Nouvelle méthode d'authentification EAP-EHash

Omar Cheikhrouhou*, Maryline Laurent-Maknavicius, Maher Ben Jemaa***

** Unité de recherche ReDCAD, Ecole Nationale d'Ingénieurs de Sfax, BP W 3038 Sfax, Tunisie*
Enis01amor@yahoo.fr, maher.benjemaa@enis.rnu.tn
*** CNRS Samovar UMR 5157, GET/INT/LOR, 9 rue Charles Fourier, 91011 Evry, France*
Maryline.Maknavicius@int-evry.fr

RÉSUMÉ. *Cet article présente une nouvelle méthode EAP appelée EAP-EHash (EHash pour Encrypted-Hash) qui allie la simplicité et la facilité de déploiement de EAP-MD5 et la robustesse de EAP-TLS. Elle se positionne dans les mêmes domaines d'application que ces deux méthodes EAP existantes (authentification d'utilisateurs à des serveurs, VPN…). Pour un meilleur positionnement de ces méthodes, l'article présente les deux méthodes EAP-TLS et EAP-MD5, puis EAP-EHash et effectue pour chacune une analyse critique. Enfin, la méthode EAP-EHash est validée formellement à l'aide de l'outil AVISPA, ce qui prouve qu'elle vérifie bien les propriétés de sécurité désirées, en particulier sa robustesse à l'attaque Man-In-The-Middle.*

ABSTRACT. *This paper describes a new EAP method called EAP-EHash (EHash for Encrypted-Hash) which combines simplicity and deployment easiness of EAP-MD5 and robustness of EAP-TLS. EAP-EHash is expected to apply in the same application domains than those two existing methods (user authentication to servers, VPN…). For a better positioning of each EAP method, the paper presents first the EAP-TLS and EAP-MD5 methods, and then EAP-EHash and realizes a critical analysis of each of them. Finally the EAP-EHash method is formally validated with the AVISPA tool, thus proving that it verifies the expected security properties, especially its robustness to Man-In-The-Middle attacks.*

MOTS-CLÉS : *Authentification, EAP, EAP-MD5, EAP-TLS, EAP-EHash*

KEYWORDS: *Authentication, EAP, EAP-MD5, EAP-TLS, EAP-EHash*

1. Introduction

Le succès du concept EAP *(Extensible Authentication Protocol)* est d'avoir introduit la distinction entre le protocole EAP (Aboba, 2004) et les méthodes EAP. Le protocole EAP se résume à encapsuler les données servant à l'authentification et les méthodes EAP prennent en charge l'authentification en mettant en forme et en interprétant ces données d'authentification. Il ressort de cette séparation que les protocoles faisant appel à une authentification par EAP ne sont plus attachés à une méthode EAP particulière. Ainsi, en cas de failles de sécurité découvertes sur une méthode EAP, il suffit de changer de méthode EAP ; les protocoles s'appuyant sur EAP sont conservés.

Ce concept est né du besoin rencontré dans les environnements PPP (Point-to-Point Protocol) et 802 d'intégrer plusieurs méthodes d'authentification. L'objectif était d'introduire de la flexibilité dans l'authentification réalisée par les utilisateurs pour se connecter à leur réseau d'accès. Il en a découlé en particulier le protocole 802.1X qui permet à un point d'accès 802.11 (Access Point) d'authentifier les mobiles avant de leur donner accès à une infrastructure de réseau. Notons que EAP était à l'origine prévu pour fonctionner au niveau de la couche 2 (modèle OSI).

Aujourd'hui, grâce à la définition du protocole PANA (Protocol for carrying Authentication for Network Access) qui encapsule EAP au dessus de la couche IP, l'authentification d'un utilisateur auprès d'un réseau d'accès est de niveau applicatif. Cette solution EAP/PANA plaît beaucoup aux opérateurs car elle est indépendante de la technologie choisie pour l'accès (802.11, PPP, Ethernet, WiMax…) et facilite le déploiement des solutions de contrôle d'accès.

A ce jour, plus d'une quarantaine de méthodes EAP existent, mais seulement six d'entre elles sont standardisées à l'IETF (Internet Engineering Task Force). Parmi ces dernières, les méthodes EAP-MD5 (Aboba, 2004) et EAP-TLS (Aboba, 1999) sont les plus simples de mise en œuvre mais elles souffrent de plusieurs inconvénients qui les rendent difficiles d'exploitation dans les réseaux IP classiques.

