Groupe de travail Réseau
Request for Comments : 5433
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

T. Clancy, LTS
H. Tschofenig, Nokia Siemens Networks
février 2009

Protocole d'authentification extensible - méthode généralisée de clé pré-partagée (EAP-GPSK)

Statut du présent mémoire
Le présent document spécifie un protocole de l'Internet sur la voie de la normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Notice de droits de reproduction
Copyright (c) 2009 IETF Trust et les personnes identifiées comme auteurs du document. Tous droits réservés.

Le présent document est soumis au BCP 78 et aux dispositions légales de l'IETF Trust qui se rapportent aux documents de l'IETF (http://trustee.ietf.org/license-info) en vigueur à la date de publication de ce document. Prière de revoir ces documents avec attention, car ils décrivent vos droits et obligations par rapport à ce document.

Résumé
Le présent mémoire définit une méthode du protocole d'authentification extensible (EAP, *Extensible Authentication Protocol*) appelée clé pré partagée EAP généralisée (EAP-GPSK, *EAP Generalized Pre-Shared Key*). Cette méthode est un protocole léger d'authentification de clé pré partagée qui prend en charge l'authentification mutuelle et la déduction de clé.

Table des matières

1. Introduction

La clé pré partagée EAP généralisée (EAP-GPSK, *EAP Generalized Pre-Shared Key*) est une méthode EAP qui définit une technique d'authentification de clé pré partagée généralisée. L'authentification mutuelle est réalisée par un échange fondé sur le nom occasionnel qui est sécurisé par une clé pré partagée.

EAP-GPSK vise un grand nombre d'objectifs de conception dans l'intention d'être applicable à une large gamme de scénarios d'usage. Les principaux buts de la conception de EAP-GPSK sont :

Simplicité : EAP-GPSK devrait être facile à mettre en œuvre.

Modèle de sécurité : EAP-GPSK a été conçu selon un modèle de menace où l'attaquant a le plein contrôle sur le canal de communication. Ce modèle de menace EAP est présenté au paragraphe 7.1 de la [RFC3748].

Efficacité : EAP-GPSK n'utilise pas la cryptographie à clé publique et s'appuie complètement sur la cryptographie symétrique. La restriction aux calculs de cryptographie symétrique permet de moindres frais généraux de calcul. Donc, EAP-GPSK est léger et convient bien pour tous les types d'appareils, en particulier ceux avec des contraintes de puissance de traitement, de mémoire, et de batterie. De plus, il cherche à minimiser le nombre d'allers-retours.

Souplesse : EAP-GPSK offre la souplesse cryptographique. Au début, le serveur EAP propose une liste de suites de chiffrement. Le client en choisit alors une. La version actuelle de EAP-GPSK inclut deux suites de chiffrement, mais d'autres pourront facilement être ajoutées.

Extensibilité : la conception de EAP-GPSK permet d'échanger en toute sécurité des informations entre l'homologue EAP et le serveur EAP en utilisant les champs de données protégés. Ces champs pourraient, par exemple, être utilisés pour échanger les informations de lien de canal ou pour fournir la prise en charge de la confidentialité de l'identité.

2. Terminologie

Dans le présent document, plusieurs mots sont utilisés pour signifier les exigences de la spécification. Ces mots sont en majuscules. Les mots clés "DOIT", "NE DOIT PAS", "EXIGÉ", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "PEUT", et "FACULTATIF" dans ce document sont à interpréter comme décrit dans la [RFC2119].

Cette section décrit les diverses variables et fonctions utilisées dans la méthode EAP-GPSK.

Variables :
CSuite_List : chaîne d'octets faisant la liste des suites de chiffrement disponibles (longueur variable).
CSuite_Sel : suite de chiffrement choisie par l'homologue (6 octets).
ID_Peer : identifiant d'accès réseau (NAI, *Network Access Identifier*) [RFC4282] de l'homologue.
ID_Server : identité du serveur comme chaîne opaque.
KS : entier représentant la taille de la clé d'entrée, en octets, de la suite de chiffrement CSuite_Sel choisie. La taille de clé est un des paramètres de la suite de chiffrement.
ML : entier représentant la longueur du résultat du code d'authentification de message (MAC, *Message Authentication*

Code) en octets, de la suite de chiffrement CSuite_Sel choisie.
PD_Payload : Données portées dans la charge utile de données protégées.
PD_Payload_Block : Bloc de PD_Payload possiblement multiples portées par un paquet GPSK.
PL : entier représentant la longueur de PSK en octets (2 octets). PL DOIT être supérieur ou égal à KS.
RAND_Peer : entier aléatoire généré par l'homologue (32 octets).
RAND_Server : entier aléatoire généré par le serveur (32 octets).

Opérations :
A || B : enchaînement des chaînes d'octets A et B.
A**B : A à la puissance B.
truncate(A,B) : retourne les B premiers octets de A.
ENC_X(Y) : chiffrement du message Y avec une clé symétrique X, en utilisant un chiffrement de bloc défini.
KDF-X(Y) : fonction de déduction de clé qui génère un nombre arbitraire d'octets de résultat en utilisant le secret X et le germe Y.
longueur(X) : fonction qui retourne la longueur de l'entrée X en octets, codée comme un entier de 2 octets dans l'ordre des octets du réseau.
MAC_X(Y) : code d'authentification de message chiffré calculé sur Y avec la clé symétrique X.
SEC_X(Y) : SEC est une fonction qui fournit la protection de l'intégrité fondée sur la suite de chiffrement choisie. La fonction SEC utilise l'algorithme défini par la suite de chiffrement choisie et l'applique au contenu de message Y avec la clé X. En bref, SEC_X(Y) = Y || MAC_X(Y).
X[A..B] : notation représentant les octets A à B du dispositif d'octets X où le premier octet du dispositif a l'indice zéro.

Les abréviations suivantes sont utilisées pour le matériel de chiffrement :
EMSK *(Extended Master Session Key)* clé de session maîtresse étendue : est exportée par la méthode EAP (64 octets).
MK *(Master Key)* : clé maîtresse spécifique de la session entre l'homologue et le serveur EAP d'où toutes les autres clés de session de la méthode EAP sont déduites (KS octets).
MSK *(Master Session Key)* clé de session maîtresse : exportée par la méthode EAP (64 octets).
PK : clé de session générée à partir de la MK et utilisée durant l'échange de protocole pour chiffrer les données protégées (KS octets).
PSK : clé à long terme partagée entre l'homologue et le serveur (PL octets).
SK : clé de session générée à partir de la MK et utilisée durant l'échange de protocole pour démontrer la connaissance de la PSK (KS octets).

3. Vue d'ensemble

Le cadre EAP (voir le paragraphe 1.3 de la [RFC3748]) définit trois étapes de base qui se produisent durant l'exécution d'une conversation EAP entre l'homologue EAP, l'authentificateur, et le serveur EAP.

1. La première phase, découverte, est traitée par le protocole sous-jacent, par exemple, IEEE 802.1X comme utilisé par [IEEE 802.11].

2. La phase d'authentification EAP avec EAP-GPSK est définie dans ce document.

3. Les phases de distribution d'association sûre et d'association sûre sont traitées différemment selon le protocole sous-jacent.

EAP-GPSK effectue l'authentification mutuelle entre l'homologue EAP ("homologue") et le serveur EAP ("serveur") sur la base d'une clé pré partagée (PSK, *Pre-Shared Key*). Le protocole consiste en des échanges de messages (GPSK-1, ..., GPSK-4) dans lesquels les deux côtés échangent des noms occasionnels et leur identité, et calculent et échangent un code d'authentification de message (MAC, *Message Authentication Code*) sur les valeurs précédemment échangées, chiffrées avec la clé pré partagée. Ce MAC est considéré comme une preuve de possession de la clé pré partagée. Deux autres messages, à savoir GPSK-Fail *(échec GPSK)* et GPSK-Protected-Fail *(échec de GPSK protégé)* sont utilisés pour traiter les situations d'erreur.

Un échange de protocole réussi est montré à la Figure 1.

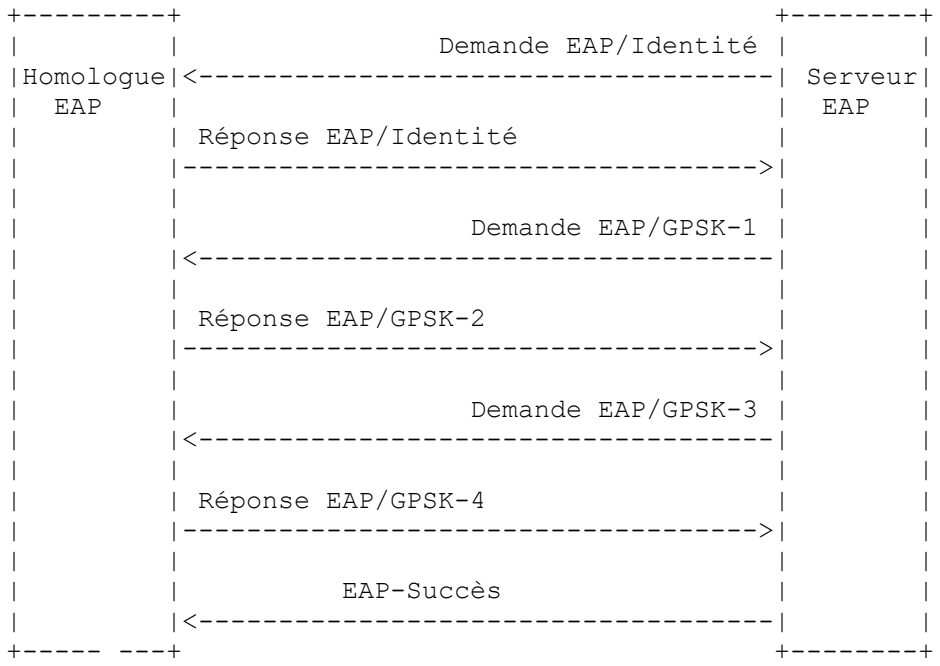


Figure 1 : Échange EAP-GPSK réussi

Le protocole EAP-GPSK complet est le suivant :

GPSK-1 : ID_Server, RAND_Server, CSuite_List

GPSK-2 : SEC_SK(ID_Peer, ID_Server, RAND_Peer, RAND_Server, CSuite_List, CSuite_Sel,
[ENC_PK(PD_Payload_Block)])

GPSK-3 : SEC_SK(RAND_Peer, RAND_Server, ID_Server, CSuite_Sel, [ENC_PK(PD_Payload_Block)])

GPSK-4 : SEC_SK([ENC_PK(PD_Payload_Block)])

Le serveur EAP commence EAP-GPSK en choisissant un nombre aléatoire RAND_Server et en codant les suites de chiffrement prises en charge dans CSuite_List. Une suite de chiffrement consiste en un algorithme de chiffrement, une fonction de déduction de clé, et un code d'authentification de message.

