EAP methods that are used. The principal function of the EAP protocol is a framework to encapsulate the confidential data (login, password, certificate, etc.) used for the EAP authentication methods. In addition, the EAP methods take in charge the authentication process itself. As a result, the EAP protocol is not attached to a particular EAP method, and in case a security flaws in one method are discovered, this method can be simply changed without changing all the protocol or platform. Currently, many EAP methods exist, but only few of them are standardized in the Internet Engineering Task Force (IETF) organization.

In this paper, the existing EAP methods will be analyzed, such as password methods (EAP-MD5, EAP-LEAP), certificated methods (EAP-TLS), tunnel and protected

methods (EAP- TTLS, EAP- PEAP) and will propose a new EAP method that named "EAP-MEAP".

The rest of this paper is organized as follows: section 2 describes the EAP authentication framework. Section 3 briefly reviews the possible wireless attacks, section 4 provides overviews of a variety of EAP authentication methods, followed by a critical analysis. Section 5, 6 introduce and illustrates the messages flow of the new proposed method "EAP-MEAP" and section 7 assessments the EAP-MEAP. Section 9 illustrates the specification and validation results of the "EAP-MEAP" by using AVISPA tool presented in section 8. Finally, we draw our concluding remarks in section10.

2. Extensible Authentication Protocol

The (RFC 2284)[9] that defined by IETF ,addressed EAP as an authentication protocol that typically rides on top of other protocols such as 802.1x.

The protocol 802.1x supports the EAP protocol as an authentication protocol between the client and the Authenticator Server (AS) via the authenticator or Access Point (AP) [6]. EAP typically runs directly over data link layers such as IEEE 802 protocol, without requiring an IP address. This protocol is named extensible protocol because it supports several methods to be fulfilled. These methods support authentication credentials that include ID, Password, certificates, and other.

The main advantage of the EAP architecture is its flexibility, because it is independent from the user authentication method where, other methods can be added or modified to the same working EAP framework in the network without defining new ones.

The RFC 4017[8] and RFC 3748 [6] that defined by IETF, listed eight desired properties of the EAP methods used as a criterion to evaluate these different methods:

. Mutual Authentication
. Identity Privacy
. Dictionary Attack Resistance
. Replay Attack Resistance
. Derivation of strong and dynamic session keys
. Tested Implementation
. Delegation
. Fast Reconnect

EAP protocol introduces three principal entities as follow:

EAP peer (Client): corresponds to the entity to authenticate.

EAP authenticator (Access Point): corresponds to the entity that has control of the authentication process.

EAP server (Authenticator Server): corresponds to the entity capable of authenticating the EAP client.

The figure- 1 illustrates four types of EAP messages (request, response, success and failure) which encapsulated in EAP packet and their exchange between EAP client and EAP authenticator server for authentication process.
The procedures start by EAP client who broadcast an EAPOL (EAP over LAN)-Start packet frame. After that, EAP authenticator replies by asking EAP client for his identity through an EAP-Request identity message.

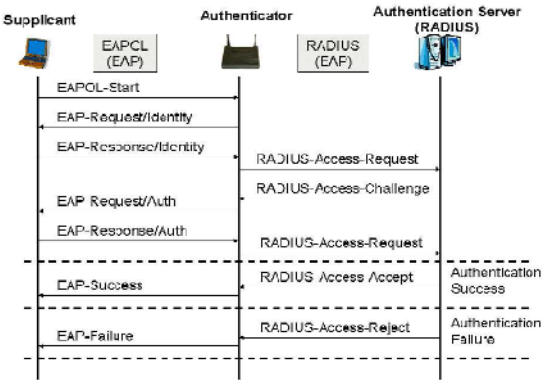

Figure-1: *EAP messages exchange*

After that, EAP client identifies himself through an EAP-Response identity message. EAP authenticator forwards that identity to the EAP authenticator server. After that, EAP authentication server sends an EAP request message with a specific authentication method to EAP client via EAP authenticator, then EAP client reply with an EAP response message for agreement to uses that same specific authentication method, and the authentication process starts.

At the end of executing the authentication method procedures, EAP authenticator server sends EAP-Success message or an EAP-Failure message to EAP client via EAP authenticator according to the statues of the authentication process success or fail.

3. EAP Methods Possible Attacks

In this section, a number of possible attacks [10, 11, 12] on the EAP methods that used in WLANs will presented as follow:

. Sniffing (Eavesdropping)

The nature of an RF (wireless) based network leaves it open to packet eavesdropping or sniffing by any radio within range of a transmitter.

. Invasion and Resource Stealing

Invasion and resource stealing would allow an attacker direct access to all devices within a network.

. Traffic Redirection

An intruder can change or redirected the route of the traffic from destined EAP wireless client to a particular attacking EAP wire client.

. Denial of Service (DoS)

Two types of DoS attacks against a WLAN can exist, in the first case; the intruder tries to bring the whole network to its knees by causing excessive interference.

In the second case, an attacking client sends 802.11 dissociate message or an 802.1x EAPOL-logoff message to the target client and effectively disconnects it.

. Rouge (Simulate) Access Point

A rogue Access Point is one that installed by an attacker to providing false information to either the EAP wireless client or the EAP wireless server also, collects the true packets from both sides.

4. EAP Existing Methods

In this section, a number of the existing EAP methods based on the different authentication approaches will be presented and analyzed [2, 35].

. Secret-key (Password) Approach

In secret-key (password) authentication methods, the AS and the client have the same secret and establish trust by proving to each other the knowledge of the shared secret key.

EAP-MD5 (EAP-Message Digest 5)

EAP-MD5 which described in RFC 2284[9] is primarily based on one-way hash function. When an account is created and a user types in his password, the authentication server stores the hash generated by a one-way hash function.