Cet article est organisé de la façon suivante. La section 3 donne une description des méthodes EAP-MD5 et EAP-TLS suivie d'une analyse critique. La section 4 introduit une nouvelle méthode EAP-EHash qui allie la simplicité de EAP-MD5, et certaines propriétés de EAP-TLS comme l'authentification mutuelle et la résistance à certaines attaques classiques. Après une analyse des propriétés de sécurité apportées par EAP-EHash (cf. section 5), la section 6 présente les résultats de validation à l'aide de l'outil AVISPA. Enfin la section 7 fait une synthèse comparative des différentes méthodes EAP.

Pour une illustration pratique de ces méthodes, nous nous placerons dans la suite dans un contexte de réseau IEEE 802.11 et nous considèrerons qu'un utilisateur veut accéder à une infrastructure de réseau IP et procède à une authentification. Les éléments de réseau considérés sont présentés à la section 2.

2. Contexte IEEE 802.11 servant à illustration des méthodes EAP

Sur les illustrations proposées dans la suite de l'article, trois types d'équipements sont considérés :

– le client EAP intégré dans le mobile permet au réseau d'accès d'authentifier le mobile avant de lui donner accès à ses ressources réseau ;

– l'authentificateur EAP est l'équipement d'accès au réseau (point d'accès 802.11) qui bloque le trafic en provenance du mobile jusqu'à ce que le mobile s'authentifie et à condition qu'il dispose des bonnes autorisations. L'authentificateur n'interprète pas le contenu des messages EAP, mais il relaie les requêtes d'authentification au serveur EAP par exemple sous la forme de messages RADIUS (Remote Authentication Dial-In User Service) ;

– le serveur EAP en général intégré dans un serveur AAA (comme RADIUS) réalise l'authentification du mobile et parfois s'authentifie.

Notons que les messages EAP entre client et authentificateur sont encapsulés dans le protocole 802.1X et que les messages EAP sont échangés entre authentificateur et serveur EAP grâce au protocole RADIUS qui peut encapsuler EAP (Aboba, 2003).

3. Description et analyse critique des méthodes EAP-MD5 et EAP-TLS

3.1. *EAP-MD5*

La méthode EAP-MD5 (Aboba, 2004) se base sur le protocole CHAP (Challenge Handshake Authentication Protocol) (Simpson, 1996) qui vise à authentifier un client en utilisant le principe de défi-réponse. EAP-MD5 nécessite une clé partagée entre client et serveur d'authentification. Cette clé consiste généralement en un mot de passe associé à un nom d'utilisateur ou à une identité (par exemple une adresse IP ou MAC).

Comme l'illustre la figure 1, après requête de l'authentificateur EAP (étape 1), le client décline son identité (étape 2) dans un message EAP qui est relayé au serveur EAP. Ce dernier envoie un défi aléatoire ou challenge au client (étape 3) qui calcule un hash à partir de ce défi et de la clé partagée avec le serveur. Le hash est retourné dans un message EAP (étape 4). Le serveur effectue le même calcul de hash que le client et compare les deux hash. Deux hash identiques signifient que le client possède la bonne clé, ce qui conduit au succès de l'authentification et à l'émission d'un message d'acceptation (étape 5a). Sinon, l'authentification échoue et le serveur rejette la demande (étape 5b). En fonction de cette décision, l'authentificateur autorise ou non le client à accéder au réseau.

