Improving the security of EAP-EHash authentication method

AIT HEMAD Miloud[1] — EL KIRAM Moulay Ahmed[2] — LAZREK Azzeddine[3]

University Cadi Ayyad,
Faculty of Sciences SEMLALIA - Department of Computer Sciences
Bd. Prince My Abdellah, B.P. 2390, 40000 Marrakech, Morocco
[1]m.aithemad@ucam.ac.ma, [2]Kiram@ucam.ac.ma, [3]lazrek@ucam.ac.ma

***Abstract.**, Several EAP authentication methods have recently been proposed to ensure authentication in 802.11 wireless networks. Most of these methods require high computation cost for the purpose of security. However, they are not suitable for use in a resource constrained network environment. EAP-EHash is one of the very few interesting methods, which were designed for use in such environment. EAP-EHash requires low computation cost and combines simplicity and security. In this paper, we show that EAP-EHash suffers from serious security defects. Then, we propose improvements to correct these flaws.*

***Résumé.** Récemment, plusieurs méthodes d'authentification EAP (Extensible Authentication Protocol) ont été proposées pour assurer l'authentification dans les réseaux sans fil 802.11. Parmi ces intéressantes propositions, figure EAP-EHash, qui a été conçue pour allier la simplicité et la sécurité. Dans ce papier, nous montrerons que EAP-EHash souffre de défauts importants. Ensuite, nous allons proposer des améliorations afin de combler ces lacunes.*

***Keywords:** Security, Authentication, Wireless network, 802.1X, EAP.*

***Mots clés** : 802.1x, Sécurité, Réseau sans fil, 802.1X, WLAN, EAP.*

*Amélioration de la sécurité de la méthode d'authentification EAP-EHash

1. Introduction

Since their appearance, the wireless local area networks (WLANs) (Gast, 2005) have been regarded as a simple extension of wired LANs. However, the increasing need to communicate freely and wirelessly made these networks boom, and they gradually increased their competitiveness compared to wired networks. However, they are very vulnerable to malicious eavesdropping and therefore very sensitive to security issues. Indeed, to transmit information over radio waves facilitates their interception by intruders.

To face the low level of wireless LANs security and to make it similar to that of wired LANs, the IEEE 802.11 working group has defined the WEP (Wired Equivalent Privacy) mechanism (Bulbul, Batmaz, Ozel, 2008). Unfortunately, WEP suffers from a number of weaknesses (Miller, Hamilton, 2002; Bulbul, Batmaz, Ozel, 2008). The new standards; specially 802.1X (IEEa, 2004a) and 802.11i (IEEa, 2004b) were put forward to correct these defects.

The 802.1X standard (Synder, 2002; IEE, 2004a) provides authentication, access control and key management. Note that 802.1X is also used in the 802.11i standard for authentication.

The principle of 802.1X is as follows: if a client, who is also known as supplicant, wants to connect to the network, the access point, that also called authenticator, blocks all traffic except the one related to the authentication process until he is authenticated to the authentication server. Often, the authentication server is RADIUS server (Remote Authentication Dial-In User Service) (Aboba, Calhoun, 2003).

802.1X is based on EAP (Extensible Authentication Protocol) (Aboba and al. 2004; Aboba, Simon, Eronen, 2004), defined by the IETF (Internet Engineering Task Force) to ensure authentication. This protocol specifies a generic framework for multiple authentication methods. These methods define authentication schemes and key distribution.

The authentication method used is transparent to the access point; only the client and authentication server use it. The access point simply relays EAP messages between the client and authentication server, except the last one which represents the result of authentication (success or failure). Following this result, the access point allows or blocks the client to access to the network. The EAP messages exchanged between the mobile station and access point are carried in EAPOL (EAP over LAN) frames. And the ones exchanged between the access point and authentication server are carried in EAPOR (EAP over RADIUS) frames.

