Groupe de travail Réseau
Request for Comments : 5216
RFC rendue obsolète : 2716
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

D. Simon, Microsoft Corporation
B. Aboba, Microsoft Corporation
R. Hurst, Microsoft Corporation
mars 2008

Protocole d'authentification EAP-TLS

Résumé
Le protocole d'authentification extensible (EAP, *Extensible Authentication Protocol*) défini dans la RFC 3748, fournit la prise en charge de plusieurs méthodes d'authentification. La sécurité de la couche transport (TLS, *Transport Layer Security*) assure l'authentification mutuelle, la négociation de suites de chiffrement protégées en intégrité, et l'échange de clés entre deux points d'extrémité. Le présent document définit EAP-TLS, qui inclut la prise en chatge de l'authentification mutuelle fondée sur le certificat et la déduction de clé.

Le présent document rend obsolète la RFC 2716. Un résumé des changements entre le présent document et la RFC 2716 est disponible à l'Appendice A.

Table des Matières

1. Introduction....1
1.1 Exigences....2
1.2 Terminologie....2
2. Vue d'ensemble du protocole....2
2.1 Généralités sur la conversation EAP-TLS....2
2.2 Vérification d'identité....8
2.3 Hiérarchie des clés....8
2.4 Négociation de suite de chiffrement et de compression....10
3. Description détaillée du protocole EAP-TLS....10
3.1 Paquet de demande EAP-TLS....10
3.2 Paquet de réponse EAP-TLS....11
4. Considérations relatives à l'IANA....12
5. Considérations sur la sécurité....12
5.1 Revendications de sécurité....12
5.2 Identités d'homologue et de serveur....12
5.3 Validation de certificat....13
5.4 Révocation de certificat....14
5.5 Attaques de modification de paquet....14
6. Références....14
6.1 Références normatives....14
6.2 Références pour information....15
Remerciements....16
Appendice A. Changements par rapport à la RFC 2716....16
Adresse des auteurs....17
Déclaration complète de droits de reproduction....17

1. Introduction

Le protocole d'authentification extensible (EAP, *Extensible Authentication Protocol*) décrit dans la [RFC3748], fournit un mécanisme standard pour la prise en charge de plusieurs méthodes d'authentification. Par l'utilisation de EAP, la prise en charge d'un certain nombre de schémas d'authentification peut être ajoutée, incluant des cartes à mémoire, Kerberos, une clé publique, des mots de passe à usage unique, et autres. EAP a été défini pour être utilisé avec diverses couches inférieures, incluant le protocole point à point (PPP, *Point-to-Point Protocol*) [RFC1661], des protocoles de tunnelage de

couche 2 comme le protocole de tunnelage en point à point (PPTP, *Point-to-Point Tunneling Protocol*) [RFC2637] ou le protocole de tunnelage de couche 2 (L2TP, *Layer 2 Tunneling Protocol*) [RFC2661], les réseaux filaires IEEE 802 [IEEE-802.1X], et les technologies sans fil comme IEEE 802.11 [IEEE-802.11] et IEEE 802.16 [IEEE-802.16e].

Alors que les méthodes EAP définies dans la [RFC3748] ne prenaient pas en charge l'authentification mutuelle, l'utilisation de EAP avec des technologies sans fil comme [IEEE-802.11] a résulté en le développement d'un nouvel ensemble d'exigences. Comme décrit dans "Exigences de méthode pour le protocole d'authentification extensible (EAP) pour les LAN sans fil" [RFC4017], il est souhaitble que les méthodes EAP utilisées pour l'authentification de LAN sans fil prennent en charge l'authentification mutuelle et la déduction de clé. D'autres couches de liaison peuvent aussi utiliser EAP pour permettre l'authentification mutuelle et la déduction de clé.

Le présent document définit la sécurité de couche transport sur EAP (EAP-TLS, *EAP-Transport Layer Security*) qui inclut la prise en charge de l'authentification mutuelle fondée sur le certificat et la déduction de clé, en utilisant les capacités de négociation protégée de suite de chiffrement, l'authentification mutuelle et la gestion de clé du protocole TLS, décrites dans "Protocole de sécurité de la couche Transport (TLS) version 1.1" [RFC4346]. Bien que le présent document rende obsolète la [RFC2716], il reste rétro compatible avec elle. Un résumé des changements entre le présent document et la RFC 2716 est disponible à l'Appendice A.

1.1 Exigences

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

1.2 Terminologie

Le présent document utilise fréquemment les termes suivants

Authentificateur : entité qui initie l'authentification EAP.

Homologue : entité qui répond à l'authentificateur. Dans [IEEE-802.1X], cette entité est appelée le soumettant.

Serveur d'authentification d'extrémité arrière : entité qui fournit un service d'authentfiication à un authentificateur. Quand il est utilisé, ce serveur exécute normalement les méthodes EAP pour l'authentificateur. Cette terminologie est aussi utilisée dans [IEEE-802.1X].

Serveur EAP : entité qui termine la méthode d'authentification EAP avec l'homologue. Dans le cas où un serveur d'authentification d'extrémité arrière n'est pas utilisé, le serveur EAP fait partie de l'authentificateur. Dans le cas où l'authentificateur opère en mode traversée, le serveur EAP est situé sur le serveur d'authentification d'extrémité arrière.

Clé de session maîtresse (MSK, *Master Session Key*) : matériel de chiffrement qui est déduit entre l'homologue EAP et le serveur et exporté par la méthode EAP.

Clé de session maîtresse étendue (EMSK, *Extended Master Session Key*) : matériel de chiffrement supplémentaire déduit entre l'homologue EAP et le serveur et qui est exporté par la méthode EAP.

2. Vue d'ensemble du protocole

2.1 Généralités sur la conversation EAP-TLS

Comme décrit dans la [RFC3748], la conversation EAP-TLS va normalement commencer par la négociation de EAP entre l'authentificateur et l'homologue. L'authentificateur va alors normalement envoyer un paquet Demande/identité EAP à l'homologue, et l'homologue va répondre par un paquet Réponse/identité EAP à l'authentificateur, contenant l'identifiant d'utilisateur *(user-Id)* de l'homologue.

À partir de ce moment, alors que nominalement la conversation EAP se produit entre l'authentificateur EAP et l'homologue, l'authentificateur PEUT agir comme un appareil de traversée, les paquets EAP reçus de l'homologue étant encapsulés pour transmission à un serveur d'authentification d'extrémité arrière. Dans la discussion qui suit, on utilisera le terme de "serveur EAP" pour noter le point d'extrémité ultime qui converse avec l'homologue.

2.1.1 Cas de base

Une fois qu'il a reçu l'identité de l'homologue, le serveur EAP DOIT répondre avec un paquet EAP-TLS/Start *(début)* qui est un paquet Demande EAP avec EAP-Type=EAP-TLS, le bit S (Start) établi, et pas de données. La conversation EAP-TLS va alors commencer, avec l'homologue qui envoie un paquet Réponse EAP avec EAP-Type=EAP-TLS. Le champ Données de ce paquet va encapsuler un ou plusieurs enregistrements TLS en format de couche d'enregistrement TLS, contenant un message TLS de prise de contact client_hello. La spécification de chiffrement courante pour les enregistrements TLS va être TLS_NULL_WITH_NULL_NULL et la compression nulle. Cette spécification de chiffrement courante reste la même jusqu'à ce que le message change_cipher_spec *(changement de spécification de chiffrement)* signale que les enregistrements qui suivent vont avoir les attributs négociés pour le reste de la prise de contact.

Le message client_hello contient le numéro de version TLS de l'homologue, un identifiant de session *(sessionId)* un nombre aléatoire, et un ensemble de suites de chiffrement supportées par l'homologue. La version offerte par l'homologue DOIT correspondre à TLS v1.0 ou ultérieure.

Le serveur EAP va alors répondre avec un paquet Demande EAP avec EAP-Type=EAP-TLS. Le champ Données de ce paquet va encapsuler un ou plusieurs enregistrements TLS. Ceux-ci vont contenir un message TLS de prise de contact server_hello, éventuellement suivi par un certificat TLS, server_key_exchange *(échange de clé de serveur)* certificate_request *(demande de certificat)* server_hello_done *(prise de contact de serveur terminée)* et/ou des messages de fin de prise de contact, et/ou un message TLS change_cipher_spec. Le message de prise de contact server_hello contient un numéro de version TLS, un autre nombre aléatoire, un sessionId, et une ciphersuite. La version offerte par le serveur DOIT correspondre à TLS v1.0 ou ultérieure.

