Groupe de travail Réseau
Request for Comments : 4359
Catégorie: Sur la voie de la normalisation

B. Weis, Cisco Systems
janvier 2006
Traduction Claude Brière de L'Isle

Utilisation des signatures RSA/SHA-1 au sein d'une charge utile de sécurité par encapsulation (ESP) et d'un en-tête d'authentification (AH)

Résumé
Le présent mémoire décrit l'utilisation de l'algorithme de signature numérique RSA comme algorithme d'authentification au sein de l'encapsulation de charge utile de sécurité IP révisée (ESP, *Encapsulating Security Payload*) comme décrit dans la RFC 4303 de l'en-tête d'authentification IP révisé (AH, *Authentication Header*) comme décrit dans la RFC 4302. L'utilisation d'un algorithme de signature numérique comme RSA, assure l'authentification de l'origine des données dans les applications quand une méthode de clé secrète (par exemple., HMAC) ne fournit pas cette propriété. Un exemple est l'utilisation de ESP et AH pour authentifier l'envoyeur d'un paquet IP en diffusion groupée.

Table des matières

1. Introduction....1
2. Algorithme et mode....2
2.1 Discussion de la taille de clé....3
3. Performance....3
4. Interaction avec le mécanisme de chiffrement ESP....3
5. Considérations de gestion de clé....4
6. Considérations sur la sécurité....4
6.1 Espionnage....4
6.2 Répétition....4
6.3. Insertion de message....5
6.4 Suppression....5
6.5 Modification....5
6.6 Interposition....5
6.7 Déni de service....5
7. Considérations relatives à l'IANA....5
8. Remerciements....6
9. Références....6
9.1 Références normatives....6
9.2 References pour information....6
Adresse de l'auteur....7
Déclaration complète de droits de reproduction....7

1. Introduction

Les en-têtes d'encapsulation de charge utile de sécurité (ESP, *Encapsulating Security Payload*) [RFC4303] et d'en-tête d'authentification (AH, *Authentication Header*) [RFC4302] peuvent être utilisés pour protéger aussi bien le trafic en envoi individuel que le trafic de groupe (par exemple, de diffusion groupée IPv4 et IPv6). Quand le trafic en envoi individuel est protégé entre une paire d'entités, les transformations HMAC (comme celles de la [RFC2404]) sont suffisantes pour prouver

l'authentification de l'origine des données. Un HMAC est une protection suffisante dans ce scénario parce que seulement les deux entités impliquées dans la communication ont accès à la clé, et la preuve de possession de la clé dans la construction HMAC authentifie l'envoyeur. Cependant, quand ESP et AH authentifient du trafic de groupe, cette propriété ne tient plus parce que tous les membres du groupe partagent la seule clé HMAC. Dans le cas du groupe, l'identité de l'envoyeur n'est pas établie de façon univoque, car tout détenteur de la clé a la capacité de former la transformation HMAC. Bien que la transformation HMAC établisse une propriété de sécurité au niveau du groupe, l'authentification de l'origine des données n'est pas réalisée.

Certaines applications de groupe exigent l'authentification de l'origine des données, où un membre du groupe ne peut pas réussir à se faire passer pour un autre membre du groupe. L'utilisation d'algorithmes de signatures numériques asymétriques, comme RSA, peut fournir une vraie authentification de l'origine des données.

Avec des algorithmes asymétriques, l'envoyeur génère une paire de clés, dont une n'est jamais partagée (appelée la "clé privée") et l'autre est distribuée aux autres membres du groupe (appelée la "clé publique").

Quand la clé privée est utilisée pour signer le résultat d'un algorithme de hachage cryptographique, le résultat est appelé une "signature numérique". Un receveur de la signature numérique utilise la clé publique, la valeur de la signature, et un hachage calculé indépendamment pour déterminer si l'origine prétendue du paquet est correcte ou non.