When the user wants to login to the system later, the user computes the hash value with it is the password using the same one-way hash function then, the hash value is transmitted over the network. If the hash received is same as the one stored in the authentication server, the user is authenticated.

As mentioned, even knowing the hash, it is computationally difficult to derive the original password producing the hash. The user password is not stored in clear text in authentication server, it will not be disclosed.

In the EAP-MD5, the attackers can easily sniff a station's identity, which is passed in clear text and password hash. Therefore, MD5 is more vulnerable to replay attack .It does not provide a means to derive dynamic keys per session.

This method is open to a dictionary attack. However, the attacker can obtain the challenge and the hashed response, and then apply any of dictionary methods. Then he knows the supplicant’s password and can steal its identity, to gain access to the network.

With only client side authentication (no mutual authentication), EAP-MD5 is also vulnerable to Man-In-The-Middle attacks. It can allow a client to talk to a rogue AP.

EAP-MD5 is typically not suitable for wireless LAN implementations, especially when strong security is required.

EAP-LEAP (EAP-Lightweight Extensible Authentication Protocol)

LEAP is developed by Cisco system [13, 34] for use on WLANs that use Cisco 802.11 wireless devices.

Initially, the client and the AS share a secret key then ,the client sends a random challenge (S)to the AS, and the AS responds to the challenge by encrypting it with the share secret key .The client authenticates the AS by decrypting the response from the AS and comparing it to the challenge. If the decrypted response matches the challenge, the AS is authenticated. Similarly, the AS authenticates the client with a random challenge(C).

If the mutual authentication is successful, the client and the authentication server derive a temporary session key from the information exchanged during the authentication process.

LEAP uses a log-on password as a shared secret and it offers mutual authentication between client and AS, this feature eliminates the man in the middle attacks by rogue APs.

It encrypts data transmissions using dynamically generated keys, and with LEAP, session keys are unique to users and not shared among them.

Although LEAP supports previous features, LEAP has some flaws described below.

LEAP does not protect the client's identity (username and password) because the EAP identity messages are sent in plaintext.

Moreover, because an eavesdropper can easily sniff the challenge-response pair sent between clients and AS. Therefore, that LEAP is vulnerable to dictionary attacks. LEAP also does not consider other desired properties such as delegation and fast reconnect.

LEAP is secure theoretically; if complex, enough passwords are used. Therefore, it is computationally infeasible to attempt an offline dictionary or brute force attack.

. Public-Key (Certificated) Approach

Unlike the secret-key approach, the public-key approach uses a mathematically connected key pair, a public key and a private key. If the client wishes to authenticate the AS, the client encrypts a challenge with the AS's public key and challenges the AS to prove its identity by decrypting the challenge with the AS's private key.

After the AS decrypts the challenge, it encrypts the challenge with the client's public key so that only the client, who has the corresponding private key, can decrypt it.

To insure that a client's public key is legitimate and to prevent an imposter from advertising his public key as a legitimate client's key, the AS and the client need to establish trust, typically through Certification Authorities (CAs), trusted independent third parties that issue certificates.

CAs signs their certificates using their private key so that one can verify the validity of the certificate using their public key. Clients are assumed to have, in advance, a copy of the CA's public key to use for validating certificates.

EAP-TLS (EAP-Transport Layer Security)

IETF defines EAP-TLS [7, 14] as based on a certificate approach, where requires trusted CAs, and uses Transport Layer Security protocol (TLS) to provide secure communicating over network .TLS is a standardized version of the Secure Socket Layer (SSL) protocol.

SSL uses a combination of cryptographic processes to provide secure communication over a network.

Communication using SSL begins with an exchange of information between the client and the server, this exchange of information is called the SSL handshake.

The three main purposes of the SSL handshake are:

- Negotiate the cipher suite they will use.
- Mutual authenticate identity for both client and server where, allowing each of two communicating parties to ensure the identity of the other party (optional)
- Establish information security exchange by agreeing on encryption mechanisms where, client and server can now communicate securely.

EAP-TLS authentication handshake will be depicts as below. The client sends a random number (C) to the AS. Then AS responds by sending its certificate, cert AS, and another random number(S). If the AS wishes to authenticate the client, it also sends a certificate request message at this stage, notifying the client that it should send the client's certificate and digital signature in response.

Just receiving the certificate from the AS, the client verifies the certificate using the CA's public key. If it is valid, the client selects another random value, (P), encrypts it with the AS's public key, and sends it back to the server. This third random value is called pre-master secret and it will be used to create the session keys.

If the network requires mutual authentication, the client also sends its certificate, cert Client, along with the certificate verify message. The former contains the client's public key and the latter is the digital signature of the handshake messages signed by the client's private key, so that the AS can authenticate the client by verifying that the client knows the private key that corresponds to the public key in the certificate.

The AS and the client derive the same session key using the random numbers they exchanged and the pre-master secret. At the end of the handshake message, the AS sends TLS-Finished message, which contains the message digest of the handshake messages, including the pre-master secret. The client authenticates the AS by checking to see if the message digest that the AS sent matches the one the client computed. If the AS does not know the private key that corresponds to the server's certificate, then it would not have been able to obtain the pre-master secret and compute the same message digest as the client.

EAP-TLS provides a way to use a secure exchange for user's identity and password over it, so they will not be revealed.

EAP- TLS is well understood and well tested. EAP-TLS supports mutual authentication between the client and the AS if the client also has a certificate signed by a CA that the AS trusts.

EAP-TLS resists most attacks, including replay, dictionary and man-in the- middle attacks.

EAP-TLS also derives a per-session key between the AP and the client after successful EAP-TLS authentication.