Si le sessionId de l'homologue est nul ou non reconnu par le serveur, le serveur DOIT choisir l'identifiant de session pour établir une nouvelle session. Autrement, l'identifiant de session va correspondre à celui offert par l'homologue, indiquant une reprise de la session précédemment établie avec cet identifiant de session. Le serveur va aussi choisir une suite de chiffrement dans celles offertes par l'homologue. Si la session correspond à celle de l'homologue, alors la suite de chiffrement DOIT correspondre à celle négociée durant l'exécution du protocole de prise de contact qui a établi la session.

Si le serveur EAP ne reprend pas une session établie précédemment, il DOIT alors inclure un message de prise de contact TLS server_certificate, et un message de prise de contact server_hello_done DOIT être le dernier message de prise de contact encapsulé dans ce paquet Demande EAP.

Le message Certificat contient une chaîne de certificats de clé publique pour une clé publique d'échange de clé (comme une clé publique d'échange de clé RSA ou Diffie-Hellman) ou une clé publique de signature (comme une clé publique de signature RSA ou de la norme de signature numérique (DSS, *Digital Signature Standard*)). Dans ce dernier cas, un message de prise de contact TLS server_key_exchange DOIT aussi être inclus pour permettre l'échange de clé.

Le message certificate_request *(demande de certificat)* est inclus lorsque le serveur désire que l'homologue s'authentifie via une clé publique. Alors que le serveur EAP DEVRAIT exiger l'authentification de l'homologue, celle-ci n'est pas obligatoire, car il y a des circonstances où l'authentification de l'homologue ne va pas être nécessaire (par exemple, pour les services d'urgence, comme décrit dans la [RFC7406]), ou lorsque l'homologue va s'authentifier par d'autres moyens.

Si l'homologue prend en charge EAP-TLS et est configuré à l'utiliser, il DOIT répondre à la demande EAP avec un paquet Réponse EAP EAP-Type=EAP-TLS. Si le message server_hello précédent envoyé par le serveur EAP dans le paquet Demande EAP précédent n'indiquait pas la reprise d'une session précédente, le champ Données de ce paquet DOIT encapsuler un ou plusieurs enregistrements TLS contenant les messages TLS client_key_exchange, change_cipher_spec, et terminé. Si le serveur EAP a envoyé un message certificate_request dans le paquet Demande EAP précédent, alors, sauf si l'homologue est configuré pour la confidentialité (voir au paragraphe 2.1.4) l'homologue DOIT de plus envoyer des messages "certificate" et "certificate_verify". Le premier contient un certificat pour la clé publique de signaure de l'homologue, tandis que le dernier contient la réponse d'authentification signée de l'homologue au serveur EAP. Après avoir reçu ce paquet, le serveur EAP va vérifier le certificat et la signature numérique de l'homologue, si c'est demandé.

Si le précédant message server_hello envoyé par le serveur EAP dans le paquet Demande EAP précédent indiquait la reprise d'une session précédente, alors l'homologue DOIT envoyer seulement les messages de prise de contact change_cipher_spec et reminé. Le message terminé contient la réponse d'authentification de l'homologue au serveur EAP.

Dans le cas où l'authentification mutuelle EAP-TLS réussit, la conversation va apparaître comme suit :

Homologue qui s'authentifie **Authentificateur**

<- EAP-Request/Identity
EAP-Response/Identity (MyID) ->
<- EAP-Request/ EAP-Type=EAP-TLS (début de TLS)
EAP-Response/ EAP-Type=EAP-TLS
(client_hello TLS)->
<- EAP-Request/ EAP-Type=EAP-TLS (server_hello TLS , certificat TLS,
[TLS server_key_exchange,] TLS certificate_request, TLS server_hello_done)
EAP-Response/
EAP-Type=EAP-TLS
(TLS certificate, TLS client_key_exchange,
TLS certificate_verify, TLS change_cipher_spec,
TLS terminé) ->
<- EAP-Request/ EAP-Type=EAP-TLS (TLS change_cipher_spec, TLS terminé)
EAP-Response/ EAP-Type=EAP-TLS ->
<- EAP-Success

2.1.2 Reprise de session

L'objet du sessionId dans le protocole TLS est de permettre une amélioration de l'efficacité dans le cas où un homologue tente de façon répétée de s'authentifier auprès d'un serveur EAP sur une courte période. Bien que ce modèle ait été développé pour être utilisé avec l'authentification HTTP, il aussi peut être utilisé pour fournir la fonction de "reconnexion rapide" définie au paragraphe 7.2.1 de la [RFC3748].

Il appartient à l'homologue de décider si il tente de continuer une session précédente, raccourcissant donc la conversation TLS. Normalement, la décision de l'homologue va être prise sur la base du temps écoulé depuis la tentative précédente d'authentification auprès de ce serveur EAP. Sur la base du sessionId choisi par l'homologue, et du temps écoulé depuis l'authentification précédente, le serveur EAP va décider si il permet la continuation ou si il choisit une nouvelle session.

Dans le cas où le serveur EAP et l'authentificateur résident sur le même appareil, l'homologue va seulement être capable de continuer les sessions quand il se connecte au même authentificateur. Si les authentificateurs sont établis à tour de rôle ou par un processus de round-robin, il peut alors n'être pas possible à l'homologue de connaître à l'avance l'authentificateur auquel il va être connecté, et donc quel sessionId tenter de réutiliser. Par suite, il est probable que la tentative de continuation va échouer. Dans le cas où l'authentification EAP est à distance, la continuation a plus de chances de réussir, car plusieurs authentificateurs vont utiliser le même serveur d'authentification d'extrémité arrière.

Si le serveur EAP reprend une session précédemment établie, il DOIT alors inclure seulement un message TLS change_cipher_spec et un message Fin de prise de contact TLS après le message server_hello. Ce message de terminaison contient la réponse d'authentification du serveur EAP à l'homologue.

Dans le cas où une session établie précédemment est reprise, et où les deux côtés réussissent à s'authentifier, la conversation va apparaître comme suit :

Homologue qui s'authentifie **Authentificateur**

<- EAP-Request/Identity
EAP-Response/Identity (MyID) ->
<- EAP-Request/ EAP-Request/ EAP-Type=EAP-TLS (Début TLS)
EAP-Response/ EAP-Type=EAP-TLS
(client_hello TLS)->
<- EAP-Request/ EAP-Type=EAP-TLS (server_hello TLS,
change_cipher_spec TLS, TLS terminé)
EAP-Response/ EAP-Type=EAP-TLS
(change_cipher_spec TLS, TLS terminé) ->
<- EAP-Success

2.1.3 Terminaison

Si l'authentification de l'homologue ne réussit pas, le serveur EAP DEVRAIT envoyer un paquet Demande EAP avec EAP-Type=EAP-TLS, encapsulant un enregistrement TLS contenant le message d'alerte TLS approprié. Le serveur EAP DEVRAIT envoyer un message d'alerte TLS terminant immédiatement la conversation afin de permettre à l'homologue

d'informer l'utilisateur ou d'enregistrer la cause de la défaillance et permettre éventuellement un redémarrage de la conversation.

Pour assurer que l'homologue reçoit le message d'alerte TLS, le serveur EAP DOIT attendre que l'homologue réponde avec un paquet Réponse EAP. Le paquet Réponse EAP envoyé par l'homologue PEUT encapsuler un message de prise de contact TLS client_hello, et dans ce cas, le serveur EAP PEUT permettre que la conversation EAP-TLS soit redémarrée, ou il PEUT contenir un paquet Réponse EAP avec EAP-Type=EAP-TLS et pas de données, et dans ce cas, le serveur EAP DOIT envoyer un paquet EAP-Failure *(échec d'EAP)* et terminer la conversation. Il appartient au serveur EAP de permettre ou non le redémarrage, et si il le permet, combien de fois la conversation peut être redémarrée. Un serveur EAP qui met en œuvre la capacité de redémarrage DEVRAIT imposer une limite par homologue du nombre de redémarrages, afin de protéger contre des attaques de déni de service.

Si l'homologue réussit à s'authentifier, le serveur EAP DOIT répondre avec un paquet Demande EAP avec EAP-Type=EAP-TLS, qui inclut, dans le cas d'une nouvelle session TLS, un ou plusieurs enregistrements TLS contenant des messages de prise de contact TLS change_cipher_spec et Terminé. Ce dernier contient la réponse d'authentification du serveur EAP à l'homologue. L'homologue va alors vérifier le message Terminé afin d'authentifier le serveur EAP.