EAP supports several authentication methods (Dantu, Clothier, Atri, 2007), including: EAP-MD5 (Aboba and al. 2004), EAP-TLS (Aboba, Simon, 1999), EAP-

TTLS (Funk, Blake-Wilson, 2004), EAP-LEAP and EAP-PEAP (Palekar and al. 2003).

802.1X does not impose any specific authentication method. This has triggered many researches and works to develop a robust, secure and fast one. Among the interesting recent methods proposed is EAP-EHash (Cheikhrouhou, Ben Jemaa, Laurent-Maknavicius, 2006; Cheikhrouhou and al. 2009), which combines speed, efficiency and mutual authentication.

The rest of this paper is organized as follows. A review of EAP-Hash authentication method is described in Section 2. In Section 3, we show the security flaws of EAP-EHash we found and we present our proposed improvements. In Section 4, we shall analyze the proposed improvement scheme. Finally, Section 5 presents the conclusion.

2. Review of EAP-EHash authentication method

The EAP-EHash method (Cheikhrouhou and al. 2009) has been proposed to be used in a resource constrained network environment. That's why EAP-EHash reduces the cryptographic load required for mutual authentication, by using the symmetric cryptography, challenge-response authentication mechanism and one-way hash function.

In EAP-EHash, only one secret key PSK (Pre-Shared Key) need to be shared between the client and authentication server. This key is used to derive two different keys AK and EK. The client and the authentication server must have these two keys to prove their identities. So, this mitigates dictionary and force attacks, and then enhances security. Indeed, to impersonate a legitimate entity, the intruder must have two keys and not just one.

EAP-EHash consists of three phases (Figure 1): negotiation phase, authentication phase and key derivation phase.

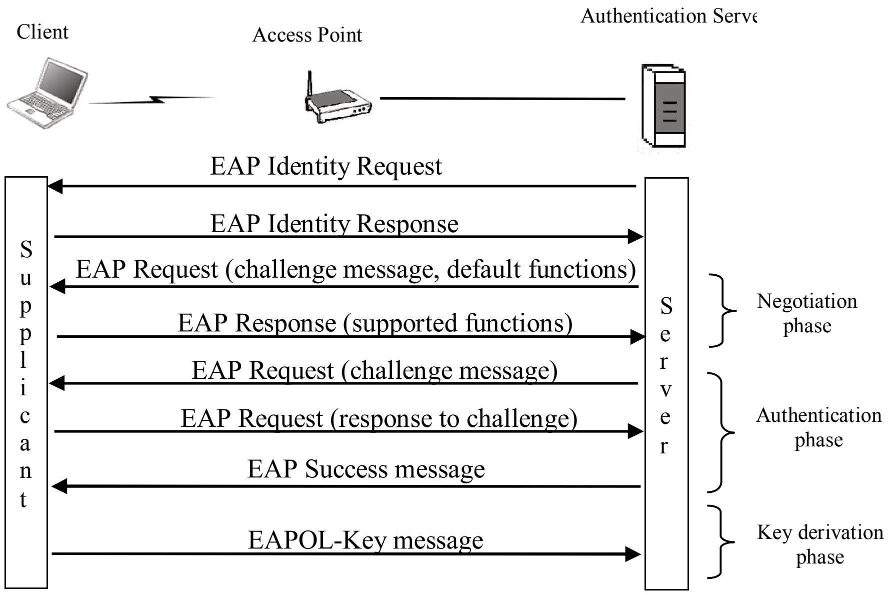

Figure 1: *EAP-EHash phases*

2.1. Negotiation phase

The negotiation phase allows the client and the authentication server to negotiate the ciphersuite which will be used during authentication phase. The ciphersuite consists of the hash function and the encryption algorithm.