Le présent mémoire décrit comment les signatures numériques RSA peuvent être appliquées dans un mécanisme d'authentification ESP et AH pour assurer l'authentification de l'origine des données.

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].

2. Algorithme et mode

L'algorithme de clé publique RSA [RFC3447] est un algorithme de clé publique largement déployé couramment utilisé pour les signature numériques. Comparé aux autres algorithmes de clé publique, la vérification de signature est relativement efficace. Cette propriété est utile pour les groupes où les receveurs peuvent avoir des capacités de traitement limitées. L'algorithme RSA est couramment pris en charge par les matériels.

Deux méthodes de codage de signature numérique sont prises en charge dans la [RFC3447]. RSASSA-PKCS1-v1_5 DOIT être pris en charge par une mise en œuvre conforme. RSASSA-PSS est généralement estimé être plus sûr, mais n'est pas largement déployé au moment de la rédaction du présent mémoire. RSASSA-PSS DEVRAIT être utilisé chaque fois qu'il est disponible. SHA-1 [SHA] DOIT être utilisé comme algorithme de hachage de signature utilisé par l'algorithme RSA de signature numérique.

Quand elle est spécifiée pour ESP, la valeur de vérification d'intégrité (ICV, *Integrity Check Value*) est de taille égale au module RSA, sauf si le module RSA n'est pas un multiple de 8 bits. Dans ce cas, la ICV DOIT être précédée de entre 1 et 7 bits réglés à zéro de façon que la ICV soit un multiple de 8 bits. Cette spécification correspond au résultat S [RFC3447], paragraphe 8.1.1 (RSASSA-PSS) et [RFC3447], paragraphe 8.2.1 (RSASSA-PKCS1-v1_5) quand le module RSA n'est pas un multiple de 8 bits. Aucun bit implicite de bourrage d'ICV ESP n'est nécessaire.

Quand elle est spécifiée pour AH, l'ICV est de taille égale au module RSA, sauf si le module RSA n'est pas un multiple de 32 bits (IPv4) ou 64 bits (IPv6) [RFC4302], paragraphe 2.6. Dans ce cas, des bits explicites de bourrage de l'ICV sont nécessaires pour créer une ICV de taille convenable [RFC4302], paragraphe 3.3.3.2.1.

Le mécanisme de distribution de la clé publique RSA et son intervalle de remplacement sont une affaire de politique de groupe. L'utilisation d'une paire de clés éphémères avec une durée de vie de l'association de sécurité ESP ou AH est RECOMMANDÉE. Cette politique recommandée réduit l'exposition de la clé privée RSA à la durée de vie des données signées par la clé privée. Aussi, cela supprime le besoin de révoquer ou transmettre la période de validité de la paire de clés.

La génération de signature numérique est effectuée comme décrit au paragraphe 8.1.1 de la [RFC3447] (RSASSA-PSS) et au paragraphe 8.2.1 de la [RFC3447] (RSASSA-PKCS1-v1_5). La portion authentifiée du paquet AH ou ESP (paragraphe 3.3.3 de la [RFC4302], paragraphe 3.3.2 de la [RFC4303]) est utilisée comme le message M, qui est passé à la fonction de génération de signature. La clé privée RSA du signataire est passée comme K. Pour résumer, le processus de génération de

signature calcule un hachage SHA-1 des octets du paquet authentifié, signe le hachage SHA-1 en utilisant la clé privée, et code le résultat avec le type de codage RSA spécifié. Ce processus résulte en une valeur S, qui est appelée la ICV dans AH et ESP.

La vérification de la signature numérique est effectuée comme décrit au paragraphe 8.1.2 de la [RFC3447] (RSASSA-PSS) et au paragraphe 8.2.2 de la [RFC3447] (RSASSA-PKCS1-v1_5). À réception, l'ICV est passée à la fonction de vérification comme S. La portion authentifiée du paquet AH ou ESP est utilisée comme le message M, et la clé publique RSA est passée comme (n, e). En résumé, la fonction de vérification calcule un hachage SHA-1 des octets du paquet authentifié, déchiffre le hachage SHA-1 dans l'ICV, et valide que le codage approprié a été appliqué et est correct. Les deux hachages SHA-1 sont comparés, et si ils sont identiques, la validation est réussie.