Dans GPSK-1, le serveur EAP envoie son identité ID_Server, un nombre aléatoire RAND_Server, et une liste des suites de chiffrement prises en charge, CSuite_List. La décision de quelles suites de chiffrement offrir et quelle suite de chiffrement prendre dépend de la politique et de la mise en œuvre et sort donc du domaine d'application du présent document.

Dans GPSK-2, l'homologue envoie son identité ID_Peer et un nombre aléatoire RAND_Peer. De plus, il répète les paramètres reçus du message GPSK-1 (ID_Server, RAND_Server, CSuite_List) et la suite de chiffrement choisie. Il calcule un code d'authentification de message sur tous les paramètres transmis.

Le serveur EAP vérifie le code d'authentification de message reçu et la cohérence des identités, noms occasionnels, et paramètres de suite de chiffrement transmis dans GPSK-1. En cas de réussite de la vérification, le serveur EAP calcule un code d'authentification de message sur le paramètre de session et le retourne à l'homologue (dans un GPSK-3). Dans GPSK-2 et GPSK-3, l'homologue EAP et le serveur EAP ont la possibilité d'échanger des paramètres de données protégées chiffrées.

L'homologue vérifie le code d'authentification de message reçu et la cohérence des identités, noms occasionnels, et paramètres de suite de chiffrement transmis dans GPSK-2. Si la vérification est réussie, GPSK-4 est préparé. Ce message peut facultativement contenir les paramètres de données protégées de l'homologue.

À réception de GPSK-4, le serveur traite tous les PD_Payload_Block inclus. Puis, le serveur EAP envoie un message de succès EAP pour indiquer la réussite de l'authentification.

4. Déduction de clé

EAP-GPSK fournit la déduction de clé conformément aux exigences des [RFC3748] et [RFC5247]. Noter que cette section donne une description abstraite de la procédure de déduction de clé qui doit être mise en place avec une suite de chiffrement spécifique.

L'accréditif à long terme partagé entre homologue et serveur EAP DEVRAIT être une forte clé PSK pré partagée d'au moins 16 octets, bien que sa longueur et son entropie soient variables. Bien qu'il soit possible d'utiliser un mot ou phrase de passe, il N'EST PAS RECOMMANDÉ de le faire car EAP-GPSK est vulnérable aux attaques de dictionnaire.

Durant une authentification EAP-GPSK, une clé maîtresse MK, une clé de session SK, et une clé de chiffrement de données protégées PK (si on utilise une suite de chiffrement chiffrante) sont déduites en utilisant la suite de chiffrement KDF spécifiée et les données échangées durant l'exécution du protocole, à savoir "RAND_Peer || ID_Peer || RAND_Server || ID_Server", appelée "inputString" dans sa forme abrégée.

En cas de succès, EAP-GPSK déduit et exporte une MSK et une EMSK, chacune de 64 octets.

La notation suivante est utilisée : KDF-X(Y, Z)[A..B], dans laquelle,
X est la longueur, en octets, du résultat désiré,
Y est une clé secrète,
Z est la inputString,
[A..B] extrait la chaîne d'octets en commençant par l'octet A et finissant par l'octet B du résultat de la fonction KDF.

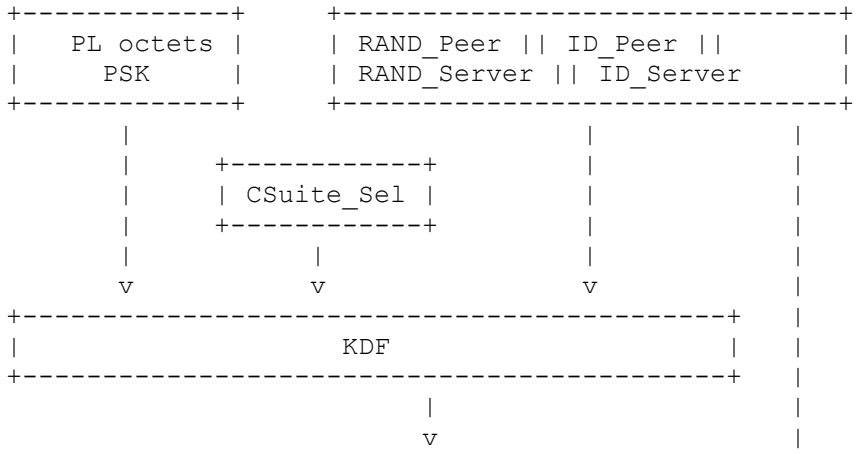
Ce matériel de chiffrement est déduit en utilisant la KDF spécifiée par la suite de chiffrement comme suit :
- inputString = RAND_Peer || ID_Peer || RAND_Server || ID_Server
- MK = KDF-KS(PSK[0..KS-1], PL || PSK || CSuite_Sel || inputString)[0..KS-1]
- MSK = KDF-{128+2*KS}(MK, inputString)[0..63]
- EMSK = KDF-{128+2*KS}(MK, inputString)[64..127]
- SK = KDF-{128+2*KS}(MK, inputString)[128..127+KS]
- PK = KDF-{128+2*KS}(MK, inputString)[128+KS..127+2*KS] (si on utilise une suite de chiffrement chiffrante)

La valeur de PL (longueur de la PSK en octets) est codée comme un entier de 2 octets dans l'ordre des octets du réseau. On rappelle que KS est la longueur en octets de la taille de clé d'entrée de la suite de chiffrement.

De plus, le cadre de chiffrement EAP [RFC5247] exige la définition d'un identifiant de méthode, d'un identifiant de session, d'un identifiant d"homologue, et d'un identifiant de serveur. Ces valeurs sont définies comme :
- Method-ID = KDF-16(PSK[0..KS-1], "Method ID" || EAP_Method_Type || CSuite_Sel || inputString)[0..15]
- Session-ID = EAP_Method_Type || Method_ID
- Peer-ID = ID_Peer
- Server-ID = ID_Server

EAP_Method_Type se réfère à la valeur de code de un octet de type EAP alloué par l'IANA.

La Figure 2 décrit la procédure de déduction de clé de EAP-GPSK.



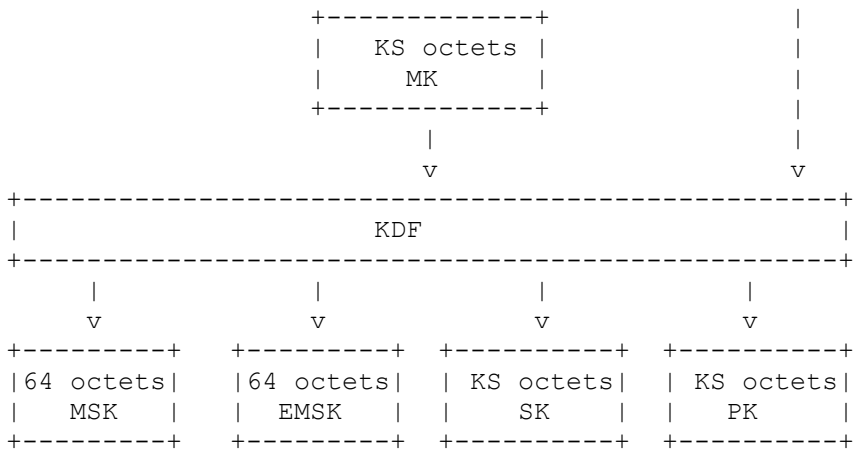


Figure 2 : Déduction de clé EAP-GPSK

5. Gestion de clé

Afin d'être interopérables, les PSK doivent être entrées de la même façon sur l'homologue et le serveur. L'interface de gestion pour entrer les PSK DOIT prendre en charge d'entrer les PSK jusqu'à 64 octets de long comme des chaînes ASCII et codées en hexadécimal.

De plus, le ID_Peer et ID_Server DOIVENT être provisionnés avec la PSK. La validation de ces valeurs est par une comparaison octet par octet. L'interface de gestion DEVRAIT prendre en charge d'entrer des octets non ASCII pour le ID_Peer et ID_Server jusqu'à 254 octets de long. Pour plus d'informations, il est conseillé au lecteur de voir le paragraphe 2.4 de la [RFC4282].

6. Suites de chiffrement

La conception de EAP-GPSK permet des algorithmes de chiffrement et des tailles de clé, appelées des suites de chiffrement, négociées durant le fonctionnement du protocole. La capacité de spécifier des suites de chiffrement fondées sur le bloc et fondées sur le hachage est offerte. L'extensibilité est fournie avec l'introduction de nouvelles suites de chiffrement ; le présent document spécifie un ensemble initial. La colonne CSuite/spécifieur de la Figure 3 identifie de façon univoque une suite de chiffrement.

Pour une suite de chiffrement spécifique du fabricant, les quatre premiers octets sont le numéro d'entreprise spécifique du fabricant qui contient la valeur allouée par l'IANA de "Codes d'entreprise privée de SMI de gestion de réseau" (voir [ENTNUM]) codée dans l'ordre des octets du réseau. Les deux derniers octets sont alloués par le fabricant pour la suite de chiffrement spécifique. Un code de fabricant de 0x00000000 indique des suites de chiffrement normalisées par l'IETF dans un registre tenu par l'IANA.

Les suites de chiffrement ci après sont spécifiées dans le présent document (on rappelle que KS est la longueur de clé d'entrée de la suite de chiffrement en octets, et que ML est la longueur du MAC résultant en octets):

CSuite/ spécifieur	KS	Chiffrement	ML	Intégrité / KDF MAC	Fonction de déduction de clé
0x0001	16	AES-CBC-128	16	AES-CMAC-128	GKDF
0x0002	32	NULL	32	HMAC-SHA256	GKDF

Figure 3 : Suites de chiffrement

Ciphersuite 1, qui se fonde sur la norme de chiffrement évolué (AES, *Advanced Encryption Standard*) comme une primitive cryptographique, DOIT être mise en œuvre. Le présent document spécifie aussi une seconde suite de chiffrement, qui PEUT être mise en œuvre. Les deux suites de chiffrement définies dans ce document utilisent la fonction de déduction de clé généralisée (GKDF, *Generalized Key Derivation Function*) comme définie à la Section 7. Les aspects suivants doivent être considérés pour s'assurer que la PSK utilisée comme entrée à la GKDF est suffisamment longue :

1. La PSK utilisée avec la suite de chiffrement 1 DOIT être de 128 bits. Les clés plus longues que 128 bits vont être tronquées.