As shown in Figure 1, after the request of the authenticator, the client shows his identity in an EAP message to access point which relays it to the authentication server. During all exchanges EAP, the role of access point is limited to relay EAP messages between the client and authentication server. The Negotiation phase starts with the authentication server sending its configured default ciphersuite to client. These data consist a field, called "Algo". After, the client checks the "Algo" field. In case of success, the client responds with EAP-Response message and the negotiation phase is done. Otherwise, another case is described into details in (Cheikhrouhou and al. 2009).

2.2. Authentication phase

The authentication phase (Figure 2) begins with the authentication server sending to client a message that contains two random numbers Challenge and RandS, server's identity ServerID, ciphersuite (Algo) and the result of encryption of MIC using key EK and the symmetric algorithm. The MIC (Message Integrity Check) is a hash value calculated as follows: MIC1 = F (AK, Challenge || ServerID || RandS || Algo) such:

- F denotes a one-way hash function (Schneier, 1996) (such as HMAC-SHA-1 or HMAC-MD5).
- AK (Authentication Key) and EK (Encryption Key) are two session keys derived from PSK as AK = F(PSK, RandS) and EK = F(PSK, RandS || ServerID || ClientID) where || denotes the concatenation and ClientID denotes the client's identity.

Then, the client calculates the same keys as the server. Then, he computes the MIC and compares it with the one received. If they match, the authentication server is authenticated to the client. Then, the client sends the server a random number RandC combined with the ciphersuite (Algo) and the result of encryption of Hash using key EK and the symmetric algorithm. The Hash is a hash value calculated as follows: Hash = F(AK, Challenge || RandC || Algo).

In turn, the server calculates the Hash and compares it with the one received. In case of success, the client is authenticated to the authentication server.

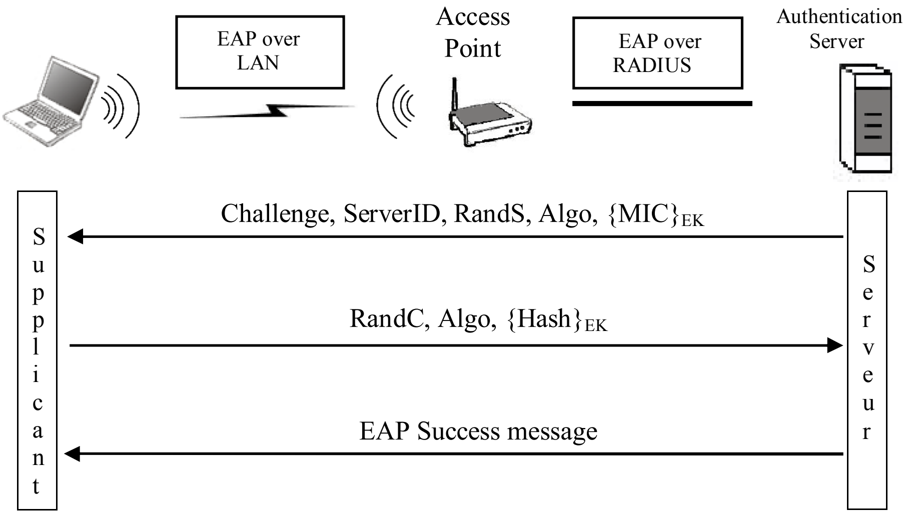

Figure 2: *EAP-EHash authentication phase*

2.3. Key derivation phase

The key derivation phase (Cheikhrouhou, 2009) allows the client and the access point to agree on a session key PTK. The goal is to secure the traffic exchanged over radio link.

3. Vulnerabilities and proposed improvement of EAP-EHash

The authentication phase of EAP-EHash presents two major security flaws:

– Vulnerability to replay attack.

– Vulnerability to known key attack.

3.1. Vulnerability to replay attack

EAP-EHash is an EAP method that is based on challenge-response authentication mechanism. In such a mechanism, the verifier entity sends a random number, it is the challenge. Using a shared secret, the entity, who claims authentication, must respond according to the challenge. This mechanism is used mainly to ensure the freshness of exchanged messages and thus to counter the replay attack which consists of replaying some old messages intercepted during an earlier communication.