2.1 Discussion de la taille de clé

Le choix de la taille du module RSA doit être fait avec soin. Si elle est trop petite, un attaquant peut être capable de reconstruire la clé privée utilisée pour signer les paquets avant qu'elle ne soit plus utilisée. Ce type d'événements peut résulter en la compromission de la propriété d'authentification de l'origine des données. Cependant, choisir une taille de module plus grande que nécessaire va résulter en un coût inutilement élevé de cycles de CPU pour l'envoyeur et tous les receveurs du paquet.

Une mise en œuvre conforme DOIT prendre en charge une taille de module de 1024 bits.

Des lignes directrices récentes [TWIRL], [RSA-TR] sur les tailles de clé font des estimations sur la somme d'efforts que doit faire un attaquant pour reconstruire une clé privée RSA. Le Tableau 1 résume le temps maximum pendant lequel le module choisi devrait être utilisé selon sa taille. Noter que ces recommandations se fondent sur des facteurs comme le coût de traitement et la mémoire, ainsi que les méthodes d'analyse cryptographique qui sont en cours au moment de la publication de ces documents. Lorsque ces facteurs changent, les choix de durée de vie de clé devraient les prendre en compte.

Nombre de bits du module	Durée de vie maximum recommandée
768	1 semaine
1024	1 an

Tableau 1. Recommandations de durée d'utilisation de clé RSA

3. Performance

L'algorithme de clé asymétrique RSA est très coûteux en termes de temps de traitement comparé aux algorithmes HMAC. Cependant, le coût de traitement est décroissant dans le temps. Des processeurs généraux plus rapides sont déployés, des mises en œuvre de logiciel plus rapides sont développées, et la prise en charge de l'accélération du matériel pour l'algorithme devient prévalente.

On devrait veiller à ce que les signatures RSA ne soient pas utilisées pour des applications où les receveurs potentiels sont connus pour n'avoir pas la puissance de calcul suffisante pour vérifier la signature. Il est aussi important d'utiliser ce schéma judicieusement quand un receveur peut être alimenté par une pile.

L'algorithme de clé asymétrique RSA est le plus convenable pour protéger du trafic réseau pour lequel :
- o L'envoyeur a une quantité substantielle de puissance de traitement, et
- o le trafic du réseau est assez faible pour que l'ajout d'une étiquette d'authentification relativement grande (de l'ordre de 62 à 256 octets) ne cause pas la fragmentation des paquets.

La génération et la signature de la paire de clés RSA sont des opérations substantiellement plus coûteuses que la vérification de signature, mais elles sont limitées à l'envoyeur.

La taille du module RSA affecte le traitement requis pour créer et vérifier les signature numériques RSA. Un grand soin devrait être apporté à la détermination de la taille du module nécessaire pour l'application. Des tailles de module plus petites peuvent être choisies tant que le trafic réseau protégé par la clé privée s'écoule pendant moins longtemps que ce qu'il est estimé nécessaire pour qu'un attaquant découvre la clé privée. Cette durée de vie est considérablement plus courte que la plupart des applications de clé publique qui mémorisent les données signées pour une certaine période. Mais comme la

signature numérique est utilisée pour les seuls besoins de vérification de l'envoyeur, un module qui est considéré comme faible dans un autre contexte peut être satisfaisant.

La taille de l'exposant public RSA peut affecter le traitement requis pour vérifier les signatures numériques RSA. Des signatures RSA à faible exposant peuvent résulter en un moindre coût de traitement de la vérification. Au moment de la rédaction, aucune attaque n'est connue contre des signatures RSA à faible exposant qui permettraient à un attaquant de créer une signature valide en utilisant le schéma RSAES-OAEP.