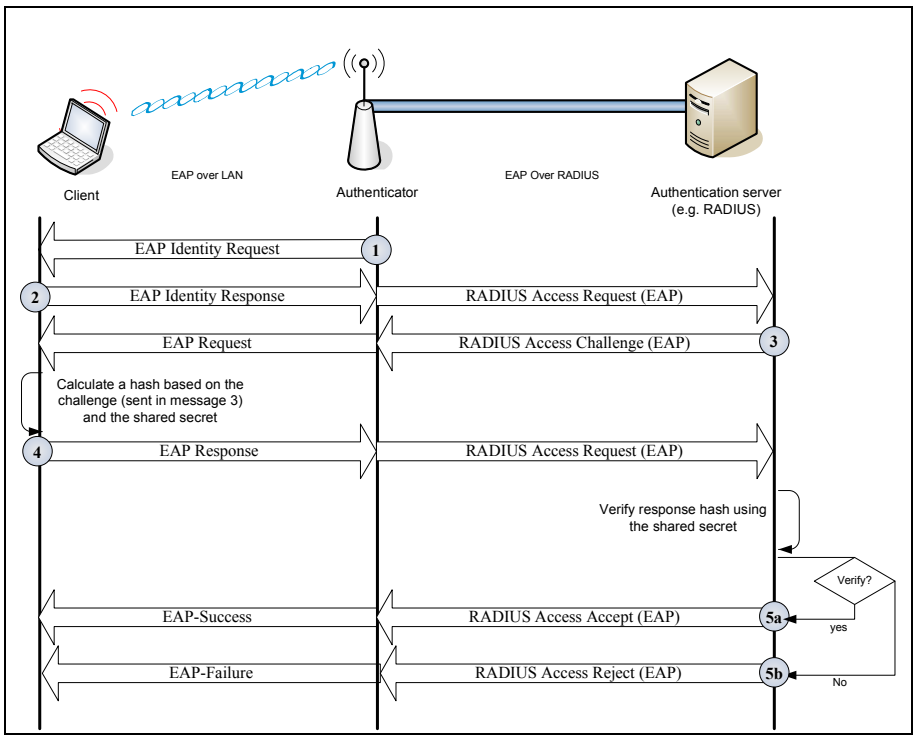

Figure 1. *Échanges EAP-MD5 dans un contexte IEEE 802.11*

3.2. *EAP-TLS*

La méthode EAP-TLS (Aboba, 1999) est issue du protocole TLS (Transport Layer Security) (Dierks, 1999) qui vise à protéger les échanges de données sur Internet en mettant en œuvre l'authentification mutuelle des participants, la confidentialité des données, leur intégrité et l'authentification de leur origine. Pour cela, elle encapsule certains messages TLS dans des enregistrements TLS. De plus, confrontée à la longueur importante des messages obtenus, EAP-TLS gère elle-même la fragmentation et réassemblage.

Comme l'illustre la figure 2, après avoir détecté la présence d'un nouveau mobile, l'authentificateur initie l'authentification EAP-TLS en demandant l'identité du client. La réponse est ensuite relayée par l'authentificateur jusqu'au serveur EAP. D'ailleurs, dans toute la suite des échanges EAP, l'authentificateur ne fait que relayer les messages EAP entre client et serveur.