EAP-EHash is vulnerable to replay attack. Indeed, the client authenticates the server before it sends a challenge. So, the challenge-response mechanism is not properly implemented in the client side. This permits any intruder, who leads this type of attack, to impersonate a legitimate server to the client.

To carry out this attack, what the intruder has to do is only intercept the first message during an earlier communication between the client and the authentication server. Once he has this message, the intruder can easily impersonate the server by sending this message to the client.

In EAP-EHash, before authenticating the server, the client must verify the MIC, which contains the challenge of server but not his. To authenticate the client, the server must check the Hash which contains the client's challenge.

To counter replay attack, we propose reversing the process, that is to say, the MIC will be used to authenticate the client and the Hash to authenticate the authentication server. Our proposed improvement of EAP-EHash authentication phase is shown in Figure 3.

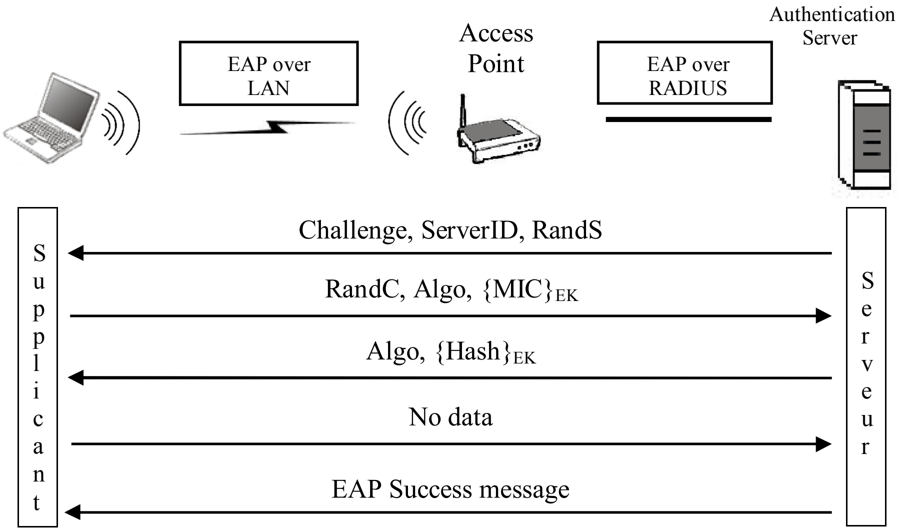

Figure 3: *Our proposed improvement of EAP-EHash authentication phase*

3.2. Vulnerability to known key attack

As session keys are stored in memory during the session, they are relatively easier to be compromised than the Pre-shared keys. The known key attack consists of reusing an old compromised session key for impersonation in a new session.

Using only the challenge generated by the authentication server when calculating two keys AK and EK, makes the EAP-EHash vulnerable to known key attack. Indeed, if an intruder can successfully compromise the keys AK and EK in a given session, he may use these two keys as often as he wishes, in order to impersonate a legitimate authentication server to the client, provided that he uses the same challenge RandS selected during the compromised session.

To correct this defect, we propose to use also the client's challenge RandC when calculating keys AK and EK. Thus, these keys will be computed as follows:

AK = F(PSK, RandS || RandC) and EK = F(PSK, RandS || RandC || ServerID || ClientID).

In these two attacks, the intruder does not obtain key PTK used for encryption of messages exchanged between the client and access point. So, the goal of these attacks is not to allow the intruder to decrypt the exchanged messages, but to mislead the client by making him think he is connected to a legitimate server and so to a legitimate network while it is not the case.

4. The security analysis of the proposed improvement

We have shown in the previous section that EAP-EHash has two serious weaknesses about the authentication of authentication server. These flaws are due essentially to inappropriate use of the challenge-response mechanism.