There are some disadvantages of EAP- TLS, where the most users do not understand or use the certificates properly.

Moreover, EAP-TLS alone does not provide a way to delegate one's access to the network to others.

. Tunneled and Protected Approach

These authentication methods [15, 16] have two phases; in the first phase, the client authenticates the AS using EAP-TLS, and use the resulting session key to establish an encrypted tunnel to encrypt their communication.

In the second phase, the AS authenticates the client through that encrypted tunnel.

The major difference between tunnel method(**EAP- Tunnel Transport Layer Security**) **EAP-TTLS** and protected method **(EAP -Protected EAP) EAP -PEAP** is, while PEAP only supports any EAP methods that used by AS to authenticate the client in second phase, EAP-TTLS supports not only EAP methods but also legacy password protocols such as (Micro Soft Challenge Authentication Protocol)MSCHAP.

The encrypted tunnel in first phase has two purposes as follow:

First, it allows use of a less secure legacy protocol for client authentication by AS in the second phase without requiring a CA at the client side.

Second, using the encrypted tunnel hides the client’s identity from an eavesdropper through uses client’s EAP Response-Identity message contains a generic domain name instead of the username in first phase. When the TLS handshake is finished, the client initiates the second phase by sending his username through the encrypted tunnel.

Tunneled and protected approaches have many advantages. Not only provide identity privacy, but they can also provide delegation if authentication method that is used in the second phase provides delegation.

Moreover, even when the authentication method is vulnerable to dictionary attacks or replay attack in the tunneled second phase it becomes no longer vulnerable to these attacks because the eavesdropper sniffing the tunneled session must break the encrypted and secure EAP-TLS tunnel to mount these attacks on the client authentication.

Finally, they also derive a per-session key and resists man-in the- middle attacks.

The major flow in these methods is that requiring more power and time consumed to execute because they done in two phases.

To conclude this section, we provide comparison of authentication mechanisms discussed in this section from point of view their features in table 1 and from point of view their procedures in table 2.

***Table -1:** comparison of the authentication mechanisms from point of views their features*

	EAP-MD5	LEAP	EAP-TLS	EAP-TTLS	PEAP
Mutual authentication	No	Yes	Yes	Yes	Yes
Identity privacy	No	No	No	Yes	Yes
Replay attack resistance	No	Yes	Yes	Yes	Yes
Dictionary attack resistance	No	No	Yes	Yes	Yes
Derivation of strong session keys	No	Yes	Yes	Yes	Yes

***Table-2:** comparison of the authentication mechanism from point of views their procedures*

	EAP-MD5	EAP-LEAP	EAP-TLS	EAP-TTLS	PEAP
Server Authentication	None	Password hash	Public key (certificate)	Public key (certificate)	Public key (certificate)
Supplicant Authentication	Password hash	Password hash	Public key (certificate)	MS-CHAP(v2), EAP, CHAP	EAP
Dynamic Key Generation	No	Yes	Yes	Yes	Yes
Ease of Deployment	Easy	Hard	Hard	Moderate	Moderate
Over all Security Performance	Poor	Ok	Good	Good	Good
Software support	Multiple OS Support	Multiple OS Support	Win2000 and XP	Multiple OS Support Requires Cisco 802.11 Wireless Card	Native to Win XP

As we see the choice of the authentication, method has a strong impact on the management system. All of the more sophisticated authentication methods (EAP-TLS, EAP-TTLS, PEAP, etc.) necessitate a high infrastructure and an administration service, which complicate the network design and increase the maintenance cost. In the other hand, the weakest methods (EAP-MD5, LEAP, etc.) are easy to implement, does not require a complicated infrastructure and are simple to employ.

5. Proposed Method: EAP-MEAP

In this section, we propose a new EAP method called EAP-MEAP, which combines between advantage of using an asymmetric cipher algorithm and simplicity of using symmetric key management. It is compatible with the existing EAP protocol since it does not require any change in the 802.X/802.11 standards. The features of EAP-MEAP are illustrated as follow:

1) In EAP-MEAP, we generate a special random number at client and it used with shared secret key as inputs to one-way hash function to create a new key (kc) at output, and then sends that special random number to authentication server to create the same key (kc) as similar as at client.
2) Uses the key (kc) as a dynamic shard secret key(due to change the special random number at each access time) in the two challenge –response pairs exchanged between client and authenticator server and vice versa instead of uses static pre- shard secret key to execute mutual authentication .
3) The key(kc) works as nonce to insure freshness of mutual authentication process, where the attacker gets the key(kc)at this access time ,he has nothing to do at next access time because both client and authentication server create a new one in next access time. So that, the attacker face difficult to work as man in the middle to impersonating both client and authentication server w.r.t other and cannot tamper or eavesdrop data traffic exchanged between them if and only if he gets key (kc) at each access time. So that, data privacy /integrity is supported in mutual authentication process.
4) Uses key (kc) to encrypt client ID(user name and password) when it transfers over the air from client to authentication server instead of, it exchanges in plaintext format to achieve client ID privacy and avoid client tracking by its ID.
5) Generates a strong session encryption key by increasing the varying value inputs {S,C} for one way hash function that used individually at each side of connection to generate the session key by the new varying one called key (kc).
6) Uses a new type of random number called special random[17,18,19,20] number (rand) which hashed with pre- shared static secret to creating the key kc where, the contents of (rand) are 128 bits described as follow:

(0 –31) particular flag used to indicate that random number is (rand).

(32–47) indicate which encryption algorithm will be used to encrypt/ decrypt the data traffic (A5/1, A5/2, A5/3 ---).

(48–128) net representative of random number.