L'ajout d'une signature numérique comme étiquette d'authentification ajoute un nombre significatif d'octets au paquet. Cela augmente la probabilité que le paquet encapsulé dans ESP ou AH puisse être fragmenté.

4. Interaction avec le mécanisme de chiffrement ESP

L'algorithme de signature RSA ne peut pas être utilisé avec un algorithme de mode combiné ESP qui comporte une ICV explicite. L'algorithme de mode combiné va ajouter le champ d'ICV ESP, qui ne permet pas l'utilisation d'un algorithme d'authentification séparé pour ajouter le champ ICV ESP. Un exemple d'un tel algorithme est l'algorithme ESP Galois/mode compteur [RFC4106].

5. Considérations de gestion de clé

Les mécanismes de gestion de clé qui négocient l'utilisation de signatures RSA DOIVENT inclure la longueur du module RSA durant la négociation de politique en utilisant l'attribut SA de longueur de clé d'authentification. Cela donne aux appareils l'opportunité de refuser l'utilisation de l'algorithme. Ceci est particulièrement important pour les appareils qui ont des contraintes de processeurs qui pourraient n'être pas capables de vérifier les signatures qui utilisent de plus grandes tailles de clé.

Les mécanismes de gestion de clé qui négocient l'utilisation de signatures RSA DOIVENT aussi inclure la méthode de codage durant la négociation de politique en utilisant l'attribut SA d'algorithme de codage de signature.

Un receveur doit avoir la clé publique RSA afin de vérifier l'intégrité du paquet. Quand elle est utilisée avec un système de gestion de clé de groupe (par exemple, de la [RFC3547]) la clé publique DEVRAIT être envoyée au titre de la politique de téléchargement de clé. Si le groupe a plusieurs envoyeurs, la clé publique de chaque envoyeur DEVRAIT être envoyée au titre de la politique de téléchargement de clé.

L'utilisation de cette transformation pour obtenir l'authentification de l'origine des données pour des SA au bit près N'EST PAS RECOMMANDÉE. Dans le cas de SA au bit près (comme celles négociées par l'échange de clé Internet [RFC4306]) l'authentification de l'origine des données peut être réalisée avec une transformation HMAC. Parce que l'impact sur les performances d'une signature RSA est normalement plus grand que celui d'un HMAC, la valeur de l'utilisation de cette transformation pour une connexion au bit près est limitée.

6. Considérations sur la sécurité

Le présent document fournit une méthode d'authentification pour les signature numériques qui utilisent ESP et AH. Cette caractéristique fournit les protections suivantes :

- o Intégrité contre la modification du message. La signature numérique permet au receveur du message de vérifier qu'il a exactement la même que celle que l'envoyeur a signé.
- o Authentification de l'hôte. La nature asymétrique de l'algorithme de clé publique RSA permet à l'envoyeur d'être vérifié de façon univoque, même quand le message est envoyé à un groupe.

La non répudiation n'est pas revendiquée comme propriété de cette transformation. Parfois, la propriété de non répudiation peut être appliquée à des signatures numériques sur des objets de niveau application (par exemple, des messages électroniques). Cependant, le présent document décrit un moyen pour authentifier des objets de niveau réseau (c'est-à-dire, des paquets IP) qui sont éphémères et non directement corrélés à une application. La non répudiation n'est pas applicable aux objets de niveau réseau (c'est-à-dire, les paquets IP).

Un certain nombre d'attaques sont suggérées par la [RFC3552]. Les paragraphes suivants décrivent les risques que présentent ces attaques quand les signatures RSA sont utilisées pour l'authentification de paquet ESP et AH.

Il a été prévu que SHA-1 serait dépassé en 2010, à cause des avancées de la technologie par laquelle un adversaire peut doubler sa puissance de calcul en gros tous les dix-huit mois. De récentes attaques sur SHA-1 soulignent l'importance de remplacer SHA-1, mais ne motivent pas de le remplacer avant cette date [SHA-COMMENTS]. L'utilisation de cette transformation après cette date DEVRAIT être précédée d'une analyse de la poursuite de son adéquation.