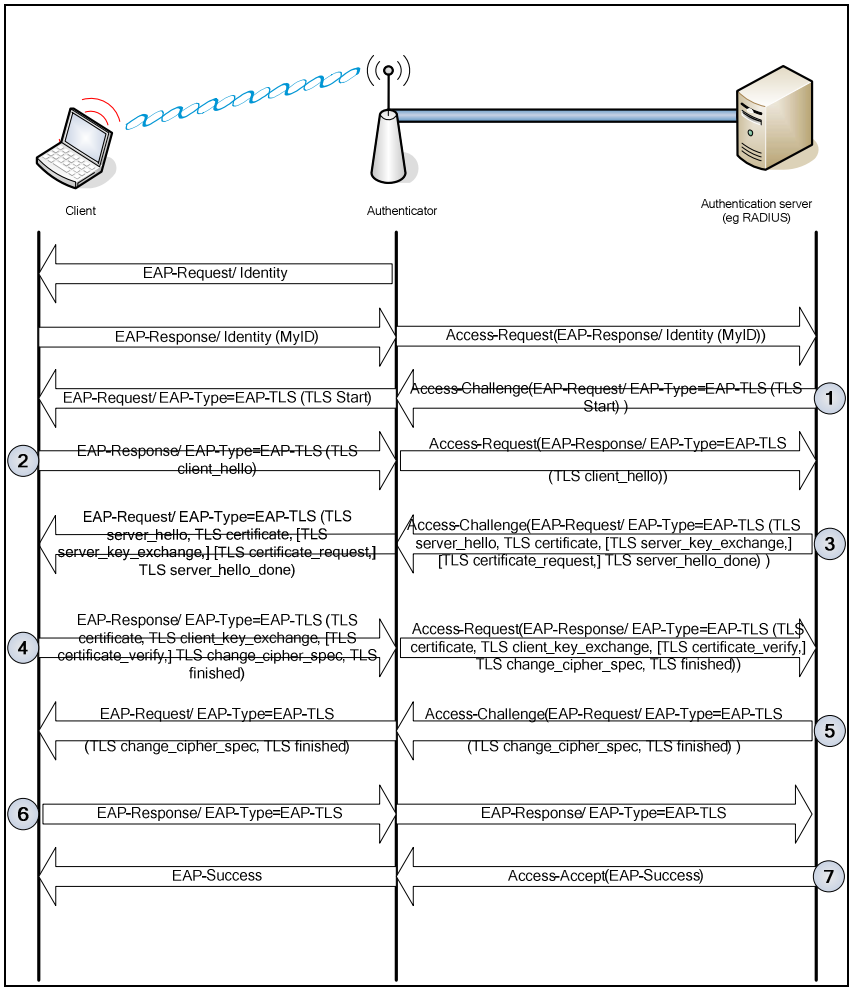

Figure 2. *Échanges EAP-TLS dans un contexte IEEE 802.11 (en cas de succès)*

C'est le serveur EAP qui engage la procédure d'authentification qui nécessite sept messages. Les messages 2 à 5 sont une adaptation des échanges du protocole Handshake de TLS (Dierks, 1999). Ces messages permettent aux client et serveur de s'authentifier mutuellement à l'aide de leurs certificats de clés publiques, de générer une clé maître commune qui servira à générer d'autres clés, de convenir d'une suite cryptographique (fonctions de hachage, algorithmes de chiffrement), de vérifier l'intégrité des échanges TLS effectués. Plus précisément, les échanges EAP-TLS contiennent la transmission des certificats des deux parties (TLS certificate), avec la possibilité d'en faire la demande explicite auprès de l'autre partie (TLS certificate_request). Ils permettent de convenir d'une clé maître commune grâce aux

échanges TLS server_key_exchange et TLS client_key_exchange, de négocier une suite cryptographique grâce aux messages TLS client_hello et TLS server_hello, de protéger les échanges TLS avec la suite cryptographique nouvellement convenue (TLS change_cipher_spec) et s'authentifier mutuellement entre client et serveur. Le client s'authentifie en émettant sa signature dans TLS certificate_verify, et le serveur grâce au message TLS finished qui prouve au client que le serveur possède la bonne clé privée.

Enfin les messages 6 et 7 terminent la session EAP-TLS avec l'émission d'une réponse EAP pour le client indiquant que le serveur s'est authentifié avec succès et l'émission par le serveur d'un message EAP-Success.

3.3. *Analyse critique*

La méthode EAP-MD5 offre l'avantage de ne pas nécessiter beaucoup de ressources pour son traitement. De plus elle n'exige pas d'infrastructure de gestion de certificats ou de clés publiques (comme le requiert EAP-TLS), mais elle n'est pas utilisée aujourd'hui car elle est reconnue comme vulnérable aux attaques par dictionnaire et aux attaques par force brute. Enfin, EAP-MD5 est une méthode d'authentification unilatérale dans la mesure où le client s'authentifie auprès du serveur, mais ne peut pas authentifier le serveur. Il n'est donc pas possible avec cette méthode de détecter de faux serveurs EAP et donc des points d'accès malveillants (contrôlés par des intrus).

La méthode EAP-TLS offre un niveau de sécurité élevé puisqu'elle permet l'authentification mutuelle entre le client. Elle est robuste face aux attaques de l'homme du milieu (Man-In-The-Middle) puisqu'elle repose sur la cryptographie asymétrique (certificats). L'inconvénient majeur de cette méthode est qu'elle repose sur une infrastructure de gestion de clé (PKI - Public Key Infrastructure) pour assurer la gestion des certificats côté client et côté serveur. Or les PKI sont extrêmement coûteuses et complexes en terme de gestion et de maintenance. De plus, elles ne s'appliquent pas très bien au contexte des mobiles. En effet, il est d'usage, pour les certificats électroniques, de vérifier la non révocation d'un certificat auprès d'une autorité ou d'un serveur miroir avant de l'utiliser pour authentifier l'entité. Dans le cas d'un mobile, cette phase de vérification ne peut pas être réalisée puisque le mobile n'a pas encore accès au réseau. Lors de sa demande de connexion, le mobile va donc prendre le risque d'accepter comme valide le certificat du serveur alors qu'il n'a aucun moyen de vérifier son état de validité.