Si l'authentification du serveur EAP ne réussit pas, l'homologue DEVRAIT supprimer la session de son antémémoire, empêchant la réutilisation du sessionId. L'homologue PEUT envoyer un paquet Réponse EAP de EAP-Type=EAP-TLS contenant un message d'alerte TLS qui identifie la raison de l'échec d'authentification. L'homologue PEUT envoyer un message d'alerte TLS plutôt que de terminer immédiatement la conversation afin de permettre au serveur EAP d'enregistrer la cause de l'erreur pour qu'elle soit examinée par l'administrateur du système.

Pour s'assurer que le serveur EAP reçoit le message d'alerte TLS, l'homologue DOIT attendre que le serveur EAP réponde avant de terminer la conversation. Le serveur EAP DOIT répondre avec un paquet Échec EAP car l'échec de l'authentification du serveur est une condition terminale.

Si l'authentification du serveur EAP réussit, l'homologue DOIT envoyer un paquet Réponse EAP de EAP-Type=EAP-TLS, et pas de données. Le serveur EAP DOIT alors répondre avec un message Succès EAP.

Dans le cas où le serveur réussit à s'authentifier auprès de l'homologue, mais où l'homologue échoue à s'authentifier auprès du serveur, la conversation va apparaître comme suit :

Homologue qui s'authentifie **Authentificateur**

<- EAP-Request/Identity
EAP-Response/Identity (MyID) ->
<- EAP-Request/ EAP-Type=EAP-TLS (Début TLS)
EAP-Response/ EAP-Type=EAP-TLS
(TLS client_hello)->
<- EAP-Request/ EAP-Type=EAP-TLS (server_hello TLS, certificat TLS,
[server_key_exchange TLS,] certificate_request TLS, server_hello_done TLS)
EAP-Response/ EAP-Type=EAP-TLS
(certificat TLS, client_key_exchange TLS,
certificate_verify TLS, change_cipher_spec TLS,
TLS Terminé) ->
<- EAP-Request/ EAP-Type=EAP-TLS (change_cipher_spec TLS, TLS Terminé)
EAP-Response/ EAP-Type=EAP-TLS ->
<- EAP-Request EAP-Type=EAP-TLS (message d'alerte TLS)
EAP-Response/ EAP-Type=EAP-TLS ->
<- EAP-Failure (Utilisateur déconnecté)

Dans le cas où l'authentification du serveur ne réussit pas, la conversation va apparaître comme suit :

Homologue qui s'authentifie **Authentificateur**

<- EAP-Request/Identity
EAP-Response/Identity (MyID) ->
<- EAP-Request/ EAP-Type=EAP-TLS (Début TLS)
EAP-Response/ EAP-Type=EAP-TLS
(client_hello TLS)->
<- EAP-Request/ EAP-Type=EAP-TLS (server_hello TLS, certificat TLS,
[server_key_exchange TLS,] certificate_request TLS, server_hello_done TLS)
EAP-Response/ EAP-Type=EAP-TLS

(message d'alerte TLS) ->
<- EAP-Failure (Utilisateur déconnecté)

2.1.4 Confidentialité

Les mises en œuvre d'homologue et de serveur EAP-TLS PEUVENT prendre en charge la confidentialité. La divulgation du nom d'utilisateur est évitée en utilisant un identifiant d'accès réseau (NAI, *Network Access Identifier*) [RFC4282] confidentiel dans la EAP-Response/Identity, et en transmettant le certificat de l'homologue au sein d'une session TLS qui assure la confidentialité.

Afin d'éviter de divulguer le nom d'utilisateur de l'homologue, un homologue EAP-TLS configuré à la confidentialité DOIT négocier une suite de chiffrement TLS qui prend en charge la confidentialité et DOIT fournir au client une liste de certificats ne conténant pas d'entrée en réponse à la demande de certificat initiale du serveur EAP-TLS.

Un serveur EAP-TLS qui prend en charge la confidentialité NE DOIT PAS traiter une liste de certificats ne contenant pas d'entrée comme une condition terminale ; il DOIT plutôt établir la session TLS et ensuite envoyer une hello_request. La prise de contact se poursuit alors normalement ; l'homologue envoie un client_hello et le serveur répond avec un server_hello, un certificat, un échange de clé de serveur, une demande de certificat, un server_hello_done, etc.

Pour le calcul du matériel de chiffrement exporté (voir le paragraphe 2.3) le secret maître déduit dans la seconde prise de contact est utilisé.

Un homologue EAP-TLS qui prend en charge la confidentialité DOIT fournir une liste de certificats contenant au moins une entrée en réponse à la demande de certificat suivante envoyée par le serveur. Si le serveur EAP-TLS qui prend en charge la confidentialité ne reçoit pas de certificat de client en réponse à la demande de certificat suivante, il DOIT alors interrompre la session.

La prise en charge de la confidentialité EAP-TLS est conçue pour permettre aux homologues EAP-TLS qui ne prennent pas en charge la confidentialité d'interopérer avec les serveurs EAP-TLS qui la prennent en charge. Les serveurs EAP-TLS qui prennent en charge la confidentialité DOIVENT demander un certificat de client, et DOIVENT être capables d'accepter un certificat de client offert par l'homologue EAP-TLS, afin de préserver l'interopérabilité avec les homologues EAP-TLS qui ne prennent pas en charge la confidentialité.

Cependant, un homologue EAP-TLS configuré pour la confidentialité ne va normalement pas être capable de réussir à s'authentifier auprès d'un serveur EAP-TLS qui ne prend pas en charge la confidentialité, car un tel serveur va normalement traiter le refus de fournir un certificat de client comme une erreur terminale. Par suite, sauf si l'échec d'authentification est considéré être préférable à la divulgation du nom d'utilisateur, les homologues EAP-TLS DEVRAIENT seulement être configurés pour la confidentialité sur les réseaux connus pour la prendre en charge.

Ceci est très facilement réalisé avec les couches inférieures EAP qui prennent en charge les annonces de réseau, de sorte que la configuration du réseau et de la confidentialité appropriée peuvent être déterminées. Afin de déterminer la configuration de confidentialité sur les couches de liaison (comme dans les réseaux filaires IEEE 802) qui ne prennent pas en charge les annonces de réseau, il peut être souhaitable d'utiliser les informations fournies dans le certificat de serveur (comme les champs "subject" et "subjectAltName") ou dans les conseils de choix d'identité [RFC4284] pour déterminer la configuration appropriée.

Dans le cas où l'homologue et le serveur prennent en charge la confidentialité et l'authentification mutuelle, la conversation va apparaître comme suit :

Homologue qui s'authentifie **Authentificateur**

<- EAP-Request/Identity
EAP-Response/ Identity (NAI anonyme) ->
<- EAP-Request/ EAP-Type=EAP-TLS (début de TLS)
EAP-Response/ EAP-Type=EAP-TLS
(client_hello TLS)->
<- EAP-Request/ EAP-Type=EAP-TLS (server_hello TLS, certificat TLS,
[server_key_exchange TLS,] certificate_request TLS, server_hello_done TLS)
EAP-Response/ EAP-Type=EAP-TLS
(certificat TLS (pas de cert), client_key_exchange TLS,
change_cipher_spec TLS, TLS Terminé) ->
<- EAP-Request/ EAP-Type=EAP-TLS (change_cipher_spec TLS, Terminé, hello_request)

```
EAP-Response/ EAP-Type=EAP-TLS
(client_hello TLS)->
                                <- EAP-Request/ EAP-Type=EAP-TLS (server_hello TLS, certificat TLS,
                                server_key_exchange TLS, certificate_request TLS, server_hello_done TLS)
EAP-Response/ EAP-Type=EAP-TLS
(certificat TLS, client_key_exchange TLS,
certificate_verify TLS, change_cipher_spec TLS,
TLS Terminé) ->
                                <- EAP-Request/ EAP-Type=EAP-TLS (change_cipher_spec TLS, TLS Terminé)
EAP-Response/ EAP-Type=EAP-TLS ->
                                <- EAP-Success
```

2.1.5 Fragmentation

Un seul enregistrement TLS peut faire jusqu'à 16384 octets de long, mais un message TLS peut s'étendre sur plusieurs enregistrements TLS, et un message de certificat TLS peut en principe faire jusqu'à 16 M octets. Le groupe de messages EAP-TLS envoyés dans un seul tour peut donc être supérieur à la taille de la MTU ou à la taille maximum de paquet du service d'authentification à distance de l'utilisateur appelant (RADIUS, *Remote Authentication Dail-In User Service*) de 4096 octets. Par suite, une mise en œuvre de EAP-TLS DOIT fournir sa propre prise en charge de la fragmentation et du réassemblage. Cependant, afin d'assurer l'interopérabilité avec les mises en œuvre existantes, les messages de prise de contact TLS NE DEVRAIENT PAS être fragmentés en plusieurs enregistrements TLS si ils tiennent dans un seul enregistrement TLS.