2. La PSK utilisée avec la suite de chiffrement 2 DOIT être de 256 bits. Les clés plus longues que 256 bits vont être tronquées.

3. Il est RECOMMANDÉ que des clés de 256 bit soient provisionnées dans tous les cas pour fournir assez d'entropie pour toutes les suites de chiffrement actuelles et de nombreuses futures possibles.

Les suites de chiffrement définies à l'avenir qui utilisent la GKDF doivent spécifier une taille minimum de PSK (comme il est fait avec les suites de chiffrement mentionnées dans ce document).

7. Fonction généralisée de déduction de clé (GKDF)

Chaque suite de chiffrement doit spécifier une fonction de déduction de clé. Les suites de chiffrement définies dans ce document utilisent la fonction de déduction de clé généralisée (GKDF) qui utilise la fonction de MAC définie dans la suite de chiffrement. De futures suites de chiffrement pourront utiliser toute autre KDF formellement spécifiée qui prend comme arguments une clé et une valeur de germe, et produit au moins 128+2*KS octets en sortie.

GKDF a la structure suivante :

GKDF-X(Y, Z)

X : longueur, en octets, du résultat désiré
Y : clé secrète
Z : chaîne d'entrée

```
GKDF-X (Y, Z)
  {
   n = entier plafond de ( X / ML ) ;          /* détermine le nombre de blocs  de sortie */
   résultat = "" ;
   pour i = 1 à n {
     résultat = résultat || MAC_Y (i || Z);
   }

   retourne tronqué(résultat, X)
  }
```

Noter que la variable "i" dans M_i est représentée par une valeur de 2 octets ans l'ordre des octets du réseau.

8. Règles de traitement des suites de chiffrement

8.1 Suite de chiffrement n° 1

8.1.1 Chiffrement

Avec cette suite de chiffrement, toute la cryptographie est construite autour d'une seule primitive cryptographique, AES-128 [AES]. Dans les trames de données protégées, AES-128 est utilisé dans le mode de fonctionnement de chaînage de bloc de chiffrement (CBC, *Cipher Block Chaining*) (voir [CBC]). Cette méthode EAP utilise le chiffrement dans une seule charge utile, dans la charge utile de données protégées (voir le paragraphe 9.4).

En abrégé, le mode CBC procède comme suit. La IV est OUXée avec le premier bloc de texte en clair avant d'être chiffrée.

Ensuite, pour les blocs successifs, le bloc de chiffrement précédent est OUXé avec le texte en clair courant, avant qu'il soit chiffré.

8.1.2 Intégrité

La suite de chiffrement 1 utilise CMAC comme code d'authentification de message. CMAC est recommandé par le NIST. Entre autres avantages, CMAC est capable de fonctionner avec des messages de longueur arbitraire. Une description détaillée de CMAC se trouve dans [CMAC].

L'application suivante est utilisée : AES-CMAC-128(SK, entrée) note le MAC d'entrée sous la clé SK où entrée se réfère au contenu suivant :
- o Paramètre dans SEC_SK(paramètre) dans le message GPSK-2
- o Paramètre dans SEC_SK(paramètre) dans le message GPSK-3
- o Paramètre dans SEC_SK(paramètre) dans le message GPSK-4

8.2 Suite de chiffrement n° 2

8.2.1 Chiffrement

La suite de chiffrement 2 n'inclut pas d'algorithme pour le chiffrement. Avec un algorithme de chiffrement NUL, le chiffrement est défini par :

E_X(Y) = Y

Quand on utilise cette suite de chiffrement, les données échangées dans le bloc de données protégées ne sont pas chiffrées. Donc, ce mode NE DOIT PAS être utilisé si des informations confidentielles apparaissent dans le bloc de données protégées.

8.2.2 Intégrité

La suite de chiffrement 2 utilise la fonction de MAC chiffrée HMAC, avec l'algorithme de hachage SHA256 (voir la [RFC4634]).

Pour la protection de l'intégrité, l'application suivante est utilisée :

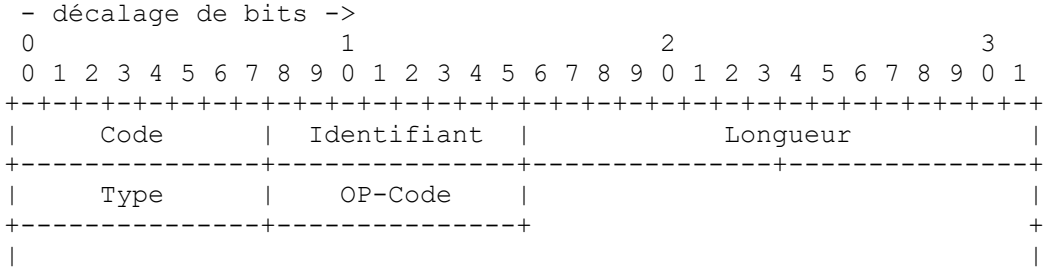
HMAC-SHA256(SK, entrée) note le MAC d'entrée sous la clé SK où entrée se réfère au contenu suivant :
- o Paramètre dans SEC_SK(paramètre) dans le message GPSK-2
- o Paramètre dans SEC_SK(paramètre) dans le message GPSK-3
- o Paramètre dans SEC_SK(paramètre) dans le message GPSK-4

9. Formats de paquet

Cette Section définit le format de paquet des messages EAP-GPSK

9.1 Format d'en-tête

L'en-tête EAP-GPSK a la structure suivante :



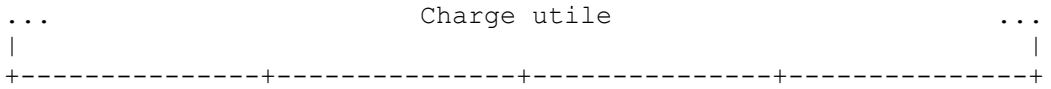


Figure 4 : En-tête EAP-GPSK

Les champs Code, Identifiant, Longueur, et Type font tous partie de l'en-tête EAP et sont définis dans la [RFC3748]. Le champ Type dans l'en-tête EAP DOIT être la valeur allouée par l'IANA pour EAP-GPSK.

Le champ OP-Code a une des 6 valeurs suivantes :
- o 0x00 : réservé
- o 0x01 : GPSK-1
- o 0x02 : GPSK-2
- o 0x03 : GPSK-3
- o 0x04 : GPSK-4
- o 0x05 : GPSK-échec
- o 0x06 : GPSK-protégé-échec

Toutes les autres valeurs de ce champ OP-Code sont disponibles via enregistrement par l'IANA.

9.2 Formatage de suite de chiffrement

Les suites de chiffrement sont codées comme des arrangements de 6 octets. Les quatre premiers octets indiquent le champ CSuite/Vendeur. Pour les suites de chiffrement spécifiques de vendeur, cela représenté le numéro d'entreprise du vendeur et contient la valeur allouée par l'IANA de "Codes d'entreprise privée de SMI de gestion de réseau" (voir [ENTNUM]) codé dans l'ordre des octets du réseau. Les deux derniers octets indiquent le champ CSuite/spécifieur, qui identifie la suite de chiffrement particulière. La valeur de 4 octets de CSuite/Vendeur de 0x00000000 indique des suites de chiffrement allouées par l'IETF.

Graphiquement, cela est représenté par :

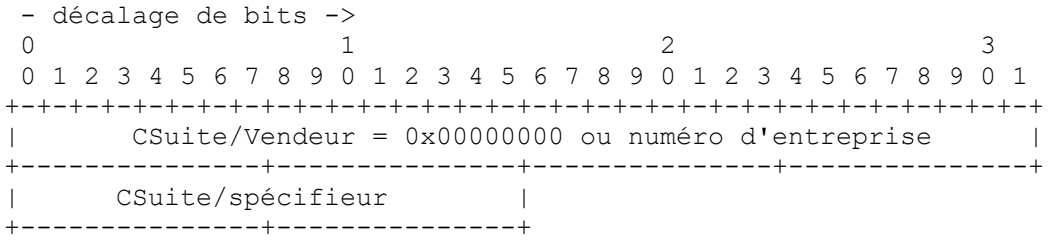


Figure 5 : Formatage de suite de chiffrement

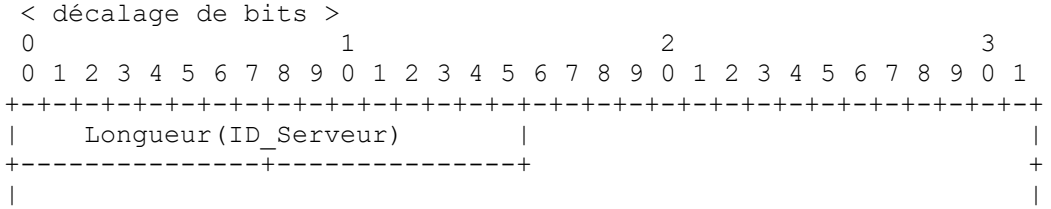
CSuite_Sel est codé comme une paire de suites de chiffrement de 6 octets CSuite/Vendeur et CSuite/spécifieur.

CSuite_List est un arrangement d'octets de longueur variable de suites de chiffrement. Il est codé en enchaînant les valeurs codées de suites de chiffrement. Sa longueur en octets DOIT être un multiple de 6.

9.3 Formatage de charge utile

Le formatage de charge utile se fonde sur la description d'échange de protocole de la Section 3.