4. Méthode EAP-EHash

Comme nous l'avons vu précédemment, les méthodes EAP-TLS et EAP-MD5 ne sont pas idéales pour fonctionner dans un environnement IEEE 802.11 du fait de leur

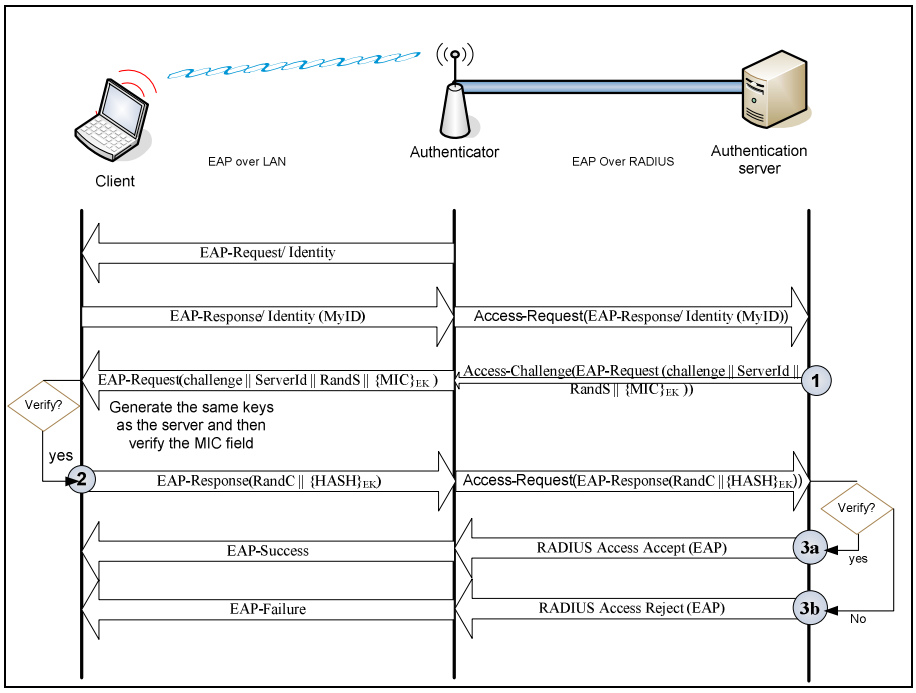

Figure 3. *Échanges EAP-EHash dans un contexte IEEE 802.11*

lourdeur de gestion, de l'authentification unilatérale, des attaques de l'homme du milieu possibles... D'autres méthodes EAP ont été standardisées comme EAP-OTP (One-Time-Password) et EAP-GTC (Generic Token Card) (Aboba, 2004), qui sont simples d'usage, mais ne proposent qu'une authentification unilatérale ou encore EAP-SIM (Subscriber Identity Modules) (Haverinen, 2006), et EAP-AKA (Authentication and Key Agreement) (Arkko, 2006) qui ont été spécifiées pour fonctionner dans les environnements GSM/UMTS et qui sont plutôt prévues pour fonctionner à l'aide de cartes à puce.

Nous proposons ici une nouvelle méthode EAP-EHash qui se veut simple, et basée sur des clés symétriques. Elle permet l'authentification mutuelle, la mise en place d'une clé maître, et tire avantage des méthodes EAP-TLS et EAP-MD5 en évitant les vulnérabilités détectées dans ces méthodes.

Pour cela, EAP-EHash suppose qu'une clé PSK (Pre-Shared Key) est pré-partagée entre le client et le serveur EAP et reprend le principe du protocole CHAP (Simpson, 1996) qui est léger dans son traitement. EAP-EHash introduit deux mécanismes : un mécanisme pour fournir l'authentification mutuelle et un

mécanisme pour éviter les attaques par dictionnaire et par force brute (cf. l'analyse de la section 5).