At first, we admit that the challenge-response mechanism permits to ensure the freshness of exchanged messages. The correct use of client's challenge for authentication of server makes it infeasible for an intruder to lead the replay attack against EAP-EHash for impersonating as valid authentication server. Same, the use of client's challenge when computing session keys avoids known key attack against EAP-EHash. Our proposed improvements based on the use of client's challenge while the authentication of server and the calculation of session keys. As a result, our proposed improvements can certainly repair the security flaws of EAP-EHash.

5. Conclusion

The EAP-EHash is an interesting proposal, as it greatly reduces the cryptographic load required by most EAP authentication methods and it provides mutual authentication. Despite its merits, EAP-EHash, as proposed by its authors, has two serious security weaknesses about the authentication of authentication server.

The authors of EAP-EHash claim that their method resists attacks. We have shown that EAP-EHash is vulnerable to known key attack. We have also shown that any intruder, who eavesdrops on the network, can replay messages intercepted during a prior communication and therefore impersonate a legitimate server. Furthermore, we have proposed the improvements to counter such attacks. These proposed improvements make EAP-EHash more secure and robust.

6. Références

Aboba, B. et al. 2004. Extensible Authentication Protocol (EAP). «IETF RFC 3748», June 2004

Aboba, B., Calhoun, P. 2003. RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP). «IETF RFC 3579». September 2003

Aboba, B., Simon, D. 1999. PPP EAP TLS Authentication Protocol. «RFC 2716»

Aboba, B., Simon, D., Eronen, P. 2008. Extensible Authentication Protocol (EAP) Key Management Framework. «IETF RFC 5247», August 2008

Bulbul, H., Batmaz, I., Ozel, M. 2008. Wireless network security: comparison of WEP (Wired Equivalent Privacy) mechanism, WPA (Wi-Fi Protected Access) and RSN (Robust Security Network) security protocols. Proceedings of the 1st international conference on «Forensic applications and techniques in telecommunications, information, and multimedia and workshop, Article 9», January 2008

Cheikhrouhou, O. et al. 2009. An EAP-EHash Authentication Method Adapted to Resource Constrained Terminals. *Annales des Télécommunications*. (Engineering Collection)

Cheikhrouhou, O., Ben Jemaa, M., Laurent-Maknavicius M. 2006. Nouvelle méthode d'authentification EAP-EHash. «12ème Colloque Francophone sur l'Ingénierie des Protocoles - CFIP»

Convery, S., Miller, D., Sundaralingam, S. 2003. *Cisco SAFE: Wireless LAN security in depth*. Cisco Systems

Dantu, R., Clothier, G., Atri, A. 2007. EAP methods for wireless networks. *Computer Standards & Interfaces*. March, 29(3). p. 289-301

Funk, P., Blake-Wilson, S. 2004. EAP Tunneled TLS Authentication Protocol (EAP-TTLS). «draft-ietf-pppext-eap-ttls-05.txt, internet draft», July 2004

Gast, M. S. 2005. 802.11 wireless network. O'Reilly

IEEE 802. 1X-2004. 2004. IEEE Standard for Local and Metropolitan Area Networks Port-Based Network Access Control. «IEEE, Piscataway»

IEEE 802.11i standard. 2004. LAN/MAN Specific Requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Specification for Enhanced Security

Miller, B.R., Hamilton, B.A. 2002. Issues in Wireless Security WEP, WPA& 802.11i. Proceedings of the «18th Annual Computer Security Applications Conference», December 2002

Palekar, A. et al. 2003. Protected EAP Protocol (PEAP) version 2. «draft-josefsson-pppext-eap-tls-eap-07.txt, internet draft», October 2003

Schneier, B. 1996. Applied Cryptography. Second edition, John Wiley & Sons.

Snyder, J. 2002. What is 802.1X?. Network World Global Test Alliance. [en ligne]. May 2002. Disponible à l'adresse : www.networkworld.com