Le format de la charge utile GPSK-1 est défini comme suit :



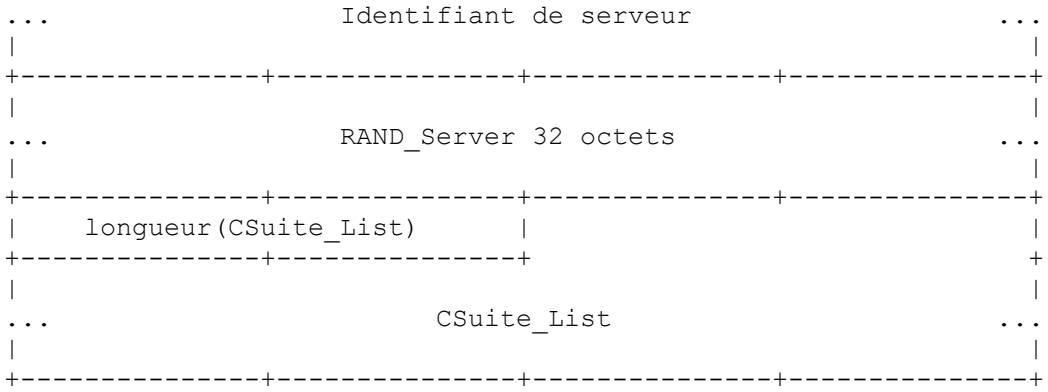


Figure 6 : Charge utile GPSK-1

Le format de charge utile GPSK-2 est défini comme suit :

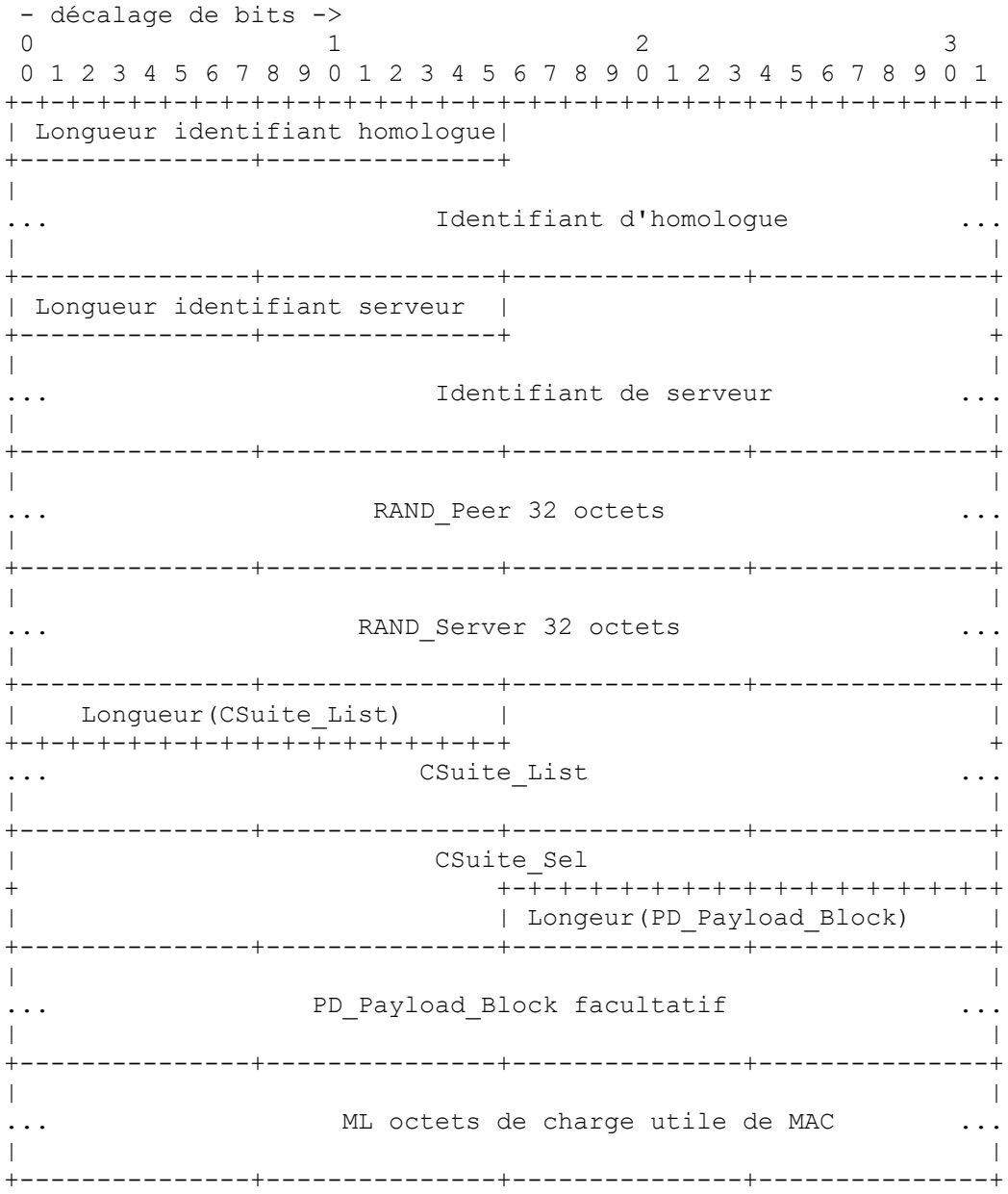


Figure 7 : Charge utile GPSK-2

Si la charge utile facultative de données protégées n'est pas incluse, alors longueur(PD_Payload_Block)=0 et la charge utile PD est exclue. la charge utile de MAC couvre le paquet entier, depuis la longueur d'identifiant d'homologue jusqu'au bloc de charge utile PD facultatif.

La charge utile GPSK-3 est définie comme suit :

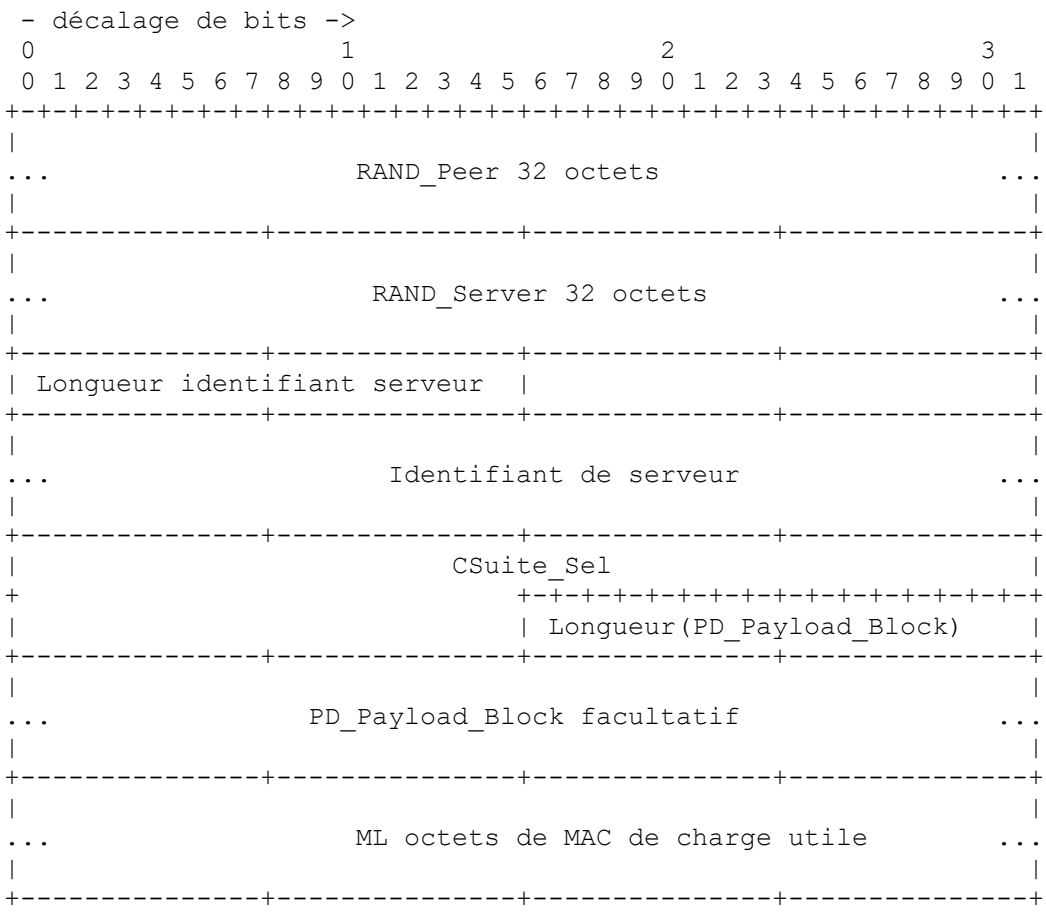


Figure 8 : Charge utile GPSK-3

Si la charge utile facultative de données protégées n'est pas incluse, alors Longueur(PD_Payload_Block)=0 et la charge utile de PD est exclue. Le MAC de charge utile couvre le paquet entier de RAND_Peer au PD_Payload_Block facultatif.

Le format de charge utile GPSK-4 est défini comme suit :

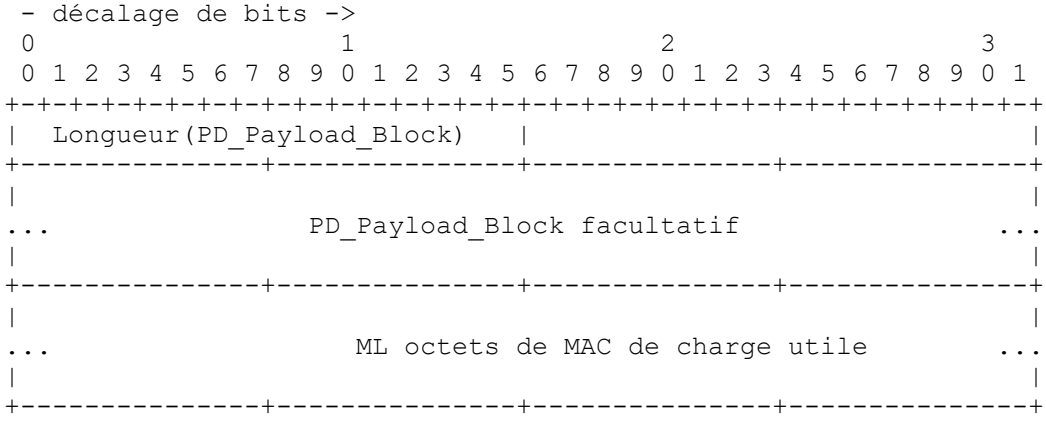


Figure 9 : Charge utile GPSK-4

Si la charge utile facultative Données protégées n'est pas incluse, alors longueur(PD_Payload_Block)=0 et la charge utile PD est exclue. Le MAC DOIT toujours être inclus, sans considération de la présence de PD_Payload_Block. Le MAC de charge utile couvre le paquet entier, depuis la longueur de PD_Payload_Block jusqu'au PD_Payload_Block facultatif.