Afin de protéger contre les attaques de verrouillage de réassemblage et de déni de service, il peut être désirable pour une mise en œuvre d'établir une taille maximum pour un tel groupe de messages TLS. Comme un seul certificat fait rarement plus de quelques milliers d'octets, et qu'aucun autre champ n'a probablement de chances d'approcher de cette longueur, un choix raisonnable de longueur maximum acceptable de message pourrait être de 64 k octets.

Comme EAP est un protocole d'ACK-NAK simple, la prise en charge de la fragmentation peut être ajoutée de façon simple. Dans EAP, les fragments perdus ou endommagés dans le transit vont être retransmis, et comme les informations de séquençage sont fournies par le champ Identifiant dans EAP, il n'y a pas besoin d'un champ de décalage de fragment comme dans IPv4.

la prise en charge de la fragmentation EAP-TLS est fournie par l'ajout d'un octet de fanions dans les paquets Demande EAP et Réponse EAP, ainsi que d'un champ Longueur de message TLS de quatre octets. Les fanions incluent les bits Longueur incluse (L), Plus de fragments à venir (M), et Début EAP-TLS (S, *Start*). Le fanion L est établi pour indiquer la présence du champ Longueur de message TLS de quatre octets, et DOIT être établi pour le premier fragment d'un message ou ensemble de messages TLS fragmenté. Le fanion M est établi sur tous les fragments sauf le dernier. Le fanion S est établi seulement dans le message de début de EAP-TLS envoyé du serveur EAP à l'homologue. Le champ Longueur de message TLS est de quatre octets, et donne la longueur totale du message ou ensemble de messages TLS fragmenté ; cela simplifie les allocations de mémoire tampon.

Quand un homologue EAP-TLS reçoit un paquet Demande EAP avec le bit M établi, il DOIT répondre avec une Réponse EAP avec EAP-Type=EAP-TLS et pas de données. Cela sert d'accusé de réception de fragment. Le serveur EAP DOIT attendre d'avoir reçu la Réponse EAP avant d'envoyer un autre fragment. Pour empêcher des erreurs de traitement des fragments, le serveur EAP DOIT incrémenter le champ Identifiant pour chaque fragment contenu dans une demande EAP, et l'homologue DOIT inclure cette valeur d'identifiant dans l'accusé de réception de fragment contenu dans la Réponse EAP. Les fragments retransmis vont contenir la même valeur d'identifiant.

De même, quand le serveur EAP reçoit une Réponse EAP avec le bit M établi, il DOIT répondre avec une Demande EAP avec EAP-Type=EAP-TLS et pas de données. Cela sert d'accusé de réception de fragment. L'homologue EAP DOIT attendre de recevoir la Demande EAP avant d'envoyer un autre fragment. Afin d'empêcher les erreurs de traitement des fragments, le serveur EAP DOIT incrémenter la valeur de l'identifiant pour chaque accusé de réception de fragment contenu dans une Demande EAP, et l'homologue DOIT inclure cette valeur d'identifiant dans le fragment suivant contenu dans une Réponse EAP.

Dans le cas où l'authentification mutuelle EAP-TLS réussit, et où la fragmentation est nécessaire, la conversation va apparaître comme suit :

```
Homologue qui s'authentifie          Authentificateur
                                  <- EAP-Request/Identity
EAP-Response/ Identity (MyID) ->
                                  <- EAP-Request/ EAP-Type=EAP-TLS (Début TLS, bit S établi)
EAP-Response/ EAP-Type=EAP-TLS
(client_hello TLS)->
                                  <- EAP-Request/ EAP-Type=EAP-TLS (server_hello TLS, certificat TLS,
                                  [server_key_exchange TLS,] certificate_request TLS, server_hello_done TLS)
                                  (Fragment 1 : bits L, M établis)
EAP-Response/ EAP-Type=EAP-TLS ->
                                  <- EAP-Request/ EAP-Type=EAP-TLS (Fragment 2 : bit M établi)
EAP-Response/ EAP-Type=EAP-TLS ->
                                  <- EAP-Request/ EAP-Type=EAP-TLS (Fragment 3)
EAP-Response/ EAP-Type=EAP-TLS
(certificat TLS, client_key_exchange TLS,
certificate_verify TLS, change_cipher_spec TLS,
TLS Terminé)(Fragment 1 : bits L, M établis)->
                                  <- EAP-Request/ EAP-Type=EAP-TLS
EAP-Response/ EAP-Type=EAP-TLS
(Fragment 2)->
                                  <- EAP-Request/ EAP-Type=EAP-TLS (change_cipher_spec TLS, TLS Terminé)
EAP-Response/ EAP-Type=EAP-TLS ->
                                  <- EAP-Success
```

2.2 Vérification d'identité

Comme noté au paragraphe 5.1 de la [RFC3748] :

Il est RECOMMANDÉ que la réponse d'identité soit utilisée principalement à des fins d'acheminement et de choix de la méthode EAP à utiliser. Les méthodes EAP DEVRAIENT inclure un mécanisme spécifique de la méthode pour obtenir l'identité, afin qu'elles n'aient pas à s'appuyer sur la réponse d'identité.

Au titre de la négociation TLS, le serveur présente un certificat à l'homologue, et si l'authentification mutuelle est demandée, l'homologue présente un certificat au serveur. EAP-TLS fournit donc un mécanisme pour déterminer à la fois l'identité de l'homologue (Peer-Id dans la [RFC5247]) et l'identité du serveur (Server-Id dans la [RFC5247]). Pour les détails, voir le paragraphe 5.2.

Comme l'identité présentée dans la EAP-Response/Identity n'a pas besoin d'être en rapport avec l'identité présentée dans le certificat de l'homologue, les mises en œuvre de EAP-TLS NE DEVRAIENT PAS exiger qu'elles soient identiques. Cependant, si elles ne sont pas identiques, l'identité présentée dans la EAP-Response/Identity est une information non authentifiée, et NE DEVRAIT PAS être utilisée pour le contrôle d'accès ou la comptabilité.

2.3 Hiérarchie des clés

La Figure 1 illustre la hiérarchie de clés de TLS, décrite au paragraphe 6.3 de la [RFC4346]. La déduction se fait comme suit :

```
master_secret = TLS-PRF-48(pre_master_secret, "secret maître", client.random || server.random)
key_block = TLS-PRF-X(master_secret, "expansion de clé", server.random || client.random)
```

Où :

TLS-PRF-X = fonction TLS pseudo aléatoire définie dans la [RFC4346], calculée à X octets.

Dans EAP-TLS, les MSK, EMSK, et valeur d'initialisation (IV) sont déduites du secret maître TLS via une fonction unidirectionnelle. Cela assure que le secret maître TLS ne peut être déduit de MSK, EMSK, ou IV que si la fonction unidirectionnelle (TLS PRF) est cassée. Comme les MSK et EMSK sont déduites du secret maître TLS, si le secret maître TLS est compromis, alors les MSK et EMSK sont aussi compromises.

La MSK est divisée en deux moitiés, correspondant à la "clé de chiffrement d'homologue à authentfiicateur" (Enc-RECV-Key, de 32 octets) et à la "clé de chiffrement d'authentificateur à homologue" (Enc-SEND-Key, de 32 octets).

La IV est une quantité de 64 octets qui est une valeur connue ; les octets 0 à 31 sont appelés la "IV d'homologue à authentificateur" ou RECV-IV, et les octets 32 à 63 sont appelés la "IV d'authentificateur à homologue", ou SEND-IV.