Comme l'illustre la figure 3, après avoir reçu l'identité du client, le serveur EAP récupère PSK et génère les éléments suivants :

– RandS : un nombre aléatoire

– Challenge : un défi (nombre aléatoire) utilisé pour authentifier le client

– AK (Authentication Key) : une clé d'authentification dérivée de la clé PSK

AK= F (PSK, RandS) où F est une fonction aléatoire à sens unique (par exemples HMAC-SHA-1 ou HMAC-MD5).

– EK (Encryption Key) : clé de chiffrement servant à chiffrer les champs MIC et HASH et dérivée de PSK

EK= F (PSK, RandS || ServerID || ClientID) où || désigne la concaténation, et ClientID/ServerID sont les identifiants des client et serveur.

– MIC (Message Integrity Check) : contrôle d'intégrité cryptographique calculé sur les données à transmettre avec la clé AK, puis chiffré avec EK.

MIC = F (AK, Challenge || ServerID || RandS)

Le serveur envoie l'ensemble de ces paramètres (message 1 de la figure 3), y compris le champ MIC chiffré avec la clé EK, au client via l'authentificateur. À réception de ce message, le client génère les mêmes clés AK et EK que le serveur (à partir de PSK) et authentifie le serveur en comparant le MIC chiffré reçu avec le MIC calculé localement par le client. Le serveur est correctement authentifié uniquement en cas de MIC identiques. Les clés AK et EK sont utilisées pour éviter de trop exposer la clé PSK en en faisant une exploitation directe dans la génération des MIC.

En cas de succès de l'authentification du serveur, le client génère un nombre aléatoire RandC et s'authentifie à son tour en calculant un HASH à partir du Challenge envoyé par le serveur comme suit : HASH = F (AK, Challenge || RandC).

Dans le message 2, le client envoie RandC et la valeur du HASH chiffrée avec EK, au serveur, dans un message EAP-Response. Le serveur qui reçoit le message authentifie le client en comparant le HASH reçu avec celui calculé en local.

Dans le cas d'une authentification du client réussie, une nouvelle clé IK est calculée : IK= F (PSK, RandS || RandC).

Cette clé IK pourra par exemple servir à initialiser le protocole IKE (Internet Key Exchange) dans le cas d'un besoin de VPN IPsec entre le client et le point d'accès, ou bien elle servira à dériver des clés de chiffrement IEEE 802.11 partagées entre le client et le point d'accès. Notons que le serveur EAP devra alors communiquer cette clé IK au point d'accès.

Le message 3 de type EAP-Success termine finalement la phase d'authentification EAP.

5. Analyse de EAP-EHash

La méthode EAP-EHash présente plusieurs avantages, à savoir :

– Traitement léger : Basée sur la cryptographie symétrique, EAP-EHash n'exige pas de traitements importants. Cette propriété est essentielle dans les réseaux sans fil puisque les nœuds mobiles présentent des ressources limitées en terme de batterie, de puissance de calcul et de capacité mémoire.

– Authentification rapide : Comme EAP-MD5, EAP-EHash est basée sur un mécanisme de défi/réponse et par conséquent exige peu d'échanges de messages. Le nombre réduit des messages échangés rend l'authentification rapide. Cette propriété est importante dans les réseaux sans fils puisque les noeuds sont mobiles et peuvent à tout moment perdre leurs connexions, ce qui est très perturbant pour les méthodes d'authentification et oblige le mobile à réinitialiser la procédure.

– Authentification mutuelle : La méthode EAP-MD5 ne permet pas au client d'authentifier le serveur, et donc de détecter un point d'accès frauduleux qui serait mis en place par une personne malveillante dans le but d'espionner les communications des mobiles. La méthode EAP-EHash permet de se prémunir de cette attaque en permettant une authentification mutuelle qui se base sur une clé pré-partagée.

– Efficacité contre les attaques par dictionnaire : Avec EAP-MD5, il est possible de réaliser une attaque par dictionnaire car un espion peut avoir accès au texte en clair et au hash correspondant et se servir de ces informations pour découvrir la clé pré-partagée. Avec EAP-EHash, cette attaque n'est plus possible car le hash (que ce soit le MIC émis par le serveur ou le HASH émis par le client) est chiffré avec la clé EK.