Le format de charge utile GPSK-échec est défini comme suit :

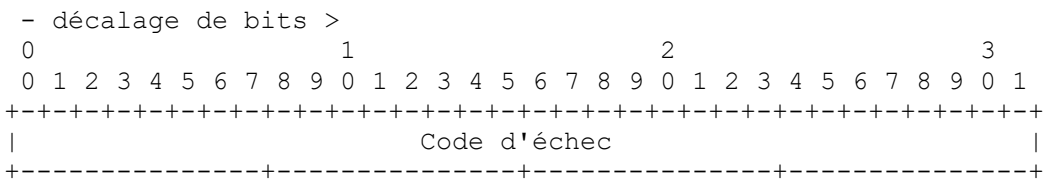


Figure 10 : Charge utile GPSK-échec

Le format de charge utile GPSK-protégé-échec est défini comme suit :

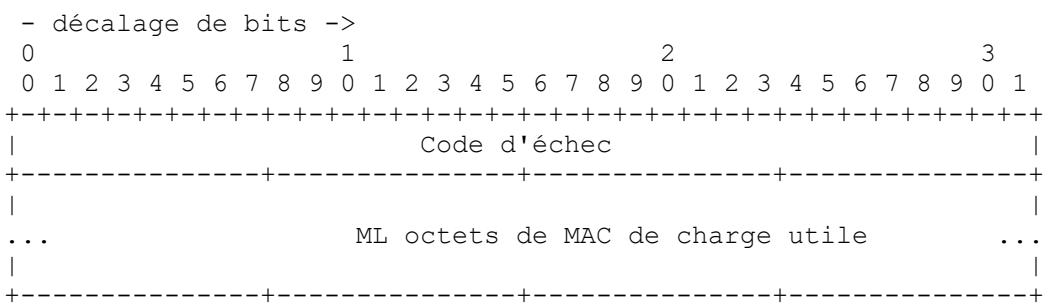


Figure 11 : Charge utile GPSK-protégé-échec

Le champ Code d'échec est une des trois valeurs suivantes, mais peut être étendu :
- o 0x00000000 : réservé
- o 0x00000001 : PSK pas trouvée
- o 0x00000002 : Échec d'authentfication
- o 0x00000003 : Échec d'autorisation

Toutes les autres valeurs de ce champ sont disponibles via enregistrement par l'IANA.

"PSK pas trouvée" indique qu'une clé pour un utilisateur particulier n'a pas pu être localisée, rendant l'authentification impossible. "Échec d'authentification" indique une défaillance de MAC due à une discordance de PSK. "Échec d'autorisation" indique que alors que la PSK utilisée est correcte, l'utilisateur n'est pas autorisé à se connecter.

9.4 Données protégées

Les blocs de données protégées sont un mécanisme générique pour que l'homologue et le serveur échangent des données en toute sécurité. Si la suite de chiffrement spécifiée a une primitive de chiffrement NULLE, alors ce canal offre seulement l'authenticité, pas la confidentialité.

Ces charges utiles sont codées comme l'enchaînement de triplets de type-longueur-valeur (TLV) appelés des PD_Payload.

Les valeurs de type sont codées comme une chaîne de 6 octets et représentées par un champ de 4 octets de fabricant et un champ de 2 octets de spécifieur. Le champ de fabricant indique le type comme spécifié standard ou spécifique de fabricant.

Si ces quatre octets sont 0x00000000, alors la valeur est spécifiée standard, et toutes les autres valeurs représentent un numéro d'entreprise spécifique du fabricant.

Le champ spécifieur indique le type actuel. pour le champ de fabricant 0x00000000, le champ spécifieur est tenu par l'IANA. Pour tout autre champ de fabricant, le champ spécifieur est tenu par le fabricant.

Les champs de longueur sont spécifiés comme des entiers de 2 octets dans l'ordre des octets du réseau, reflètent seulement

la longueur de la valeur, et n'incluent pas la longueur des champs de type et de longueur.

Graphiquement, cela peut être décrit comme suit :

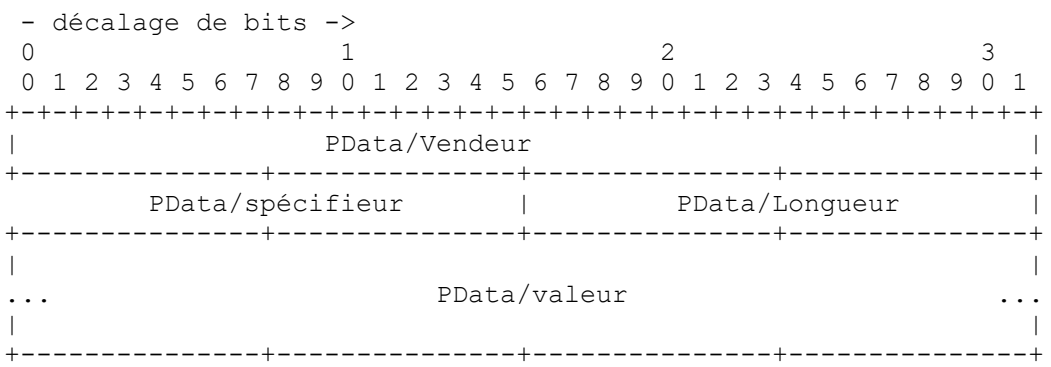


Figure 12 : Formatage de charge utile de données protégées (PD_Payload)

Ces PD_Payload sont enchaînés ensemble pour former un PD_Payload_Block. Si le CSuite_Sel inclut la prise en charge du chiffrement, alors le PD_Payload_Block inclut des champs qui spécifient une valeur d'initialisation (IV) et le bourrage nécessaire. Cela peut être décrit comme suit :

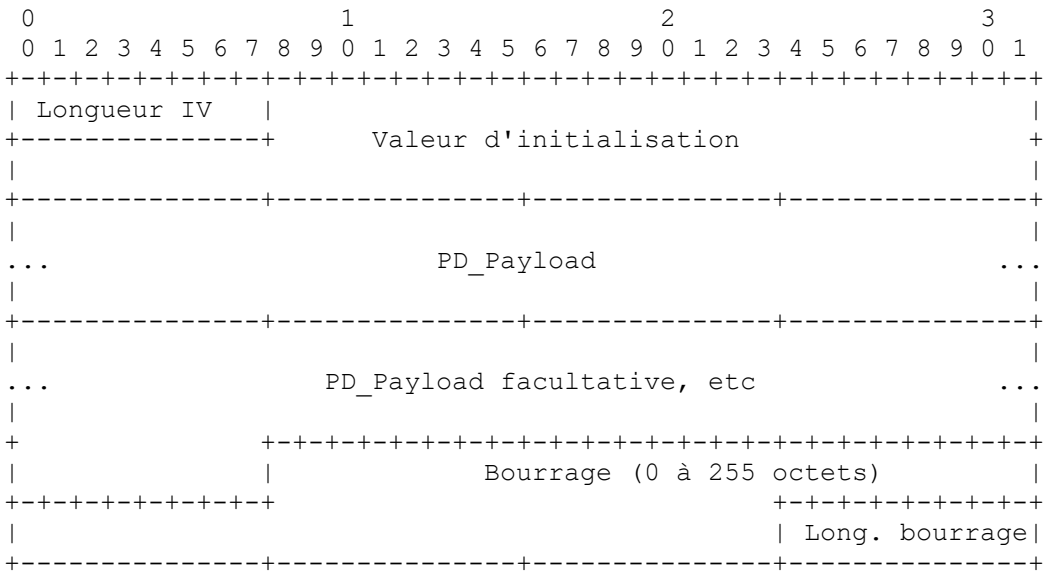


Figure 13 : Format de bloc de données protégées (PD_Payload_Block) si le chiffrement est pris en charge

La valeur d'initialisation est une valeur choisie au hasard dont la longueur est égale à la longueur d'IV spécifiée. La longueur requise est définie par la suite de chiffrement. Les receveurs DOIVENT accepter toute valeur. Les envoyeurs DEVRAIENT soit prendre cette valeur pseudo aléatoire et indépendante pour chaque message, soit utiliser le bloc de texte chiffré final du précédent message envoyé. Les envoyeurs NE DOIVENT PAS utiliser la même valeur pour chaque message, utiliser une suite de valeurs avec une faible distance de Hamming (par exemple, un numéro de séquence) ou utiliser du texte chiffré provenant d'un message reçu. Les IV devraient être choisies selon les exigences de sécurité du chiffrement sous-jacent. Si les données ne sont pas chiffrées, alors la longueur d'IV DOIT être 0. Si la suite de chiffrement n'exige pas d'IV, ou a un moyen auto-contenu de communiquer l'IV, alors le champ Longueur d'IV DOIT être 0. Dans ce cas, la définition de la suite de chiffrement définit comment l'IV est encapsulée dans la PD_Payload.

L'enchaînement des PD_Payload avec le bourrage et la longueur de bourrage sont toutes chiffrées en utilisant le chiffrement de bloc négocié. Si aucun chiffrement de bloc n'est spécifié, alors ces champs ne sont pas chiffrés.

Le champ Bourrage PEUT contenir toute valeur choisie par l'envoyeur. Pour les modes de chiffrement fondés sur le bloc, le bourrage DOIT avoir une longueur qui fait de la combinaison de l'enchaînement des PD_Payload, du bourrage, et de la

longueur de bourrage un multiple de la taille de bloc de chiffrement. Si la suite de chiffrement sous-jacente n'exige pas de bourrage (par exemple, un mode de chiffrement fondé sur le flux) ou si aucun chiffrement n'est utilisé, alors la longueur de bourrage DOIT quand même être présente et être 0.

Le champ Longueur de bourrage est la longueur du champ Bourrage. L'envoyeur DEVRAIT régler la longueur de bourrage à la valeur minimum qui fait de la combinaison des PD_Payload, Bourrage, et Longueur de bourrage un multiple de la taille de bloc (dans le cas des modes de chiffrement fondés sur le bloc) mais le receveur DOIT accepter toute longueur qui résulte en un alignement approprié. Ce champ est chiffré avec le chiffrement négocié.

Si la suite de chiffrement négociée ne prend pas en charge le chiffrement, alors le champ IV DOIT être de longueur 0 et le champ Bourrage DOIT être de longueur 0. La longueur des champs IV et Longueur de bourrage DOIT quand même être présente, et contenir la valeur 0. La raison d'exiger quand même les champs de longueur est de permettre des mises en œuvre modulaires où le traitement du chiffrement est indépendant du traitement de la charge utile. Cela est décrit dans la figure suivante.