6.1 Espionnage

Le présent document ne traite pas de la confidentialité. Cette fonction, si elle est désirée, doit être traitée par un chiffrement ESP qui est utilisé avec la méthode d'authentification des signatures RSA La signature RSA elle-même n'a pas besoin d'être protégée contre un espion.

6.2 Répétition

Le présent document ne traite pas des attaques en répétition. Cette fonction, si elle est désirée, est traitée par l'utilisation de numéros de séquence ESP et AH comme défini dans les [RFC4303] et [RFC4302].

6.3. Insertion de message

Le présent document traite directement les attaques d'insertion de message. Les messages insérés vont échouer à l'authentification et vont être éliminés par le receveur.

6.4 Suppression

Le présent document ne traite pas des attaques de suppression. Il est concerné seulement par la validation de la légitimité des messages qui ne sont pas supprimés.

6.5 Modification

Le présent document traite directement les attaques de modification de message. Les messages modifiés vont échouer à l'authentification et être éliminés par le receveur.

6.6 Interposition

Pour autant que la clé publique RSA de l'envoyeur soit donnée de façon fiable à un receveur (par exemple, par un protocole de gestion de clé) il va être capable de vérifier que la signature numérique est correcte. Un interposé ne sera pas capable de tromper l'envoyeur réel sauf si il acquiert la clé privée RSA par d'autres moyens.

La taille du module RSA doit être choisie avec soin pour s'assurer que le temps dont un interposé a besoin pour déterminer la clé privée RSA par cryptanalyse est plus long que le temps pendant lequel les paquets sont signés avec cette clé privée.

6.7 Déni de service

Selon les règles de traitement de IPsec, un receveur d'un paquet ESP et AH commence par chercher l'association de sécurité dans la base de données des SA. Si il la trouve, le numéro de séquence ESP ou AH du paquet est vérifié. Aucun autre traitement n'est appliqué aux paquets dont le numéro de séquence est invalide.

Un attaquant qui envoie un paquet ESP ou AH correspondant à une SA valide sur le système et qui a aussi un numéro de séquence valide, va causer l'exécution de l'étape d'authentification ESP ou AH par le receveur. Comme le processus de vérification d'une signature numérique RSA consomme des quantités relativement grandes de traitement, un grand nombre de ces paquets pourrait conduire à une attaque de déni de service sur le receveur.

Si le message a été envoyé à un groupe de diffusion groupée IPv4 ou IPv6, tous les membres du groupe qui ont reçu le paquet vont être simultanément soumis à l'attaque.

Cette attaque peut être atténuée contre la plupart des attaquants en encapsulant ESP ou AH et en utilisant une signature RSA pour l'authentification au sein de ESP ou AH avec une transformation HMAC pour l'authentification. Dans ce cas, la transformation HMAC va être validée en premier, et tant que l'attaquant ne possède pas la clé HMAC, aucune signature numérique ne va être évaluée sur les paquets de l'attaquant. Cependant, si l'attaquant possède la clé HMAC (par exemple, l'attaquant est un membre légitime du groupe qui utilise la SA) alors l'attaque de déni de service ne peut pas être contrée.

7. Considérations relatives à l'IANA

L'allocation d'un numéro est nécessaire dans l'espace de noms "Algorithme d'authentification IPsec" dans le registre du protocole de gestion de clés et d'associations de sécurité Internet (ISAKMP, *Internet Security Association and Key Management Protocol*) [ISAKMP-REG]. Le mnémonique devrait être "SIG-RSA".

L'allocation d'un numéro est aussi nécessaire dans l'espace de noms "Identifiant de transformation AH IPsec" dans le registre ISAKMP. Son mnémonique devrait être "AH_RSA".

Un nouvel "attribut d'association de sécurité IPsec" est nécessaire dans le registre ISAKMP pour passer la taille de module RSA. La classe d'attribut devrait être appelée "Longueur de clé d'authentification", et elle devrait être de type variable.