For example, if bits (0–31) are equal to the flag string, and bits (32–47) are equal to 0001000000000000, and bits (48–128) are equal to the net random number representative, then the resulting key kc only used with A5/3 algorithm not with any other A5 algorithms.

The (rand) generated at the client side so, the client decides which one of several small and weak encryption algorithms such as (A5/1 , A5/2 , A5/3----) will be used and it sends the (rand) to the authentication server side to perform it also which one of several small encryption algorithms will be used . So that ,there exist ability to change the encryption algorithm at each access time and with each session to achieve more protection(privacy and integrity) for exchanging two challenge – response pairs used for mutual authentication and data traffic which exchanged between both client ,

authentication server and vice versa against dictionary , man in the middle and reply attacks.

7) Uses a small and weak encryption algorithms , leads to reduce the number of operations to execute encryption/ decryption processing so that, save consumed power and time, instead of using one large and strong encryption algorithm that used more operations and consumed more power and time.

8) Creates a new record called session ID[21] at authentication server side after created the new dynamic shared secret key (kc) to identify this session, then the authentication server sends this session ID record to the client.

9) When session disconnected, client re-send session ID record to authentication server, which check existence of session ID record in its database. if exist, it will be re-establish that session with client using same parameters but only both generates a new one dynamic shared secret key kc. Then they go to direct to generate a new one session encryption key regardless all procedures in between, instead of execute whole protocol again to reduce consumed time and power .

10) The generation of new session encryption key will be done by hashing old one session encryption key with new one dynamic shared secret key (kc) using one way hash function.

6. How EAP- MEAP Works and Its Messages Flow

Figure-2 shows the EAP- MEAP messages flow between the client and the authentication server as follow:

1) Massage (1) → request association from client to authenticator to associate with the network.

2) Massage (2) → response association from authenticator to client to perform accepts of associate.

3) Massage (3) → request EAP -M EAP from client to authenticator to authenticate using M EAP.

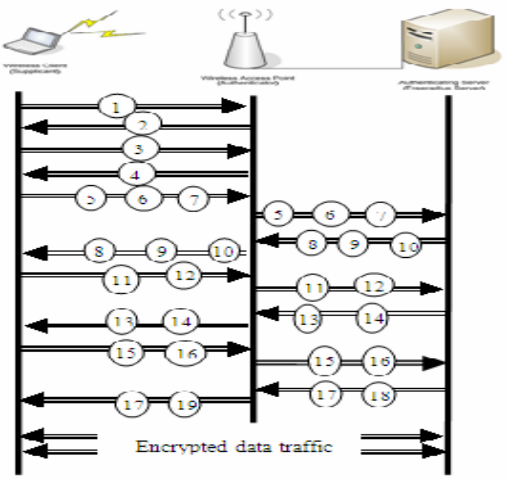

Figure-2: *shows the EAP- MEAP messages flow*

4) Massage (4) → EAP-MEAP- request (ID) from authenticator to client to perform accept authentication using EAP- MEAP by request client name and password (client ID).

After that, the client side generates special random number and creates dynamic shared secret key (kc) by hashing static pre- shared secret key with special random number using one way - hash function, then client constructs the following three messages and send them to authentication server via authenticator.

5) Massage (5) → special random number; {rand}.
6) Massage (6) → encrypted user password; {user password} Kc.
7) Massage (7) → encrypted user name; {user name} Kc.

After that, at authentication server side uses the received rand and static pre-shared secret key, to create same (kc) key by using same one way hash function , generates challenge random number (S) and also create session identify.

Note: According to contents of special rand (bit 32 up to bit 47), the client assigns the selected one encryption algorithm that will be used in mutual authentication processes with authentication server.

After that, the authentication server constructs the following three messages and sends them to client via authenticator.

8) Massage (8) → challenge random number ;(S).
9) Massage (9) → session identify ;(session ID).
10) Massage (10) → state.

At client side, received session identify ;(session ID) will be kept for re-authentication process , encrypts the random number challenge (S) by key (kc) as response to random number challenge (S) , constructs the following two messages and sends them to authentication server via authenticator.

11) Massage (11) → encrypted random number challenge (S) ;{ S} Kc.

12) Massage (12) → state.

At authentication server side, decrypts {S} Kc, and gets (S), then compares it with it is original one and sends the result to client if and only if succession, else terminates the authentication process, constructs the following two messages and sends them to client via authenticator.

13) Message (13) → EAP - M EAP successes; (success).

14) Massage (14) → state.

Note: According to the comparing result the authentication server assigns to completes the authentication process if and only if decrypted {S} Kc matching with original (S), or terminates the authentication process if and only if decrypted {S} Kc not matching with original (S).

After that, the client side generates challenge random number (C), constructs the following two messages, and sends them to authentication server via authenticator.

15) Massage (15) → challenge random number ;(C).

16) Massage (16) → state.

Then, authentication server side encrypts challenge random number (C) by (kc) key as response to random number challenge (C), computes the encryption session key by hashing all information exchanged from client to authentication server such as(S,C, ID, kc) using hash function , constructs the following two messages and sends them to authenticator only.

17) Massage (17) → encrypted random number challenge (C) ;{ C} kc.

18) Massage (18) →computed encryption session key ;(session key).

At authenticator, the received encryption session key, (session key) will kept it for encryption process of the data traffic with the client.

In addition, it constructs notification massage to perform the client to generates same encryption session key; (session key) by hashing all information exchanged from the authentication server to the client such as(S, C, ID, kc) by using same one way hash function.

Both encrypted random number challenge (C) ;{ C} kc and notification massage will be allowed to send to client.

19) Massage (17) → encrypted random number challenge (C) ;{ C} kc.

20) Massage (19) →notification massage (notification).