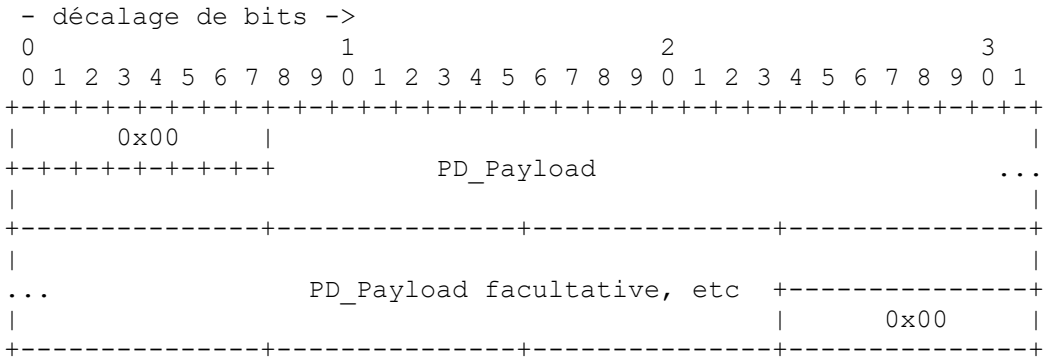


Figure 14 : Format de bloc de données protégées (PD_Payload_Block) sans chiffrement

Pour le champ PData/Vendeur 0x00000000, le champ PData/spécifieur suivant est défini :
o 0x0000 : réservé

Toutes les autres valeurs de ce champ sont disponibles via enregistrement par l'IANA.

10. Règles de traitement de paquet

Cette Section définit comment l'homologue et le serveur EAP DOIVENT se comporter quand un paquet reçu est réputé invalide.

Tout paquet EAP-GPSK qui ne peut pas être analysé par l'homologue ou le serveur EAP DOIT être éliminé en silence. Un homologue ou serveur EAP qui reçoit un paquet inattendu (par exemple, un homologue EAP qui reçoit un GPSK-3 avant de recevoir un GPSK-1 ou avant de transmettre GPSK-2) DOIT éliminer en silence le paquet.

GPSK-1 ne contient pas de protection de MAC, donc pourvu qu'il soit analysé correctement, il DOIT être accepté par l'homologue. Si l'homologue EAP n'a pas de suite de chiffrement en commun avec le serveur ou décide que l'identifiant de serveur est celui d'un serveur d'authentification, autorisation, et comptabilité (AAA) auprès duquel il ne souhaite pas s'authentifier, l'homologue EAP DOIT répondre avec un EAP-NAK.

Pour GPSK-2, si l'identifiant d'homologue est pour un utilisateur inconnu, le serveur EAP DOIT envoyer un message GPSK-échec "PSK pas trouvée" ou "Échec d'authentification", selon sa politique. Si la validation de MAC échoue, le serveur DOIT transmettre un message GPSK-échec spécifiant "Échec d'authentification". Si le champ RAND_Server ou CSuite_List dans GPSK-2 ne correspond pas aux valeurs dans GPSK-1, le serveur DOIT éliminer en silence le paquet. Si la politique du serveur détermine que l'homologue n'est pas autorisé et si le MAC est correct, le serveur DOIT transmettre un message GPSK-protégé-échec indiquant "Échec d'autorisation", et éliminer le paquet reçu.

Un homologue qui reçoit un message GPSK-échec/GPSK-protégé-échec en réponse à un message GPSK-2 DOIT répéter le message GPSK-échec/GPSK-protégé-échec reçu. Ensuite, le serveur EAP retourne un EAP-échec après avoir reçu le message GPSK-échec/GPSK-protégé-échec pour finir correctement la conversation EAP. Si la validation de MAC sur un

paquet GPSK-protégé-échec échoue, alors le paquet reçu DOIT être éliminé en silence.

Pour GPSK-3, un homologue DOIT éliminer en silence les messages où les champs RAND_Peer, ID_Server, ou CSuite_Sel ne correspondent pas à ceux transmis dans GPSK-2. Un homologue EAP DOIT éliminer en silence tout paquet dont le MAC échoue.

Pour GPSK-4, un serveur DOIT éliminer en silence tout paquet dont le MAC échoue à la validation.

Si un échec de déchiffrement d'une charge utile protégée est détecté, le receveur DOIT éliminer en silence le paquet GPSK.

11. Exemple d'échanges de messages

Cette section montre quelques exemples de flux de messages.

Un échange de message EAP-GPSK réussi est montré à la Figure 1.

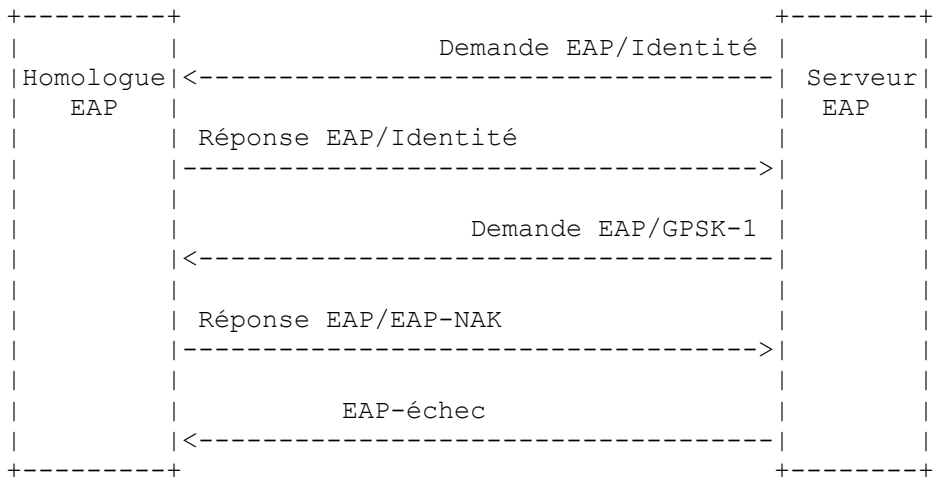


Figure 15 : EAP-GPSK : Échange non réussi (Identité de serveur AAA inacceptable ; ID_Server)

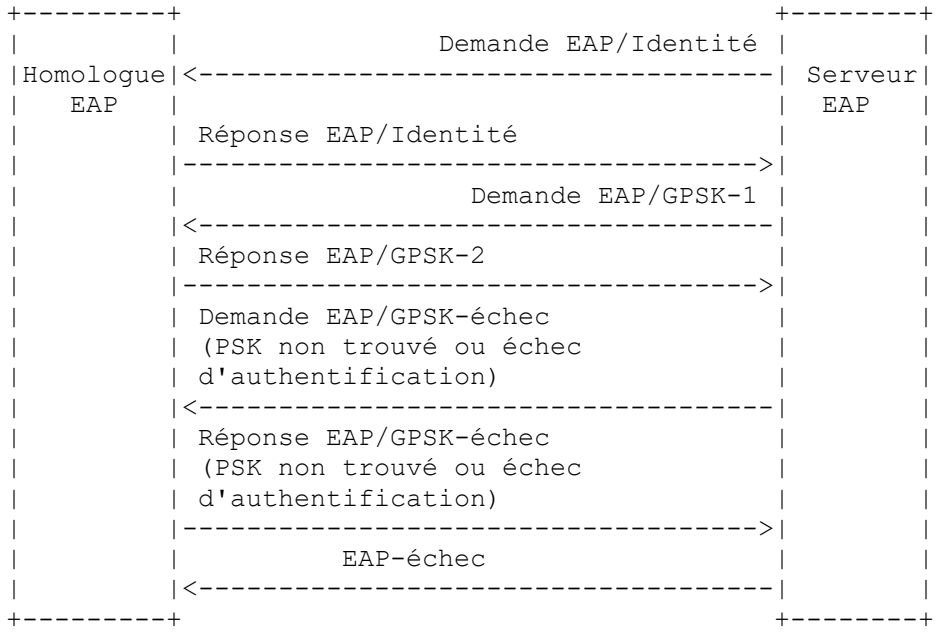


Figure 16 : EAP-GPSK : Échange non réussi (utilisateur inconnu)

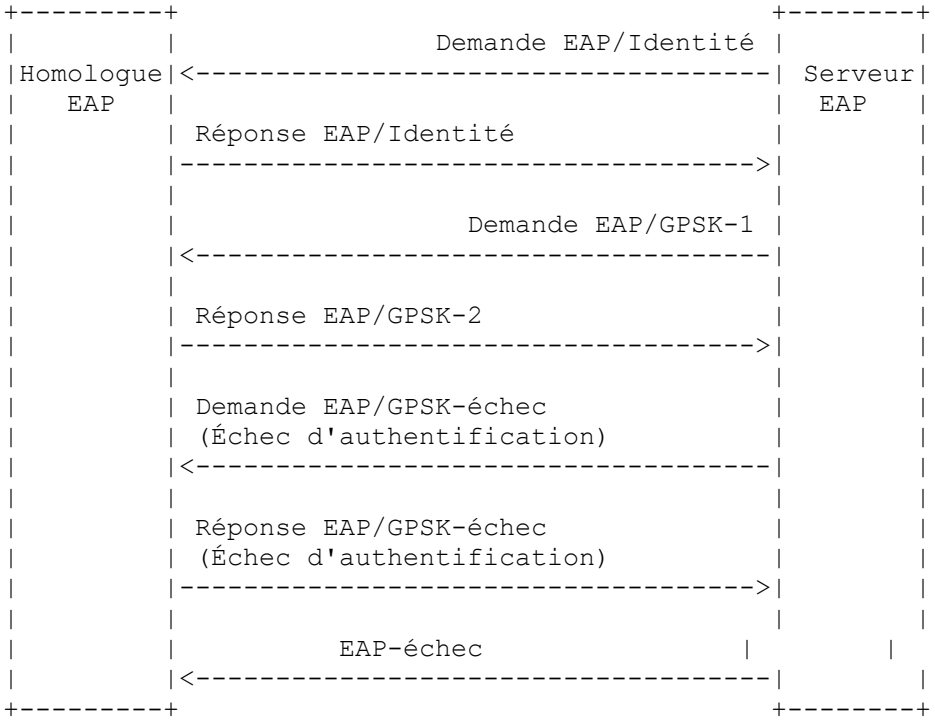


Figure 17 : EAP-GPSK : Échange non réussi (MAC invalide dans GPSK-2)

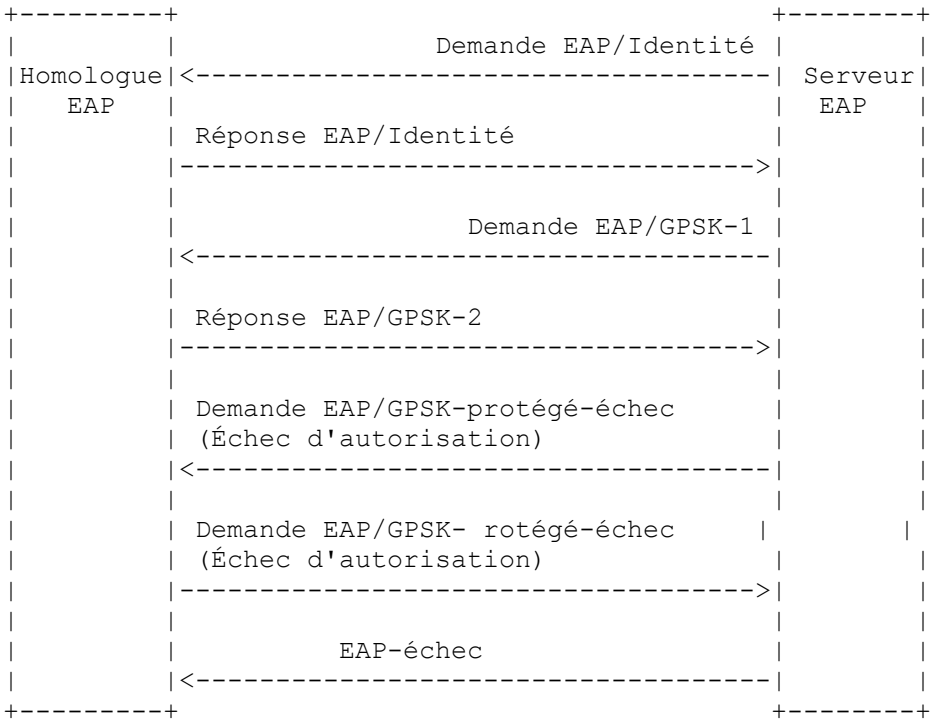


Figure 18 : EAP-GPSK : Échange non réussi (échec d'autorisation)

12. Considérations sur la sécurité

La [RFC3748] souligne plusieurs attaques possibles contre EAP car EAP lui-même ne fournit aucune sécurité.