Un second "attribut d'association de sécurité IPsec" est nécessaire dans le registre ISAKMP pour passer le type de codage de signature RSA. La classe d'attribut devrait être appelée "Algorithme de codage de signature", et elle devrait être un type de base. Les règles suivantes s'appliquent pour définir les valeurs de l'attribut :

Nom	**Valeur**
Réservé	0
RSASSA-PKCS1-v1_5	1
RSASSA-PSS	2

Les valeurs 3 à 61439 sont réservées à l'IANA. De nouvelles valeurs DOIVENT être ajoutées selon une action de normalisation comme défini dans la [RFC2434]. Les valeurs 61440 à 65535 sont pour utilisation privée et peuvent être allouées par les mises en œuvre pour leurs propres besoins.

8. Remerciements

Scott Fluhrer et David McGrew ont fourni des conseils sur les tailles de clé applicables. Scott Fluhrer a aussi donné des conseils en ce qui concerne les durées de vie des clés. Ian Jackson, Steve Kent, et Ran Canetti ont fourni de nombreux commentaires utiles. Sam Hartman, Russ Housley, et Lakshminth Dondeti ont fourni de précieux conseils durant le développement de ce document.

9. Références

9.1 Références normatives

[ISAKMP-REG] http://www.iana.org/assignments/isakmp-registry

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

[RFC3447] J. Jonsson et B. Kaliski, "Normes de cryptographie à clés publiques (PKCS) n° 1 : Spécifications de la cryptographie RSA version 2.1", février 2003. *(Obsolète, remplacée par* RFC8017) *(Information)*

[RFC3552] E. Rescorla, B. Korver, "Lignes directrices pour la rédaction d'une section de considérations sur la sécurité dans les RFC", juillet 2003. (BCP0072)

[RFC4302] S. Kent, "En-tête d'authentification IP", décembre 2005. *(P.S.)*

[RFC4303] S. Kent, "Encapsulation de charge utile de sécurité dans IP (ESP)", décembre 2005. (*Remplace* RFC2406) *(P.S.)*

[SHA] FIPS PUB 180-2. "Specifications for the Secure Hash Standard", août 2002. http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf.

9.2 Références pour information

[RFC2404] C. Madson, R. Glenn, "Utilisation de HMAC-SHA-1-96 au sein d'ESP et d'AH", novembre 1998. *(P.S.)*

[RFC2434] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, octobre 1998. *(Rendue obsolète par la* RFC5226)

[RFC3547] M. Baugher, B. Weis, T. Hardjono et H. Harney, "Le domaine d'interprétation de groupe", juillet 2003. *(Obsolète, voir la* RFC6407)

[RFC4106] J. Viega, D. McGrew, "Utilisation du mode Galois/Compteur (GCM) dans une encapsulation IPsec de charge utile de sécurité (ESP)", juin 2005. *(P.S.)*

[RFC4306] C. Kaufman, "Protocole d'échange de clés sur Internet (IKEv2)", décembre 2005. *(Obsolète, voir la* RFC5996)

[RSA-TR] B. Kaliski, "TWIRL and RSA Key Size", RSA Laboratories Technical Note, http://www.rsasecurity.com/rsalabs/node.asp?id=2004, 6 mai 2003.

[SHA-COMMENTS] NIST, "Brief Comments on Recent Cryptanalytic Attacks on Secure Hashing Functions and the Continued Security Provided by SHA-1", août 2004. http://csrc.nist.gov/hash_standards_comments.pdf .

[TWIRL] A. Shamir et E. Tromer, "Factoring Large Numbers with the TwIRL Device", Travail en cours, 2003.

Adresse de l'auteur

Brian Weis
Cisco Systems
170 W. Tasman Drive,
San Jose, CA 95134-1706
USA

téléphone : (408) 526-4796
mél : bew@cisco.com

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.