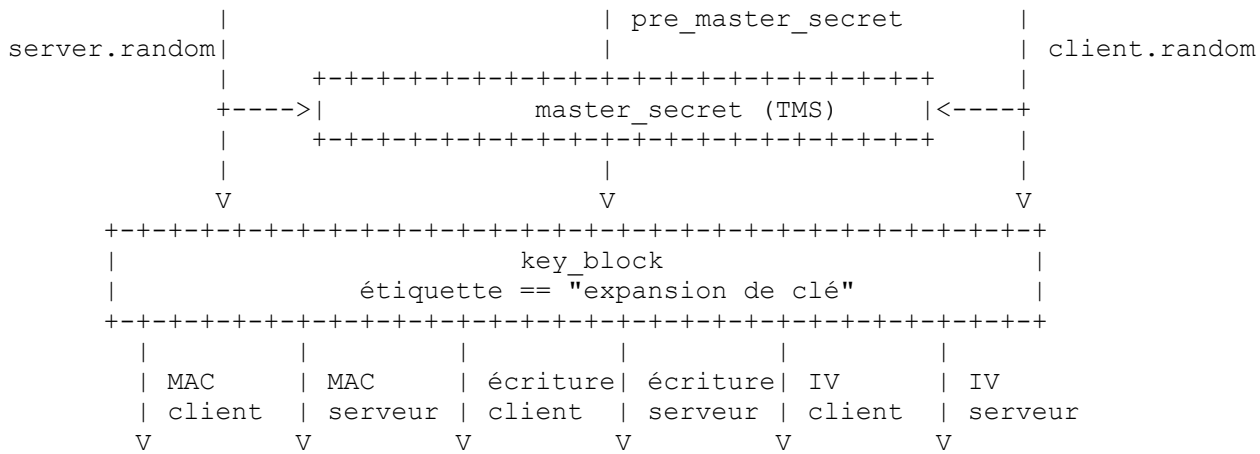


Figure 1 : Hiérarchie de clés TLS [RFC4346]

EAP-TLS déduit le matériel de chiffrement et les paramètres exportés comme suit :
Key_Material = TLS-PRF-128(master_secret, "chiffrement EAP client", client.random || server.random)
MSK = Key_Material(0,63)
EMSK = Key_Material(64,127)
IV = TLS-PRF-64("", "chiffrement EAP client", client.random || server.random)
Enc-RECV-Key = MSK(0,31) = clé de chiffrement d'homologue à authentfiicateur (MS-MPPE-Recv-Key dans la [RFC2548]). Aussi appelé la PMK dans [IEEE-802.11].
Enc-SEND-Key = MSK(32,63) = clé de chiffrement d'authentificateur à homologue (MS-MPPE-Send-Key dans la [RFC2548])
RECV-IV = IV(0,31) = valeur d'initialisation d'homologue à authentfiicateur
SEND-IV = IV(32,63) = valeur d'initialisation d'authentificateur à homologue
Session-Id = 0x0D || client.random || server.random
Où :
Key_Material(W,Z) = octets W à Z inclus du matériel de clé.
IV(W,Z) = octets W à Z inclus de l'IV.
MSK(W,Z) = octets W à Z inclus de la MSK.
EMSK(W,Z) = octets W à Z inclus de la EMSK.
TLS-PRF-X = fonction PRF TLS calculée à X octets.
client.random = nom occasionnel généré par le client TLS.
server.random = nom occasionnel généré par le serveur TLS.

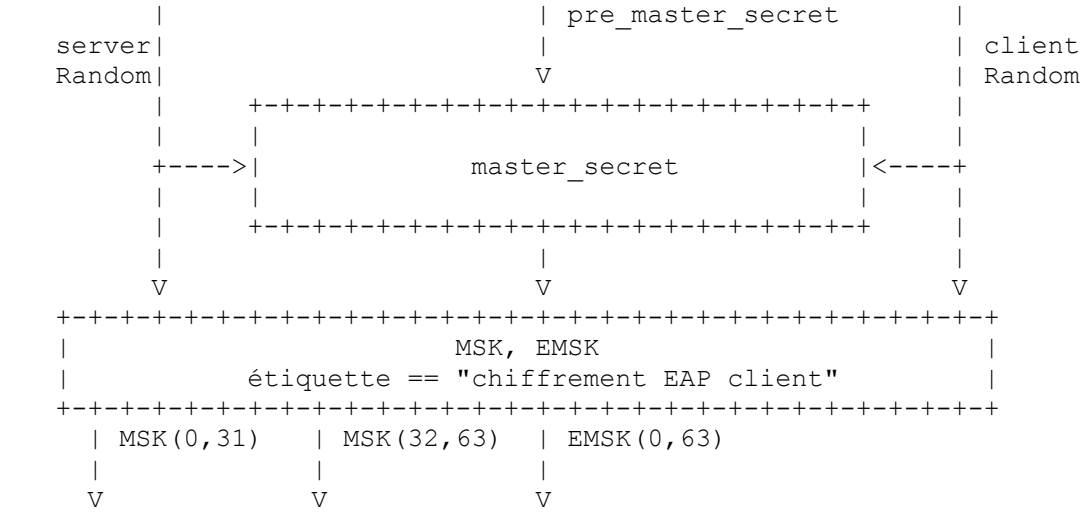


Figure 2 : Hiérarchie de clés EAP-TLS

L'utilisation de ces clés est spécifique de la couche inférieure, comme décrit au paragraphe 2.1 de la [RFC5247].

2.4 Négociation de suite de chiffrement et de compression

Les mises en œuvre EAP-TLS DOIVENT prendre en charge TLS v1.0.

Les mises en œuvre EAP-TLS n'ont pas nécessairement besoin de prendre en charge toutes les suites de chiffrement de TLS mentionnées dans la [RFC4346]. Toutes les suites de chiffrement TLS ne sont pas prises en charge par les outils TLS disponibles, et des licences peuvent être requises dans certains cas.

Pour assurer l'interopérabilité, les homologues et serveurs EAP-TLS DOIVENT prendre en charge la suite de chiffrement TLS [RFC4346] de mise en œuvre obligatoire TLS_RSA_WITH_3DES_EDE_CBC_SHA.

Les homologues et serveurs EAP-TLS DEVRAIENT aussi prendre en charge et être capables de négocier les suites de chiffrement TLS suivantes :

TLS_RSA_WITH_RC4_128_SHA [RFC4346]
TLS_RSA_WITH_AES_128_CBC_SHA [RFC3268]

De plus, les serveurs EAP-TLS DEVRAIENT prendre en charge et être capables de négocier la suite de chiffrement TLS TLS_RSA_WITH_RC4_128_MD5 [RFC4346].

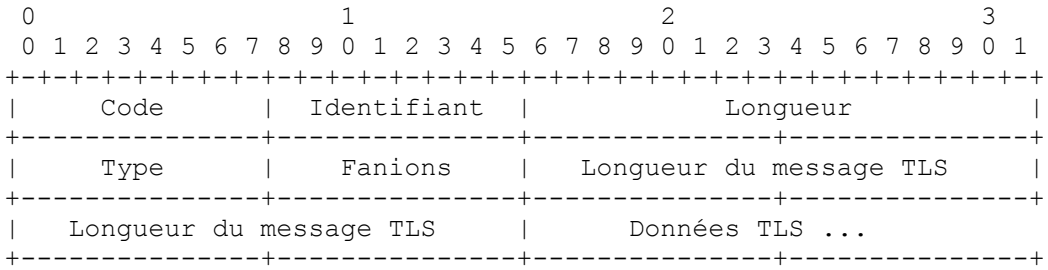
Comme TLS prend en charge la négociation de suite de chiffrement, les homologues qui réalisent la négociation TLS vont aussi avoir choisi une suite de chiffrement, qui inclut des méthodes de chiffrement et de hachage. Comme la suite de chiffrement négociée dans EAP-TLS s'applique seulement à la conversation EAP, la négociation de suite de chiffrement TLS NE DOIT PAS être utilisée pour négocier les suites de chiffrement utilisées pour sécuriser les données.

TLS prend aussi en charge la compression ainsi que la négociation de suite de chiffrement. Cependant, durant la conversation EAP-TLS, l'homologue EAP et le serveur NE DOIVENT PAS demander ou négocier la compression.

3. Description détaillée du protocole EAP-TLS

3.1 Paquet de demande EAP-TLS

Le format du paquet de demande EAP-TLS est montré ci-dessous. Les champs sont transmis de gauche à droite.



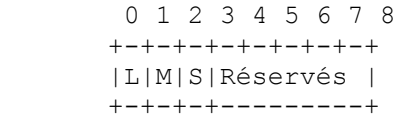
Code : 1

Identifiant : le champ Identifiant fait un octet et aide à confronter les réponses aux demandes. Le champ Identifiant DOIT être changé à chaque paquet de demande.

Longueur : le champ Longueur fait deux octets et indique la longueur du paquet EAP incluant les champs Code, Identifiant, Longueur, Type, et Données. Les octets en dehors de la gamme du champ Longueur devraient être traités comme du bourrage de couche de liaison des données et DOIVENT être ignorés à réception.