At client side, decrypts {C} Kc and gets (C), compares it with it is original one if and only if succession, else terminates the authentication process, computes the encryption session key by hashing all information exchanged from authentication server to client such as(S,C, ID, kc) by using same one –way hash function.

Note: According to the comparing result the client assigns to completes the authentication process if and only if decrypt {C} Kc matching with original (C), or terminates the authentication process if and only if decrypt {C} Kc not matching with original (C).

Then, the trustable secure link between client and authentication server will be established to exchanges encrypted data traffic uses encryption session key.
Note: According to contents of special rand (bit 32 up to bit 47) which pre- generated by the client, he selects the encryption algorithm will be used for encryption process, as same as one which used in mutual authentication process.
If this session disconnected, the connection will be resumed as follow:-
The message (1) up to message (7) in EAP-MEAP messages flow (figure -10) repeated again.

The client will be construct a new message (8) including (session ID) of the disconnected session and send it together messages (5, 6, 7) to authentication server via authenticator.

If and only if this (session ID) storied in the database of the authentication server, it is not generates the challenge random number (S) and it is not creates a new (session ID).

At this point, the authentication server has the client user name and password also both client and authentication server has a new dynamic shared secret key (kc) which created after message (4) at client side and after message (7) at authentication server side.

After that, the authentication server directly computes the new encryption session key by using one way hash function to hashing old one encryption session key which retrieved from authentication server data base corresponding to (session ID) with new one dynamic shared secret key (kc).

In addition, authentication server will be constructed a new message (9) including the new encryption session key and sends it to authenticator only.

At authenticator, the received new encryption session key, (session key) will kept for encryption process with client. In addition, the authenticator constructs a notification massage(10) to perform the client to generates new encryption session key; (session key) by using same one –way hash function to hashing old one encryption session key which used by client in disconnected session with new one dynamic shared secret key(kc).
Then, the trustable secure link between client and authentication server will be re-established again to exchanges encrypted data traffic uses new encryption session key.

7. New Proposal Assessments

The EAP-MEAP does not only address all the mandatory features required by the RFC 4017 [8]; it also has several advantages in comparison to other EAP methods as follow:
1) Uses the dynamic shared secret key kc instead of static pre- shared secret key to execute mutual authentication process between client and authentication server ,which work together with two random number challenges(S&C)as nonce to insure freshness of authentication process at each access time ,so achieves more robust against man in the middle and reply attacks.

2) User identity privacy will be achieved by using dynamic shared secret key kc to encrypt /decrypt the user identity that transfer from client to authentication server instead of transfer as plaintext format so, it is more difficult to track the client by its ID.
3) Dynamic shared secret key (kc) generated by hashing two inputs, one as variable value and other as fixed value (special random number and pre- shared static secret key) ,of hash function at each access time so, it is difficult to get or sniff the changeable key kc by dictionary attack methods .
4) Encryption session key will be generated by hashing all information exchanged between both client and authentication server at each side. In addition, the changeable value of the dynamic shared secret key (kc) make it is difficult to get that strong encryption session key. In addition, integrity and privacy of exchanged data traffic will be more robust against tamper or sniff process.
5) Uses one changeable small size encryption algorithm selected from several small size encryption algorithms as (A5/1, A5/2, and A5/3---) existed at both client and authentication server side for user identity privacy, mutual authentication and encryption data traffic processes at each access time. According to contents of special random number(bit 32 up to bit 47) which generated at client side and sends to the authentication server leads to ,reduce processing time to execute the protocol and save consumed power resources instead of using one fixed large strong encryption algorithm which takes more time and consumed power resources to execute.
6) Reduce the processing time and save consumed power resources for re-authentication process by creating a new record called session ID at authentication server side and sends to the client, this session ID re –send from client to authentication server to resume disconnected session again with the client ,instead of execute the whole protocol again .
7) Uses hash function in re-authentication process to generate a new encryption session key to insure integrity and privacy of exchanged data traffic between client and authentication server when resume disconnected session via hashing old one encryption session key with new one dynamic shared secret key kc instead of, uses same encryption session key of disconnected session.
8) The prediction of the implementation of random number generator, which creates that special random number at the client side to decide which one encryption algorithm will be used, will be complex due to limitations of mobile station with respect to size , power resources, and processor to execute this operation .
9) The prediction of the provisioning the mobile station with several small size encryption algorithms as (A5/1 , A5/2 , A5/3----) to select one among of them to use via protocol, will be complex due to limitations of mobile station with respect to memory size.
10) The prediction processes of the new installation pre-shared static secret key, changing the existing one, or re- installation same key are so difficult. Since the client

has to physically, take the mobile station back to the operator, if such operations could be done "Over-the-Air" these processes will be simplified and increased the client satisfaction if and only if the third party cannot deduce the value of the pre-shared static secret key.

8. AVISPA Description and Architecture

The formal verification tool is a method for proving security properties of security protocols network.

In the last decade, we have a number of verification tools like, Murphy [22], CSP [23], FDR [24], NRL protocol analyzer [25], Isabelle [26] and AVISPA [27, 30].

The Automated Validation of Internet Security Protocol and Applications tool (AVISPA) will be briefly describe as follow:

It is a push-button tool, which provides a modular and expressive formal language for specifying protocols and their security properties. It is structure integrates four different back ends as shown in figure -3 that implement a variety of state-of-the-art automatic analysis techniques.

Protocols and their intended security properties which studied by the AVISPA tool, have to be specified in HLPSL (standing for High Level Protocols Specification Language). The semantics of HLPSL is based on Lamport's Temporal Logic of Actions (TLA) [28, 29].