6. Spécifications et validation

Cette section présente les résultats de validation de la méthode EAP-EHash obtenus à l'aide de l'outil AVISPA (Automated Validation of Internet Security Protocols and Applications) (AVISPA, 2006). Les spécifications obtenues sont disponibles à (Cheikhrouhou, 2006-1), et la figure 4 en présente un extrait.

Les propriétés suivantes ont pu être testées :

– Authentification mutuelle : Le client authentifie le serveur par vérification du champ MAC car l'instruction suivante permet de vérifier que le champ MAC reçu du serveur est égal à la valeur attendue, ce qui prouve que le serveur a pu calculer la valeur des clés AK et EK et donc qu'il connaît la valeur de la clé pré-partagée PSK `MAC'={MIC(AK',Challenge'.serverId.RandS')} EK'.`

De même, l'instruction `HASH'={HMAC(AK,Challenge.RandP')}_EK` permet au serveur de vérifier que le HASH calculé par le client correspond bien au défi (Challenge) envoyé.

```
role server(
...
played_by S def=
...
1.State=1 /\ RCV(respond_id.peerId)=|>
  State':=2 /\ RandS':=new()
               /\ Challenge':=new()
               /\ AK':=PRF(PSK.RandS')
               /\ EK':=PRF(PSK.RandS'.serverId.peerId)
               /\ MAC':={MIC(AK',Challenge'.serverId.RandS')}_EK'
            /\ SND(Challenge'.serverId.RandS'.MAC')
               /\ witness(S,P,rs,RandS')
               /\ witness(S,P,ch,Challenge')
            /\ secret(AK',sec_ak,{P,S})
            /\ secret(EK',sec_ek,{P,S})

2. State=2 /\ RCV(RandP'.HASH')
            /\HASH'={HMAC(AK,Challenge.RandP')}_EK
=|>
   State':=3 /\ request(S,P,RandP')
                /\ SND(success)
end role %server
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role peer(
...
played_by P def=
...
1.State=1 /\ RCV(Challenge'.serverId.RandS'.MAC')
           /\ AK'=PRF(PSK.RandS')
          /\ EK'=PRF(PSK.RandS'.serverId.peerId)
           /\ MAC'={MIC(AK',Challenge'.serverId.RandS')}_EK'
=|>
  State':=2 /\ RandP':=new()
               /\ HASH':={HMAC(AK',Challenge'.RandP')}_EK'
               /\ SND(RandP'.HASH')
               /\ witness(P,S,rp,RandP')
               /\ request(P,S,rs,RandS')
               /\ request(P,S,ch,Challenge')
            /\ secret(AK',sec_ak,{P,S})
            /\ secret(EK',sec_ek,{P,S})
...
end role %peer
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
role environnement()
def=
  const
        p,s,i:agent,
        mic,h,prf:function,
        psk_ps,psk_pi,psk_is:symmetric_key

intruder_knowledge={p,s,mic,h,prf,psk_pi,psk_is}

  composition
        session(p,s,psk_ps,mic,h,prf)
       /\session(p,i,psk_pi,mic,h,prf)
       /\session(i,s,psk_is,mic,h,prf)
end role
```

Figure 4. Extrait des spécifications de EAP-EHash dans le langage HLPSL (cf. (Chreikhrouhou, 2006-2) pour les spécifications détaillées)

– Secret des clés : Dans le langage de spécification HLPSL (High-Level Protocol Specification Language) (Chevalier, 2004) défini par AVISPA, il est possible d'exprimer que les clés AK et EK doivent rester secrètes entre client (P) et serveur (S), grâce aux deux instructions : `secret(AK',sec_ak,{P,S})` et `secret(EK',sec_ek,{P,S})`.

– Robustesse à l'attaque de l'homme du milieu : Le but de l'homme du milieu est de partager une clé de session IK_as avec le serveur et IK_pa avec le client en leur faisant croire qu'ils communiquent en direct de façon protégée. Cette attaque est possible si les paramètres échangés servant à la construction de la clé ne sont pas authentifiés. Les deux primitives request et witness de HLPSL permettent d'authentifier l'origine des messages. L'instruction `witness(P,S,rp,RandP')`, signifie que le client P veut communiquer avec le serveur `S` en utilisant la valeur `RandP'` comme valeur du paramètre `rp`, et l'instruction `request(S,P,rp,RandP')`, que le serveur S a accepté la valeur `RandP'` comme valeur de `rp` et maintenant doit vérifier que cette valeur est réellement envoyée par P. De même le client doit authentifier le serveur sur la valeur de `RandS'`. Cette propriété a été vérifiée avec succès dans la méthode EAP-EHash et par conséquent elle est robuste contre l'attaque Man-In-the-Middle.