Type : 13 -- EAP-TLS

Fanions :



L = longueur incluse
M = plus de fragments à venir
S = début de EAP-TLS

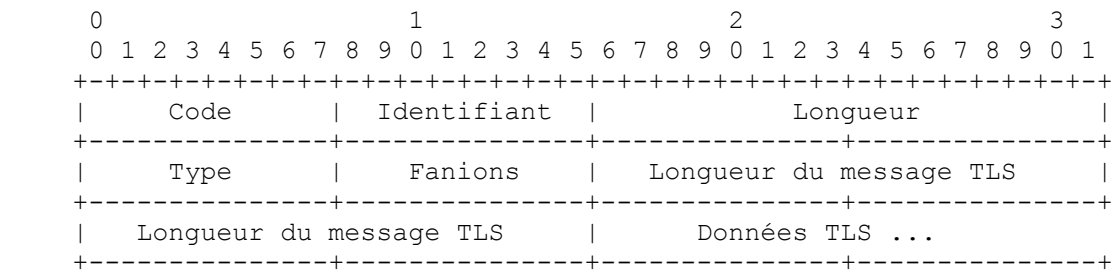
Le bit L (longueur incluse) est établi pour indiquer la présence du champ Longueur de message TLS de quatre octets, et DOIT être établi pour le premier fragment d'un message ou ensemble de messages TLS fragmenté. Le bit M (plus de fragments à venir) est établi sur tous les fragments sauf le dernier. Le bit S (début de EAP-TLS) est établi dans le message Début d'EAP-TLS. Cela différencie le message Début d'EAP-TLS d'un accusé de réception de fragment. Les mises en œuvre de cette spécification DOIVENT régler les bits réservés à zéro, et DOIVENT les ignorer à réception.

Longueur de message TLS : le champ Longueur de message TLS fait quatre octets, et n'est présent que si le bit L est établi. Ce champ donne la longueur totale du message ou ensemble de messages TLS qui va être fragmenté.

Données TLS : elles consistent en le paquet encapsulé TLS dans le format d'enregistrement TLS.

3.2 Paquet de réponse EAP-TLS

Le format du paquet Réponse EAP-TLS est montré ci-dessous. Les champs sont transmis de gauche à droite.



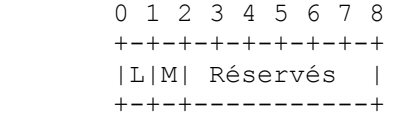
Code : 2

Identifiant : le champ Identifiant fait un octet et DOIT correspondre au champ Identifiant de la demande correspondante.

Longueur : le champ Longueur fait deux octets et indique la longueur du paquet EAP incluant les champs Code, Identifiant, Longueur, Type, et Données. Les octets hors de la gamme du champ Longueur devraient être traités comme du bourrage de couche de liaison des données et DOIVENT être ignorés à réception.

Type : 13 -- EAP-TLS

Fanions :



L = Longueur incluse
M = Plus de fragments à venir

Le bit L (longueur incluse) est établi pour indiquer la présence d'un champ Longueur de message TLS de quatre octets, et DOIT être établi pour le premier fragment d'un message ou ensemble de messages TLS fragmenté. Le bit M (plus de fragments à venir) est établi sur tous les fragments sauf le dernier. Les mises en œuvre de la présente spécification DOIVENT régler les bits réservés à zéro, et DOIVENT les ignorer à réception.

Longueur de message TLS : le champ Longueur de message TLS fait quatre octets, et n'est présent que si le bit L est établi. Ce champ donne la longueur totale du message ou ensemble de messages TLS qui est fragmenté.

Données TLS : elles consistent en le paquet TLS encapsulé en format d'enregistrement TLS.

4. Considérations relatives à l'IANA

L'IANA a alloué le type EAP 13 pour EAP-TLS. L'allocation a été mise à jour pour référencer le présent document.

5. Considérations sur la sécurité

5.1 Revendications de sécurité

Les revendications de sécurité EAP sont définies au paragraphe 7.2.1 de la [RFC3748]. Les revendications de sécurité pour EAP-TLS sont les suivantes :

Mécanisme d'authentification	**Certificats**
Négociation de suite de chiffrement :	Oui [4]
Authentification mutuelle :	Oui [1]
Protection de l'intégrité :	Oui [1]
Protection contre la répétition :	Oui [1]
Confidentialité :	Oui [2]
Déduction de clé :	Oui
Force de clé :	[3]
Protection contre l'attaque de dictionnaire :	Oui
Reconnexion rapide :	Oui
Lien cryptographique :	non applicable
Indépendance de session :	Oui [1]
Fragmentation :	Oui
Lien de canal :	Non

Notes :

[1] Une preuve formelle de la sécurité de EAP-TLS utilisé avec [IEEE-802.11] est fournie dans [He]. Cette preuve repose sur l'hypothèse que les paires de clés privées utilisées par l'homologue et le serveur EAP ne sont pas partagées avec d'autres parties ou applications. Par exemple, un serveur d'authentification d'extrémité arrière qui prend en charge EAP-TLS NE DEVRAIT PAS utiliser le même certificat qu'avec https.

[2] La confidentialité est une caractéristique facultative décrite au paragraphe 2.1.4.

[3] La Section 5 du BCP 86 [RFC3766] offre un avis sur la taille en bits de sous groupe de module RSA ou Diffie-Hellman (DH) et d'algorithme de signature numérique (DSA, *Digital Signature Algorithm*) requise, pour un certain niveau de résistance à l'attaque. Par exemple, une clé RSA de 2048 bits est recommandée pour fournir une force de clé équivalente à 128 bits. L'institut américain des normes et technologies (NIST, *National Institute of Standards and Technology*) offre aussi des avis sur les tailles de clé appropriées dans [SP800-57].

[4] EAP-TLS hérite des caractéristiques de négociation sûre de suite de chiffrement de TLS, incluant la négociation de fonction de déduction de clé quand utilisé avec TLS v1.2 [RFC5246].

5.2 Identités d'homologue et de serveur

Le nom de l'homologue EAP-TLS (Peer-Id) représente l'identité à utiliser pour les besoins de contrôle d'accès et de comptabilité. Le Server-Id représente l'identité du serveur EAP. Ensemble, le Peer-Id et le Server-Id désignent les entités impliquées dans la déduction de la MSK/EMSK.

Dans EAP-TLS, le Peer-Id et le Server-Id sont déterminés à partir des champs Subject ou subjectAltName dans les certificats d'homologue et de serveur. Pour les détails, voir le paragraphe 4.1.2.6 de la [RFC3280]. Lorsque le champ subjectAltName est présent dans le certificat de l'homologue ou du serveur, le Peer-Id ou Server-Id DOIT être réglé au contenu du subjectAltName. Si les informations de désignation du sujet ne sont présentes que dans l'extension subjectAltName d'un certificat d'homologue ou de serveur, alors le champ subject DOIT être une séquence vide et l'extension subjectAltName DOIT être critique.

Lorsque l'identité de l'homologue représente un hôte, un subjectAltName de type dnsName DEVRAIT être présent dans le certificat de l'homologue. Lorsque l'identité de l'homologue représente un utilisateur et pas une ressource, un subjectAltName de type rfc822Name DEVRAIT être utilisé, conformément à la grammaire pour l'identifiant d'accès réseau (NAI) défini au paragraphe 2.1 de la [RFC4282]. Si un dnsName ou rfc822Name n'est pas disponible, un autre type de champ (par exemple, un subjectAltName de type ipAddress ou uniformResourceIdentifier) PEUT être utilisé.

Une identité de serveur va normalement représenter un hôte, et non un utilisateur ou une ressource. Par suite, un subjectAltName de type dnsName DEVRAIT être présent dans le certificat de serveur. Si un dnsName n'est pas disponible, d'autres types de champs (par exemple, un subjectAltName de type ipAddress ou uniformResourceIdentifier) PEUT être utilisé.

Les mises en œuvre conformes qui génèrent de nouveaux certificats avec des identifiants d'accès réseau (NAI) DOIVENT utiliser le rfc822Name dans le champ de nom de sujet de remplacement pour décrire une telle identité. L'utilisation du champ Nom du sujet pour contenir un nom distinctif relatif (RDN, *Relative Distinguished Name*) emailAddress est déconseillé, et NE DOIT PAS être utilisé. Le champ Nom du sujet PEUT contenir d'autres RDN pour représenter l'identité du sujet.