Cette section discute les propriétés de sécurité revendiquées par EAP-GPSK ainsi que les vulnérabilités et les recommandations de sécurité dans le modèle de menaces de la [RFC3748].

12.1 Revendications de sécurité

Mécanisme d'authentification : clés partagées
Négociation de suite de chiffrement : oui (paragraphe 12.16)
Authentification mutuelle : oui (paragraphe 12.2)
Protection de l'intégrité : oui (paragraphe 12.4)
Protection contre la répétition : oui (paragraphe 12.5)
Confidentialité : non (paragraphes 12.15, 12.17)
Déduction de clé : oui (paragraphe 12.8)
Force de clé : variable (paragraphe 12.8)
Protection contre l'attaque de dictionnaire : non (paragraphe 12.7)
Reconnexion rapide : non (paragraphe 12.14)
Lien cryptographique : non applicable (paragraphe 12.18)
Indépendance de session : oui (paragraphe 12.10)
Fragmentation : non (paragraphe 12.12)
Lien de canal : extensible (paragraphe 12.13)

12.2 Authentification mutuelle

EAP-GPSK assure l'authentification mutuelle.

Le serveur croit que l'homologue est authentique quand il vérifie le MAC dans le message GPSK-2 ; l'homologue croit que le serveur est authentique quand il vérifie le MAC qu'il reçoit avec le message GPSK-3.

La clé utilisée pour l'authentification mutuelle est déduite sur la base de la PSK secrète à long terme, des noms occasionnels fournis par les deux parties, et d'autres paramètres. La PSK secrète à long terme doit fournir une entropie suffisante et donc une force suffisante. Les noms occasionnels (RAND_Peer et RAND_Server) doivent être frais et uniques pour chaque session. De cette façon, EAP-GPSK n'est pas différent des autres protocoles d'authentification fondés sur des clés pré partagées.

12.3 Indications de résultat protégé

EAP-GPSK prend en charge l'indication de résultats protégés via le message GPSK-Protected-Fail. Cela permet à un serveur de fournir des informations supplémentaires à l'homologue sur la raison de l'échec de la session, et de le faire de façon authentifiée (si possible). En particulier, le serveur peut indiquer le manque de PSK (compte non présent), l'échec de l'authentification (PSK incorrecte), ou l'échec de l'autorisation (compte désactivé ou non autorisé). Seul le troisième message pourrait être protégé en intégrité.

On devrait noter que ces options rendent plus facile le débogage du réseau et la découverte d'erreurs de compte, mais elles laissent aussi échapper des informations sur les comptes à des attaquants. Un attaquant peut déterminer si un ID_Peer particulier est ou non un utilisateur valide sur le réseau. Donc, les mises en œuvre devraient faire attention lorsque elles activent cette option sur leurs serveurs. Si elles sont dans un environnement où de telles attaques sont un problème, les capacités d'indication de résultat protégé devraient alors être désactivées.

12.4 Protection de l'intégrité

EAP-GPSK fournit la protection de l'intégrité sur la base des suites de chiffrement suggérées dans ce document. La protection de l'intégrité est une caractéristique minimale que chaque suite de chiffrement doit fournir.

12.5 Protection contre la répétition

EAP-GPSK fournit la protection contre la répétition de ses parties mutuellement authentifiées grâce à l'utilisation des nombres aléatoires RAND_Server et RAND_Peer. Comme RAND_Server fait 32 octets, on s'attend à avoir à enregistrer 2**64 (c'est-à-dire, approximativement 1.84*10**19) authentifications EAP-GPSK réussies avant qu'un jeu de protocole puisse être répété. Donc, EAP-GPSK fourni une protection contre la répétition de ses parties mutuellement authentifiées pour autant que RAND_Server et RAND_Peer sont choisis au hasard ; l'aléa est critique pour la protection contre la répétition. La [RFC4086] décrit les techniques de production de quantités aléatoires.

12.6 Attaques de réflexion

Des attaques de réflexion se produisent dans des protocoles d'authentification mutuelle bidirectionnels de défi/réponse, où un attaquant, quand l'authentificateur lui produit un défi, répond en produisant le même défi à l'authentificateur, obtenant ainsi la réponse, et en "reflétant" ensuite cette même réponse au défi original.

EAP-GPSK fournit la protection contre les attaques en réflexion parce que les formats de message sont différents pour le défi. Le protocole ne consiste pas en deux authentifications indépendantes, mais plutôt en authentifications étroitement couplées.

Noter aussi que EAP-GPSK ne fournit pas de protection de MAC du champ OP-Code, mais là encore, comme chaque message est de construction différente, il ne serait pas possible de changer le OP-Code d'un message valide et de l'avoir encore analysable et acceptable par le receveur.

12.7 Attaques de dictionnaire

EAP-GPSK s'appuie sur un secret partagé à long terme (la PSK) qui DEVRAIT être fondé sur au moins 16 octets d'entropie pour être pleinement sûr. Le protocole EAP-GPSK ne prend pas de disposition particulière pour assurer que les clés fondées sur des mots de passe sont utilisées de façon sûre. Les utilisateurs qui utilisent des mots de passe comme base de leur PSK ne sont pas protégés contre les attaques de dictionnaire. La déduction du secret partagé à long terme à partir d'un mot de passe est fortement déconseillée.

Le succès d'une attaque de dictionnaire contre EAP-GPSK dépend de la force du secret partagé à long terme (PSK) qu'il utilise. La PSK utilisée par EAP-GPSK DEVRAIT être tirée d'un réservoir de secrets grand d'au moins 2^128 bits et dont la distribution est uniformément aléatoire. Noter que cela n'implique pas la résistance aux attaques de dictionnaire -- seulement que la probabilité de succès d'une telle attaque est repoussée à une distance acceptable.

12.8 Déduction de clé et force de clé

EAP-GPSK prend en charge la déduction de clé comme montré à la Section 4.

Les clés utilisées dans EAP-GPSK sont toutes fondées sur la sécurité de la PSK génératrice. Les PSK DEVRAIENT avoir au moins 16 octets d'entropie. Indépendamment de l'échange de protocole (c'est-à-dire, sans connaître RAND_Peer et RAND_Server) les clés ont été déduites avec une entropie d'entrée suffisante pour les rendre aussi sûres que la longueur de clé de résultat de la KDF sous-jacente.

12.9 Résistance au déni de service

Trois formes d'attaque de déni de service (DoS) sont pertinentes pour ce document, à savoir (1) les attaques qui conduisent à l'allocation de grandes quantités d'état, (2) les attaques qui tentent d'empêcher la communication entre l'homologue et le serveur, et (3) les attaques contre les ressources de calcul.

Dans une conversation EAP-GPSK, le serveur doit maintenir l'état, à savoir les 32 octets de RAND_Server, quand il transmet le message GPSK-1 à l'homologue. Un adversaire pourrait inonder un serveur avec un grand nombre de tentatives de communication EAP-GPSK. Un serveur EAP peut donc s'assurer qu'un état établi se périme après une période relativement courte quand aucun autre message n'est reçu. Cela permet une sorte de collecte des ordures.

Le client doit conserver des informations d'état après avoir reçu le message GPSK-1. Pour empêcher une attaque en répétition, tout ce que le client a besoin de faire est de s'assurer que la valeur de RAND_Peer est cohérente entre GPSK-2 et GPSK-3. Le message GPSK-3 contient tout le matériel nécessaire pour recalculer le matériel de chiffrement. Donc, si un client choisit de mettre en œuvre ce mécanisme de protection contre le DoS côté client, il peut gérer RAND_Peer et CSuite_Sel sur la base du serveur pour les serveurs qu'il connaît, plutôt que message par message.

Les attaques qui perturbent la communication entre l'homologue et le serveur sont atténuées en éliminant en silence les messages qui ont un MAC invalide. Les attaques contre les ressources de calcul sont atténuées en exigeant seulement des opérations de chiffrement très légères durant chaque tour de protocole.

Les considérations sur la sécurité de EAP lui-même (voir le paragraphe 5.2 et la Section 7 de la [RFC3748]) sont aussi applicables à la présente spécification (par exemple, concernant les notifications fondées sur EAP).

12.10 Indépendance de session

Grâce à ses mécanismes de déduction de clés, EAP-GPSK fournit l'indépendance de session : les attaques passives (comme la capture de la conversation EAP) ou les attaques actives (incluant de compromettre la MSK ou la EMSK) ne permettent pas de compromettre les MSK ou EMSK précédentes ou suivantes. L'hypothèse que RAND_Peer et RAND_Server sont aléatoires est centrale pour la sécurité de EAP-GPSK en général et de l'indépendance de session en particulier.

12.11 Compromission de la PSK

EAP-GPSK ne fournit pas de secret parfait vers l'avant. La compromission de la PSK conduit à la compromission des sessions enregistrées passées.

La compromission de la PSK permet à l'attaquant de se faire passer pour l'homologue et le serveur, et permet à l'adversaire de compromettre de futures sessions.