– Protection contre le rejeu des messages : Pendant la phase de validation, l'outil AVISPA rejoue d'anciens messages (utilisés dans une session antérieure). Le résultat de la validation montre que la méthode EAP-EHash est robuste contre le rejeu. En effet, les deux paramètres RandP et RandS permettent d'assurer la fraîcheur des messages échangés.

Méthodes EAP	Authentification mutuelle	Principe	Infrastructure PKI exigée	Rapidité de l'authentification
EAP-TLS	Oui	Basée sur TLS et des certificats électroniques	Oui	Lente
EAP-MD5	Non	Défi-réponse	Non	Rapide
EAP-EHash	Oui	Défi-réponse avec chiffrement de la réponse	Non	Rapide

Tableau 1. Synthèse comparative des méthodes EAP

7. Conclusions

Partant du constat que le concept du protocole EAP est couramment utilisé dans les réseaux IP filaires ou sans fil pour authentifier les utilisateurs, et que les méthodes EAP courantes (EAP-TLS et EAP-MD5) dans ces environnements là n'offrent pas toutes les propriétés attendues pour une mise en œuvre aisée, nous

proposons la méthode EAP-EHash. Cette dernière offre des propriétés intéressantes de simplicité, de rapidité d'exécution, d'authentification mutuelle et de robustesse face à des attaques comme l'homme du milieu ou des attaques par dictionnaire.

Le tableau 1 compare les propriétés obtenues pour EAP-EHash et pour deux autres méthodes classiques (EAP-TLS et EAP-MD5). En particulier, il met en évidence sa simplicité de déploiement sans besoin d'infrastructure PKI (contrairement à EAP-TLS), l'assurance de se connecter à un réseau légitime grâce à l'authentification mutuelle, et une consommation en énergie modeste au niveau du mobile pendant la phase d'authentification du fait de la rapidité d'exécution et du traitement simple de EAP-EHash. Pour faciliter son déploiement à grande échelle, il serait envisageable de considérer que la clé pré-partagée est générée à partir d'un mot de passe connu de l'utilisateur ; ainsi, serait éliminé le problème du stockage (confidentiel) de la clé dans le mobile et on se retrouverait avec la même facilité d'usage que pour EAP-MD5 tant sur les environnements IP de réseaux filaires que sans fil.

8. Remerciements

Nous tenons à remercier Monsieur Mohamed Jmaiel (ENIS, Tunisie) pour son aide sur la validation, ainsi que le GET qui a soutenu le projet EcoMesh.

9. Bibliographie

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

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

Aboba B., Blunk L., Vollbrecht J., Carlson J., Levkowetz H., Extensible Authentication Protocol (EAP), RFC 3748, June 2004.

Arkko J., Haverinen H., Extensible Authentication protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA), RFC 4187, January 2006.

AVISPA project, disponible à http://www.avispa-project.org, 2006.

Cheikhrouhou O., Laurent-Maknavicius M., Ben Jemaa M., Sécurité des réseaux mesh sans fil, rapport de Master Recherche, Ecole Nationale d'Ingénieurs de Sfax, 2006-1.

Cheikhrouhou O., Laurent-Maknavicius M., Ben Jemaa M., http://www-lor.int-evry.fr/~maknavic/articles/EAP-EHash-web.pdf, 2006-2.

Dierks T., Allen C., The TLS Protocol Version 1.0, RFC 2246, January 1999.

Haverinen H., Salowey J., Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM), RFC 4186, January 2006.

Chevalier Y., Compagna L., Cuellar J., Drielsma P.H., Mantovani J., M¨odersheim S., Vigneron L.., « A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols », *Automated Software Engineering*, Vol.180, Sept. 2004, p. 193-205.

Simpson W., PPP Challenge Handshake Authentication Protocol (CHAP), RFC 1994, August 1996