Lorsque il n'est pas vide, le champ Nom du sujet DOIT contenir un nom distinctif (DN, *distinguished name*) X.500. Si les informations de désignation du sujet sont présentes seulement dans le champ Nom du sujet d'un certificat d'homologue et si l'identité de l'homologue représente un hôte ou un appareil, le champ Nom du sujet DEVRAIT contenir un RDN CommonName (CN) ou un RDN serialNumber. Si les informations de désignation du sujet sont présentes seulement dans le champ Nom du sujet d'un certificat de serveur, le champ Nom du sujet DEVRAIT alors contenir un RDN CN ou un RDN serialNumber.

Il est possible que plus d'un champ subjectAltName soit présent dans le certificat d'un homologue ou d'un serveur en plus d'un nom distinctif de sujet vide ou non. Les mises en œuvre de EAP-TLS qui prennent en charge l'exportation du Peer-Id et Server-Id DEVRAIENT exporter tous les champs subjectAltName dans les Peer-Id ou Server-Id, et DEVRAIENT aussi exporter un champ de nom distinctif de sujet non vide dans les Peer-Id ou Server-Id. Tous les Peer-Id et Server-Id exportés sont considérés comme valides.

Les mises en œuvre de EAP-TLS qui prennent en charge l'exportation de Peer-Id et Server-Id DEVRAIENT exporter les Peer-Id et Server-Id dans le même ordre que celui dans lequel ils apparaissent dans le certificat. Un tel ordre canonique va aider aux opérations de comparaison et va permettre d'utiliser ces identifiants pour la déduction de clé si c'est réputé utile. Cependant, l'ordre des champs dans le certificat NE DEVRAIT PAS être utilisé pour le contrôle d'accès.

5.3 Validation de certificat

Comme le serveur EAP-TLS est normalement connecté à l'Internet, il DEVRAIT prendre en charge la validation du certificat de l'homologue en utilisant la validation de chemin conforme à la [RFC3280], incluant la capacité de restituer les certificats intermédiaires qui peuvent être nécessaires pour valider le certificat de l'homologue. Pour les détails, voir le paragraphe 4.2.2.1 de la [RFC3280].

Lorsque le serveur EAP-TLS est incapable de restituer les certificats intermédiaires, soit il va devoir être pré-configuré avec les certificats intermédiaires nécessaires pour achever la validation de chemin, soit il va s'appuyer sur l'homologue EAP-TLS pour fournir ces informations au titre de la prise de contact TLS (voir le paragraphe 7.4.6 de la [RFC4346]).

À la différence du serveur EAP-TLS, l'homologue EAP-TLS peut ne pas avoir la connexité à l'Internet. Donc, le serveur EAP-TLS DEVRAIT fournir sa chaîne de certificat entière moins la racine pour faciliter la validation de certificat par l'homologue. L'homologue EAP-TLS DEVRAIT prendre en charge la validation du certificat du serveur en utilisant la validation de chemin conforme à la [RFC3280].

Une fois qu'une session TLS est établie, les mises en œuvre d'homologue et de serveur EAP-TLS DOIVENT valider que les identités représentées dans le certificat sont appropriées et autorisées pour l'usage avec EAP-TLS. Le processus d'autorisation utilise le contenu des certificats ainsi que d'autres informations de contexte. Alors que les exigences d'autorisation varient d'un déploiement à l'autre, il est RECOMMANDÉ que les mises en œuvre soient capables d'autoriser sur la base du Peer-Id et Server-Id EAP-TLS déterminées comme décrit au paragraphe 5.2.

Dans le cas de l'homologue EAP-TLS, cela implique de s'assurer que le certificat présenté par le serveur EAP-TLS était destiné à être utilisé comme certificat de serveur. Les mises en œuvre DEVRAIENT utiliser l'extension d'usage de clé étendu (voir le paragraphe 4.2.1.13 de la [RFC3280]) et s'assurer qu'au moins une des conditions suivantes est vraie :
1) Le producteur du certificat n'a pas inclus d'identifiant d'usage de clé étendu dans le certificat.
2) Le producteur a inclus l'identifiant anyExtendedKeyUsage dans le certificat (paragraphe 4.2.1.13 de la [RFC3280]).
3) Le producteur a inclus l'identifiant id-kp-serverAuth dans le certificat (paragraphe 4.2.1.13 de la [RFC3280]).

Lorsque elles effectuent cette comparaison, les mises en œuvre DOIVENT suivre les règles de validation spécifiées au paragraphe 3.1 de la [RFC2818]. Dans le cas du serveur, cela implique de s'assurer que le certificat présenté par

l'homologue EAP-TLS était destiné à être utilisé comme certificat de client. Les mises en œuvre DEVRAIENT utiliser l'extension d'usage de clé étendu (paragraphe 4.2.1.13 de la [RFC3280]) et s'assurer qu'au moins une des conditions suivantes est vraie :
1) Le producteur du certificat n'a pas inclus d'identifiant d'usage de clé étendu dans le certificat.
2) Le producteur a inclus l'identifiant anyExtendedKeyUsage dans le certificat (paragraphe 4.2.1.13 de la [RFC3280]).
3) Le producteur a inclus l'identifiant id-kp-clientAuth dans le certificat (paragraphe 4.2.1.13 de la [RFC3280]).

5.4 Révocation de certificat

Les certificats sont des assertions d'identité de longue durée. Donc, il est important pour les mises en œuvre de EAP-TLS d'être capables de vérifier si ces assertions ont été révoquées.

Les mises en œuvre d'homologue et de serveur EAP-TLS DOIVENT prendre en charge l'utilisation des listes de révocation de certificats (CRL, *Certificate Revocation List*) ; voir les détails au paragraphe 3.3 de la [RFC3280]. Les mises en œuvre d'homologue et de serveur EAP-TLS DEVRAIENT aussi prendre en charge le protocole d'état de certificat en ligne (OCSP, *Online Certificate Status Protocol*) décrit dans la [RFC2560]. Les messages OCSP sont normalement beaucoup plus courts que les CRL, ce qui peut raccourcir les temps de connexion en particulier dans les environnements où la bande passante est restreinte. Alors que les serveurs EAP-TLS sont normalement connectés à l'Internet durant la conversation EAP, un homologue EAP-TLS peut ne pas avoir la connexité à l'Internet jusqu'à l'achèvement de l'authentification.

Dans le cas où l'homologue initie un tunnel volontaire de couche 2 en utilisant PPTP [RFC2637] ou L2TP [RFC2661], l'homologue va normalement avoir déjà une interface PPP et la connexité à l'Internet établies au moment de l'initiation du tunnel.

Cependant, dans le cas où l'homologue EAP-TLS tente d'obtenir l'accès au réseau, il ne va pas avoir la connexité au réseau et n'est donc pas capable de vérifier la révocation de certificat avant la fin de l'authentification et tant que la connexité au réseau n'est pas disponible. Pour cette raison, les homologues et serveurs EAP-TLS DEVRAIENT mettre en œuvre les messages Demande d'état de certificat, comme décrit au paragraphe 3.6 de la [RFC4366]. Pour permettre la vérification de révocation dans les situations où les serveurs ne prennent pas en charge les messages Demande d'état de certificat, et où la connexité au réseau n'est pas disponible avant l'achèvement de l'authentification, les mises en œuvre d'homologues DOIVENT aussi prendre en charge la vérification de la révocation de certificat après l'achèvement de l'authentification et la disponibilité de la connexité au réseau, et elles DEVRAIENT utiliser cette capacité par défaut.

5.5 Attaques de modification de paquet

La protection de l'intégrité des paquets EAP-TLS ne s'étend pas aux champs d'en-tête EAP (Code, Identifiant, Longueur) ou aux champs Type ou Fanions. Par suite, ces champs peuvent être modifiés par un attaquant.

Dans la plupart des cas, la modification des champs Code ou Identifiant va seulement résulter en une attaque de déni de service. Cependant, un attaquant peut ajouter des données supplémentaires à un paquet EAP-TLS afin de le rendre plus long que ce qui est impliqué par le champ Longueur. Les homologues, authentificateurs, ou serveurs EAP qui ne vérifient pas cela pourraient être vulnérables à un débordement de mémoire tampon.

Il est aussi possible à un attaquant de modifier les champs Type ou Fanions. En modifiant le champ Type, un attaquant pourrait causer la négociation d'une méthode EAP fondée sur TLS plutôt qu'une autre. Par exemple, le champ Type EAP-TLS (13) pourrait être changé pour indiquer une autre méthode EAP fondée sur TLS. Sauf si la méthode EAP fondée sur TLS de remplacement utilise une formule différente de déduction de clé, il est possible qu'une conversation de méthode EAP altérée par un interposé puisse avoir lieu jusqu'à l'achèvement sans détection. Sauf si les politiques de choix de suite de chiffrement sont identiques pour toutes les méthodes EAP fondées sur TLS qui utilisent la même formule de déduction de clé, il sera possible à un attaquant de monter avec succès une attaque en dégradation, causant l'utilisation par l'homologue d'une suite de chiffrement ou méthode EAP fondée sur TLS inférieure.