This language is based on roles: basic roles for representing each participant role, and composition roles for representing scenarios of basic roles finally, the environment role defines the effective principals and sessions whose execution is to consider. Each role is independent from the others, and modeled as a 'state' and each state has variables which are responsible for the state transitions, retrieves its initial information by parameters, and communicates synchronously with other roles by channel [29].

The security goals of protocol are the most important feature of this tool and they are specified in HLPSL too. These goals are used to specify secrecy and different forms of authentication to be verification by AVISPA.

Once the protocol is modeled in HLPSL, AVISPA translates it into a lower-level language Intermediate Format (IF). Intermediate Format (IF) is executed directly by one of the four back-ends tools (OFMC, CL-AtSe, SATMC or, TA4SP). For more information about the back-end, tools refer to AVISPA user manual [30].

When the analysis of a protocol has been successful, the output describes and verifies precisely if the security goals are satisfied or violated and it indicates if the protocol is safe or unsafe.

Since HLPSL is a far more expressive language than basic "Alice & Bob" notation, writing an HLPSL specification is still not an easy task.

For this reason, a new feature "Security Protocol Animator" (SPAN) was created to facilitate the specification phase by allowing the animation of the language HLPSL [31].

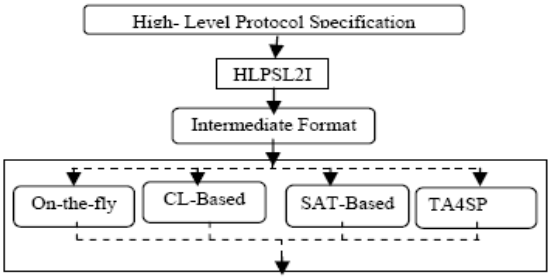

Figure -3: *Architecture of the AVISPA tool*

SPAN helps in interactively producing the Message Sequence Charts (MSC for short) which can be seen as an "Alice & Bob" trace from an HLPSL specification [32].

The CAS+ is language [33] that designed for easy specification of security protocols and leading to specifications as precise as HLSPL.

The specification of a security protocol through CAS+ comes in six parts; these parts respectively declare identifiers, message sequences, agent knowledge, session instances, and intruder knowledge and verification goals.

9. Specification and Validation EAP-MEAP

This section presents the validation results of the EAP-MEAP, obtained by using the AVISPA and SPAN tools. Since the authenticator only passes through the authentication-exchanged messages between the client and authentication server, the authenticator can be neglected in the formal verification.

EAP-MEAP protocol is defined in A (client) and B (authentication server) model.

After that, we designed the specification of EAP-MEAP protocol in the CAS+ language via six parts as displayed in pervious section and figure -4. In addition, we generated the formal language HLPSL for the specification of EAP-MEAP protocol, which used in SPAN from the CAS+ language.

The correctness of the written HLPSL code is checked using the SPAN and produced the Message Sequence Charts (MSC) of the protocol simulation.

The intruder attack simulation was done using the SPAN to check the robustness and whether it makes any abnormal flaws in the protocol run and produced the Message Sequence Charts (MSC) as shown in figure -5. The goals verification of the protocol was done with SPAN, using the back –end tool OFMC.

Figure- 6 shows the safe result of the OFMC verification for EAP-MEAP protocol. As we can see, no attacks were detected by the OFMC and all the stated security goals were satisfied.

```
protocol MEAP; % symmetric key
identifiers
A,B : user;
Hash: function;
Sessionk,Hasha,Nsessionk: function;
Na,Nb,Nc,Nd,Id,Password,Sid,Connected,Hello,Start,Meap,Ok : number;
Ks: symmetric_key;
Kd: symmetric_key;
Ke: symmetric_key;
Kc: symmetric_key;
Kn: symmetric_key;
messages
1.A->B: Meap
2.B->A:Hello
3. A ->B : {Id,Password}(Hash((Ks),Nc)),Nc
4. B ->A : {Na}(Hash((Ks),Nc))
5. A ->B : {Na,Nb}(Hash((Ks),Nc))
6. B ->A : {Nb}(Hash((Ks),Nc))
7. A -> B : {Start}(Sessionk((Ks),(Hash((Ks),Nc))))
8. B -> A : {Start,Sid}(Sessionk((Ks),(Hash((Ks),Nc))))
9.A ->B : {Sid,Nd} (Sessionk((Ks),(Hash((Ks),Nc))))
10.B -> A :{Connected}(Nsessionk((Ks),(Hasha((Ks),Nd))))
11.A ->B :{Connected,Ok}(Nsessionk((Ks),(Hasha((Ks),Nd))))
knowledge
A :B,Ks,Id,Password,Meap;
B :A,Ks,Hello;
session_instances
[A:client,B:server,Hash:kd,Ks:ks,Start:start,Ok:ok,Id:id,Password:password,Connected:connected,Sessionk:ke,Hasha:kc,Nsessionk:kn,Sid:sid];
intruder_knowledge
client, server;
goal
B authenticates A on Na;
A authenticates B on Nb;
secrecy_of Password [ A,B];
secrecy_of Id [ A,B];
secrecy_of Sid [];
secrecy_of Start [ ];
secrecy_of Ok[A,B];
secrecy_of Connected [];
```

***Figure-4:** Designed specification EAP-MEAP protocol in the CAS+ language*

We assume that the client and the authentication server had a pre-shared key (Ks) in advance.

In addition, the client generates two random numbers (NC, Nd), generates a nonce value (Nb) while the authentication server generates a nonce value (Na) and they used the same four one-way hash functions: Hash, Sessionk, Hasha, and Nsessionk .

The CAS+ code of the goals instructions are verified as follow:

1) Identity Privacy:

The identity privacy of the client was done by instruction {Id,Password}(Hash((Ks) ,Nc)), Nc where,(Id, Password) is the client identity which encrypted by dynamic shared secret key kd= (Hash((Ks),Nc)).