EAP-GPSK ne fournit pas de protection contre un homologue légitime qui partage sa PSK avec un tiers. Une telle protection peut être fournie par des répertoires appropriés pour la PSK, dont le choix sort du domaine d'application de ce document. La PSK utilisée par EAP-GPSK doit seulement être partagée entre deux parties : l'homologue et le serveur. En particulier, cette PSK ne doit pas être partagée par un groupe d'homologues (par exemple, ceux avec des valeurs différentes de ID_Peer) communiquant avec le même serveur.

La PSK utilisée par EAP-GPSK doit être cryptographiquement séparée des clés utilisées par d'autres protocoles, autrement, la sécurité de EAP-GPSK peut être compromise.

12.12 Fragmentation

EAP-GPSK ne prend pas en charge la fragmentation et le réassemblage car la taille de message est relativement petite. Cependant, on devrait noter que cela impacte la longueur des charges utiles de données protégées qui peuvent être rattachées aux messages. Aussi, si la trame EAP fait plus que la MTU du transport sous-jacent, et si ce transport ne prend pas en charge la fragmentation, la trame ne va très probablement pas être transportée. Par conséquent, les mises en œuvre et les déploiements devraient veiller à s'assurer que les trames EAP-GPSK sont assez courtes pour fonctionner correctement sur le mécanisme de transport cible sous-jacent.

12.13 Lien de canal

Le présent document donne la capacité d'échanger des informations de lien de canal. Il ne définit cependant pas le codage des informations de lien de canal dans ce document.

12.14 Reconnexion rapide

EAP-GPSK ne fournit pas de capacité de reconnexion rapide car cette méthode est déjà à la limite inférieure (ou proche) du nombre d'allers-retours et des opérations cryptographiques.

12.15 Protection de l'identité

La protection de l'identité n'est pas spécifiée dans ce document. Des extensions pourront être définies qui améliorent ce protocole pour fournir cette caractéristique.

12.16 Négociation de suite de chiffrement protégée

EAP-GPSK fournit une négociation protégée de la suite de chiffrement via l'indication des suites de chiffrement disponibles par le serveur dans le premier message, et une confirmation par l'homologue dans le message suivant.

Noter cependant, que le message GPSK-2 peut facultativement contenir une charge utile, ENC_PK(PD_Payload_Block),

protégée avec un algorithme fondé sur une suite de chiffrement choisie avant que la liste de suites de chiffrement ait été authentifiée. Dans l'attaque en dégradation classique, un adversaire choisirait une suite de chiffrement qui soit assez faible pour pouvoir la casser en temps réel ou tenterait de désactiver en même temps la protection cryptographique. Cette dernière attaque n'est pas possible car toute suite de chiffrement définie pour EAP-GPSK doit au moins fournir l'authentification et la protection de l'intégrité. La protection de la confidentialité est facultative. Quand, à un moment futur, une suite de chiffrement contiendra des algorithmes qui peuvent être cassés en temps réel, une politique des homologues et du serveur devra alors indiquer qu'une telle suite de chiffrement ne doit pas être choisie par une des parties.

De plus, un adversaire peut modifier le choix de la suite de chiffrement pour que le client choisisse une suite de chiffrement qui ne fournit pas de protection de la confidentialité. Par suite, cela causerait la transmission en clair du contenu du PD_Payload_Block. Quand des concepteurs de protocole étendent EAP-GPSK pour porter des informations dans le PD_Payload_Block du message GPSK-2, il doit alors être indiqué si la protection de la confidentialité est obligatoire. Dans le cas où une telle extension exigerait une suite de chiffrement avec protection de la confidentialité, la politique de l'homologue ne doit alors pas transmettre l'information de cette extension dans le PD_Payload_Block du message GPSK-2. L'homologue peut, si possible, retarder la transmission de cet élément d'information au message GPSK-4 où la négociation de la suite de chiffrement a déjà été confirmée. En général, quand une suite de chiffrement est choisie qui ne fournit pas la protection de la confidentialité, alors les informations qui demandent la protection de la confidentialité ne doivent pas être incluses dans un des objets du PD_Payload_Block.

12.17 Confidentialité

Bien que EAP-GPSK fournisse la confidentialité dans ses charges utiles de données protégées, il ne peut pas revendiquer de le faire, selon le paragraphe 7.2.1 de la [RFC3748], car il ne prend pas en charge la protection de l'identité.

12.18 Lien cryptographique

Comme EAP-GPSK ne tunnelle pas d'autre méthode EAP, il ne met pas en œuvre de lien cryptographique.

13. Considérations relatives à l'IANA

L'IANA a alloué un nouveau type EAP pour EAP-GPSK (51).

L'IANA a créé un nouveau registre pour les suites de chiffrement, les types de données protégées, les codes d'échec, et les op-code. L'IANA a ajouté les suites de chiffrement, types de données protégées, codes d'échec, et op-codes spécifiés à ces registres comme défini ci-dessous. Les valeurs qui définissent des suites de chiffrement (fondées sur le bloc ou fondées sur le hachage) les charges utiles de données protégées, les codes d'échec, et les op-codes peuvent être ajoutées ou modifiées par revue de l'IETF [RFC5226].

La Figure 3 représente le contenu initial du registre "EAP-GPSK Ciphersuites". Le champ CSuite/Spécifieur fait 16 bits. Toutes les autres valeurs sont disponibles via enregistrement par l'IANA. Chaque suite de chiffrement doit fournir les règles de traitement et spécifier comment les algorithmes suivants sont instanciés : chiffrement, intégrité, déduction de clé, et longueur de clé.

Voici le contenu initial du registre "Charges utiles de connées protégées EAP-GPSK" :

o 0x0000 : réservé

Le champ PData/Spécifieur fait 16 bits, et toutes les autres valeurs sont disponibles via enregistrement par l'IANA. Chaque extension doit indiquer si la protection de la confidentialité pour la transmission entre l'homologue et le serveur EAP est obligatoire.

Voici le contenu initial du registre "Codes d'échec EAP-GPSK" :
o 0x00000000 : réservé
o 0x00000001 : PSK non trouvée
o 0x00000002 : Échec d'authentification
o 0x00000003 : Échec d'autorisation

Le champ Failure-Code fait 32 bits, et toutes les autres valeurs sont disponibles via enregistrement par l'IANA.

Voici le contenu initial du registre "Codes OP EAP-GPSK" :
- 0x00 : réservé
- 0x01 : GPSK-1
- 0x02 : GPSK-2
- 0x03 : GPSK-3
- 0x04 : GPSK-4
- 0x05 : GPSK-échec
- 0x06 : GPSK-Protégé-échec

Le champ OP-Code fait 8 bits, et toutes les autres valeurs sont disponibles via enregistrement par l'IANA.

14. Contributeurs

Ce travail est un effort conjoint de l'équipe de conception de mise à jour de la méthode EAP (EMU) du groupe de travail EMU qui a été créé pour développer un mécanisme fondé sur de forts secrets partagés qui satisfont les exigences des [RFC3748] et [RFC4017]. Les membres de l'équipe de conception étaient (par ordre alphabétique) : Jari Arkko, Mohamad Badra, Uri Blumenthal, Charles Clancy, Lakshminath Dondeti, David McGrew, Joe Salowey, Sharma Suman, Hannes Tschofenig, Jesse Walker

Finalement, nous tenons à remercier Thomas Otto de ses relectures, retours, et contributions de texte.

15. Remerciements

Nous tenons à remercier:
- Jouni Malinen et Bernard Aboba de leurs commentaires sur le document de juin 2006. Jouni Malinen a développé le premier prototype de mise en œuvre.
- Lakshminath Dondeti, David McGrew, Bernard Aboba, Michaela Vanderveen, et Ray Bell pour leurs apports aux discussions sur les suites de chiffrement entre juillet et août 2006.
- Lakshminath Dondeti pour ss relecture détaillée (envoyée à la liste de diffusion EMU le 12 juillet 2006).
- Sur la base d'une révision demandés par le NIST, Quynh Dang a suggéré des changements à la fonction GKDF (décembre 2006).
- Jouni Malinen et Victor Fajardo de leur revue en janvier 2007.
- Jouni Malinen pour ses suggestions concernant les exemples et la fonction de déduction de clé en février 2007.
- Bernard Aboba et Jouni Malinen pour leur révision de février 2007.
- Vidya Narayanan pour sa relecture de mars 2007.
- Pasi Eronen pour la relcture de l'IESG en mars et juillet 2008.
- Dan Harkins pour s relecture de juin 2008.
- Joe Salowey, le président du groupe de travail EMU, a fourni une révision du document en avril 2007. Jouni Malinen a aussi revu le document durant le même mois.
- Nous tenons à remercier Paul Rowe, Arnab Roy, le Prof. Andre Scedrov, et le Prof. John C. Mitchell de leur analyse de EAP-GPSK, de leurs apports à la fonction de déduction de clé, et pour avoir souligné une attaque de DoS côté client et une attaque en dégradation. Sur la base de ces apports, la fonction de déduction de clé a été modifiée et le texte des considérations sur la sécurité a été mis à jour.
- Finalement, nous tenons à remercier le président de notre groupe de travail, Joe Salowey, de son soutine et du temps passé aux discussions sur les questions ouvertes avec nous.

16. Références

16.1 Références normatives

[AES] National Institute of Standards and Technology, "Specification for the Advanced Encryption Standard (AES)", Federal Information Processing Standards (FIPS) 197, novembre 2001.

[CBC] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Encryption --

Methods and Techniques", Special Publication (SP) 800-38A, décembre 2001.

[CMAC] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication", Special Publication (SP) 800-38B, mai 2005.

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

[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)

[RFC4634] D. Eastlake 3rd, T. Hansen, "Algorithmes de hachage sécurisé aux USA (SHA et HMAC-SHA)", juillet 2006. *(Info.)*

[RFC5226] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, mai 2008. (*Remplace* RFC2434 *; remplacée par* RFC8126)

[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*)*

16.2 Références pour information

[IEEE 802.11] Norme IEEE 802.11-2007, "Information technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", mars 2007.

[ENTNUM] IANA, "SMI Network Management Private Enterprise Codes", Private Enterprise Numbers, <http://www.iana.org>.

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

[RFC4086] D. Eastlake 3rd, J. Schiller, S. Crocker, "Exigences d'aléa pour la sécurité", juin 2005, DOI 10.17487/RFC4086, (*Remplace* RFC1750) (BCP0106)

Adresse des auteurs

T. Charles Clancy
DoD Laboratory for Telecommunications Sciences
8080 Greenmead Drive
College Park, MD 20740
USA
mél : clancy@ltsnet.net

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland
mél : Hannes.Tschofenig@gmx.net