6. Références

6.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2560] M. Myers, R. Ankney, A. Malpani, S. Galperin et C. Adams, "Protocole d'état de certificat en ligne d'infrastructure de clé publique X.509 pour l'Internet - OCSP", juin 1999. *(P.S.) (Remplacée par* RFC6960)

[RFC2818] E. Rescorla, "HTTP sur TLS", mai 2000. *(Information ; remplacée par* RFC9110*)*

[RFC3268] P. Chown, "Suites de chiffrement de la norme de chiffrement évolué (AES) pour la sécurité de la couche Transport (TLS)", juin 2002. (*Obsolète, voir* RFC5246) *(P.S.)*

[RFC3280] R. Housley, W. Polk, W. Ford et D. Solo, "Profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", avril 2002. *(Obsolète, voir* RFC5280)

[RFC3748] B. Aboba et autres, "Protocole extensible d'authentification (EAP)", juin 2004. *(P.S., MàJ par* RFC5247)

[RFC4282] B. Aboba et autres, "L'identifiant d'accès réseau", décembre 2005. *(P.S., Remplacée par* RFC7542)

[RFC4346] T. Dierks et E. Rescorla, "Protocole de sécurité de la couche Transport (TLS) version 1.1", avril 2006. *(Remplace* RFC2246 *; Remplacée par* RFC5246 *; MàJ par* RFC4366, 4680, 4681, 5746, 6176, 7465, 7507, 7919)

[RFC4366] S. Blake-Wilson et autres, "Extensions de sécurité de la couche Transport (TLS)", avril 2006. (*Obsolète,* RFC5246) *(P.S.)*

6.2 Références pour information

[IEEE-802.1X] Institut des ingénieurs en électricité et en électronique, "Réseaux locaux et de zone métropolitaine : contrôle d'accès réseau fondé sur l'accès", Norme IEEE 802.1X-2004, décembre 2004.

[IEEE-802.11] "Technologie de l'information - télécommunications et échanges d'informations entre systèmes-- réseaux locaux et de zone métropolitaine – Exigences spécifiques -- Partie 11 : spécifications du contrôle d'accès au support de LAN sans fil et de couche physique", Norme IEEE 802.11-2007.

[IEEE-802.16e] Institut des ingénieurs en électricité et en électronique, "Technologie de l'information - télécommunications et échanges d'informations entre systèmes-- réseaux locaux et de zone métropolitaine – Exigences spécifiques -- Partie 16 : interface radio pour systèmes d'accès sans fil à haut débit fixes et mobiles : amendement pour les couches physique et de contrôle d'accès au support pour le fonctionnement combiné fixe et mobile dans les bandes autorisées", Norme IEEE 802.16e, août 2005.

[He] He, C., Sundararajan, M., Datta, A., Derek, A. and J. Mitchell, "A Modular Correctness Proof of IEEE 802.11i et TLS", CCS '05, November 7-11, 2005, Alexandria, Virginia, USA

[RFC1661] W. Simpson, éditeur, "Protocole point à point (PPP)", STD 51, juillet 1994. *(MàJ par la* RFC2153)

[RFC2548] G. Zorn, "Attributs Microsoft spécifiques du fabricant pour RADIUS", mars 1999. *(Information)*

[RFC2637] K. Hamzeh, et autres, "Protocole de tunnelage point à point (PPTP)", juillet 1999.

[RFC2661] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn et B. Palter, "Protocole de tunnelage de couche 2 "L2TP"", *(P.S.)*

[RFC2716] B. Aboba, D. Simon, "Protocole d'authentification des TLS d'EAP dans PPP" octobre 1999. (*Obs., voir* RFC5216) *(Exp.)*

[RFC3766] H. Orman, P. Hoffman, "Détermination de la force des clés publiques utilisées pour l'échange de clés symétriques", avril 2004. (BCP0086)

[RFC4017] D. Stanley et autres, "Exigences de méthode pour le protocole d'authentification extensible (EAP) pour les LAN sans fil", mars 2005. *(Information)*

[RFC4284] F. Adrangi et autres, "Conseils de choix d'identité pour le protocole d'authentification extensible (EAP)", janvier 2006. *(Info.)*

[RFC5246] T. Dierks, E. Rescorla, "Version 1.2 du protocole de sécurité de la couche Transport (TLS)", août 2008. *(P.S. ; remplace* RFC3268, 4346, 4366 ; *MàJ* RFC4492 ; *rendue obsolète par la* RFC8446)

[RFC5247] B. Aboba et autres, "Cadre de gestion des clés du protocole d'authentification extensible (EAP)", août 2008. (*P. S. ;MàJ* RFC3748 *; MàJ par* RFC8940*)*

[RFC7406] H. Schulzrinne, et autres, "Extensions à l'architecture de services d'urgence pour traiter les appareils non authentifiés et non autorisés", décembre 2014. *(Information)*

[SP800-57] National Institute of Standards et Technology, "Recommendation for Key Management", Special Publication 800-57, mai 2006.

Remerciements

Merci à Terence Spies, Mudit Goel, Anthony Leibovitz, et Narendra Gidwani de Microsoft, à Glen Zorn de NetCube, Joe Salowey de Cisco, et Pasi Eronen de Nokia pour les utiles discussions sur ces problèmes.

Appendice A. Changements par rapport à la RFC 2716

Cet appendice fait la liste des changements majeurs entre la [RFC2716] et le présent document. Les changements mineurs, incluant de style, de grammaire, d'orthographe, et rédactionnels, ne sont pas mentionnés ici.

- Comme EAP est maintenant utilisé avec diverses couches inférieures, et pas seulement PPP pour lequel il a d'abord été conçu, la mention de PPP est restreinte aux situations relatives à un comportement spécifique de PPP et la référence est faite aux autres couches inférieures comme IEEE 802.11, IEEE 802.16, etc.

- Le document cite maintenant TLS v1.1 comme référence normative (Section 1 et paragraphe 6.1).

- La section de terminologie a été mise à jour pour refléter les definitions de la [RFC3748] (paragraphe 1.2) et le cadre de gestion de clé EAP [RFC5247] (paragraphe 1.2).

- L'utilisation pour l'accès d'un homologue non authentifié est précisée (paragraphe 2.1.1).

- La confidentialité est prise en charge comme caractéristique facultative (paragraphe 2.1.4).

- Il n'est plus recommandé que l'identité présentée dans la EAP-Response/Identity soit comparée à l'identité fournie dans le certificat de l'homologue (paragraphe 2.2).

- La hiérarchie de clés EAP-TLS est définie, en utilisant la terminologie de la [RFC3748]. Cela inclut des formules pour le calcul des TEK ainsi que des MSK, EMSK, IV, et Session-Id (paragraphe 2.3).

- Des suites de chiffrement TLS obligatoires et recommandées sont fournies. L'utilisation de la négociation de suite de chiffrement TLS pour déterminer la suite de chiffrement de couche inférieur est interdite (paragraphe 2.4).

- Le bit S (début) n'est pas établi dans un paquet de réponse EAP (paragraphe 3.2).

- Un paragraphe a été ajouté sur la revendication de sécurité et un avis sur la force de clé est fourni (paragraphe 5.1).

- Peer-Id et Server-Id sont définis (paragraphe 5.2) et des exigences pour la validation (paragraphe 5.3) et la révocation (paragraphe 5.4) de certificat sont fournies.

- Les attaques de modification de paquet sont décrites (paragraphe 5.5).

- Les exemples ont été mis à jour pour refléter les messages normaux envoyés dans les scénarios décrits. Par exemple, lorsque l'authentification mutuelle est effectuée, le serveur EAP-TLS est montré demandant un certificat de client et l'homologue est montré fournissant un message certificate_verify. Un exemple de confidentialité est fourni, et deux exemples fautifs d'échec de reprise de session ont été supprimés.

Adresse des auteurs

Dan Simon
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
USA
téléphone : +1 425 882 8080
mél : dansimon@microsoft.com

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
USA
téléphone: +1 425 706 6605
mél : bernarda@microsoft.com

Ryan Hurst
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
USA
téléphone : +1 425 882 8080
mél : rmh@microsoft.com

Déclaration complète de droits de reproduction