2) Mutual Authentication:

The mutual authentication between the authenticator server and the client was done by using the two encrypted nonce values (Na, Nb) by the dynamic shared secret key kd, via exchanging the three instructions between them in sequence ,as follow{Na}(Hash((Ks),Nc)), {Na,Nb}(Hash((Ks),Nc)) and {Nb}(Hash((Ks),Nc)). In addition, at each side checking the coincidence of the generated nonce value with the received one, which proves the mutual authentication if and only if they are coincident.

3) Strong key secrecy:

The secrecy and the strong encryption key Ke(dynamic key) was done by calculating it at each side from the instruction (Sessionk((Ks),(Hash((Ks),Nc)))) by using a new

value of random number Nc for each new session , which asserts that the key Ke is strong and kept secret between the authenticator server and the client.

4) Attack robustness:

The roles of using (one time use) both the dynamic shared secret key Ke, changeable two random numbers (NC,Nd) and changeable two nonce values (Nb Na) for each new session of authentication and protected traffic exchanged processes.

In addition using four one-way hash functions permit to detect and overcome several types of attacks such as Man in the Middle, dictionary and replay attacks.

5) Fast re-authentication:

Fast re-authentication was done by creating a new record called session ID (Sid) at authentication server side to identify this session, then the authentication server sends (Sid) to the client by using the instruction

{Start, Sid}(Sessionk((Ks),(Hash((Ks),Nc)))) that encrypted by key Ke.

In case of that, session disconnected, the client re-sends (Sid) and sends the new random number (Nd) to the authentication server via the instruction

{Sid, Nd}(Sessionk((Ks),(Hash((Ks),Nc))))that encrypted by key Ke.

The authentication server check the existence of session ID record(Sid) in its database , if exist, it will be re–establish that session with client with same parameters but only both of them generates a new one dynamic shared secret key kc from the instruction (Hasha((Ks),Nd)). Then, they go to direct to generate a new one session encryption key Kn from the instruction (Nsessionk((Ks),(Hasha((Ks),Nd)))).

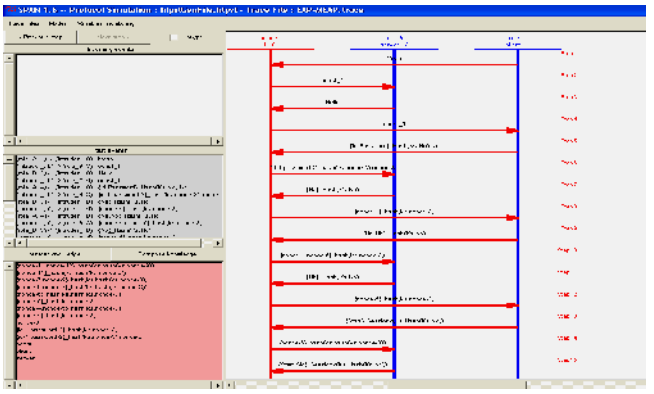

Figure-5: *EAP-MEAP protocol intruder attack simulation by SPAN*

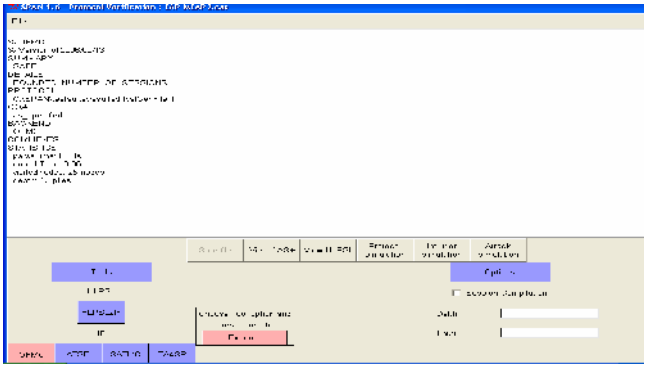

Figure-6: *Verification EAP-MEAP protocol by OFMC*

10. CONCLUSION AND FUTURE WORK

The EAP protocol gives dynamicity and flexibility to the wireless networks. However, the existed EAP methods do not offer the expected properties for a secure authentication and easy implementation. In this paper, we proposed a new EAP method called EAP-MEAP, which offers interesting properties of fast and mutual authentication, and robustness to man- in the- middle, reply and dictionary attacks, and provides a strong key.

The proposed method EAP-MEAP provides simplicity of use and can be deployed inside wireless networks without changing in the existed network hardware.

EAP-MEAP supports client identity privacy to protect it from tracking.

EAP-MEAP supports dynamic shared secret key and changeable values of both random numbers and nonce has to insure freshness of mutual authentication process and generates encryption session key at each access time.

EAP-MEAP supports a new record called session ID used to speed up resume of authentication session in case of session disconnects.

Many points can be tackled to improve our proposed protocol; in particular, we plan to study provisioning of the pre-shared static secret key at client side via “Over-the-Air” using secure technology.

Moreover, the implementation of random number generator to create the special random numbers and nonce's is a critical issue and needs more investigations.

In addition, the implementation of large memory size to store several small encryption algorithms at the client side is a challenge, especially with regard to limitations of mobile station.

Finally, using EAP-MEAP will be recommended to increase both security of mutual authentication process and protection of exchanged data traffic in WLAN.

References

[1] [S3-030689] Technical Specification: "Wireless Local Area Network (WLAN) Interworking Security", 3GPP TSG SA WG3 Security meeting, 31 November 2003.
[2] Monis Akhlaq, Baber Aslam, Muzammil A Khan and M noman Jafri, "Comparative Analysis of IEEE 802.1x Authentication Methods", Proceedings of the 11th WSEAS International Conference on communications, July2007,PAKISTAN.
[3] John Vollbrecht and Robert Moskowitz, "Wireless LAN Access Control and Authentication" © 2002 Interlink Networks, Inc. Revision D. 12-02.
[4] Sean Convery, Darrin Miller, Sri Sundaralingam, Mark Doering, Pej Roshan, Stacey Albert, Bruce McMurdo and Jason Halpern, "Wireless LAN Security in Depth", Cisco Systems Inc., September 2005.
[5] Kwang-Hyun Baek, Sean W. Smith and David Kotz , "A Survey of WPA and 802.11i RSN Authentication Protocols", Dartmouth College Computer Science, November 2004.
[6] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson and H.Levkowetz, Extensible Authentication Protocol , RFC3748, June 2004.
[7] B. Aboba, D. Simon, RFC 2716, PPP EAP TLS Authentication Protocol, the Internet Society, Oct. 1999.
[8] D. Stanley, J. Walker and B. Aboba, Extensible Authentication Protocol Method Requirements for Wireless LANs, RFC 4017, March 2005.
[9] L. Blunk and J. Vollbrecht, PPP Extensible Authentication Protocol, Merit Network, RFC 2284 March 1998.
[10] Younes El Hajjaji El Idrissi , Noureddine Zahid and Mohamed Jedra, A New EAP Authentication Method for IEEE 802.11 Wireless, IJCSNS, VOL.11 No.6, June 2011,
[11] H. Hwang, G. Jung, K. Sohn and S. Park, A Study on Man in the Middle Vulnerability in Wireless Network Using 802.1X and EAP, International Conference on Information Science and Security , Seoul, Korea, 2008, pp. 164-170.
[12] R. Dantu, G. Clothier and A. Atri, EAP methods for wireless networks, Computer Standards Interfaces 29 (3) (2007) 289–301.
[13] Cameron Macnally, "LEAP protocol description", Cisco , August 2001.
[14] T. Dierks and C. Allen, The TLS Protocol Version 1.0, IETF ,RFC 2246, January 1999.
[15]"EAP Methods for 802.11 Wireless LAN Security", International Engineering Consortium ,September 2005, http://www.iec.org.
[16] Michel Barbeau and Lei Han ,"A Threat Analysis of The Extensible Authentication Protocol", Carleton University. ,April 2006.
[17] [S3-040030] Technical Specification: "Introducing the special RAND mechanism", 3GPP TSG SA WG3 Security meeting #32, Scotland, February 2004.
[18] [S3-030651] Technical Specification: " Further development of the Special RAND mechanism", 3GPP TSG SA WG3 Security meeting #30; October 2003.

[19] [S3-030693] Technical Specification:"More elements on the Special RAND mechanism", 3GPP TSG SA WG3 Security meeting #31, Germany, November 2003.
[20] [S3-040529] Technical Specification: "Introducing the special rand mechanism as a principle for GSM/GPRS ", 3GPP TSG SA WG3 Security meeting #34, July2004.
[21] Narayanan, V. and L. Dondeti,"EAP Extensions for EAP Re-authentication Protocol (ERP)", RFC:5296 /(IETF),August 2008.
[22] J.C Mitchell, M. Mitchell and U. Stern, "Automated Analysis of Cryptographic Protocols Using Murphi", IEEE Symposium on Security and Privacy, IEEE Computer Society Press (1997) 141-151.
[23] S. Schneider, "Verifying Authentication Protocols in CSP", IEEE Transactions on software engineering, 24 (1998) 741–758.
[24] A. W Roscoe, " Modelling and verifying key exchange protocols using CSP and FDR", In Proceedings of 8th IEEE Computer Security Foundations Workshop, IEEE Computer Society Press, County Kerry, 1995.
[25] C. Meadows, "The NRL Protocol Analyzer: an overview", Journal of Logic Programming, pp. 113–131 ,26(1996).
[26] University Of Cambridge, http://www.cl.cam.ac.uk/research/hvg/Isabelle/overview.html, updated 12-07 2006.
[27] A. Armando, D. Basin, J. Cuellar, M. Rusinowitch and L. Viganò, "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications", CAV 2005, LNCS 3576, 2005, pp 281– 285.
[28] L. Lamport, "The temporal logic of actions. ACM Transactions on Programming Languages and Systems", , May 1994 872-923.
[29] Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drieslma, J. Mantovani, S. Modersheim, L. Vigneron. "A high level protocol specification language for industrial security-sensitive protocols". In Proceedings of Workshop on Specification and Automated Processing of Security Requirements (SAPS 2004),2004.
[30] Y. Glouche and T. Genet, "SPAN – a Security Protocol Animator for AVISPA – User Manual". IRISA / Universit´e de Rennes 1, span, 2006 http ://www.irisa.fr.
[31] Y. Boichut, T. Genet, Y. Glouche and O. Heen, "Using animation to improve formal specifications of security protocols", In Proceedings of SAR-SSI'07, 2007.
[32] D. Harel and P. S. Thiagarajan. "Message sequence charts. UML for Real : Design of Embedded Real-time Systems", 1 ed, 2003.
[33] Saillard and Thomas Genet , "CAS+ Manual", Ronan, March 21, 2011.
[34] Lianfen Huang , Ying Huan and Zhibin Gao," Performance of Authentication Protocols in LTE Environments", Int. Conference on Computational Intelligence and Security China,2009.
[35] Jyh-Cheng Chen, and Yu-Ping Wang," Extensible Authentication Protocol (EAP) and IEEE 802.1x", IEEE Communications Magazine, 2005.