Groupe de travail Réseau
**Request for Comments : 2747**
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

F. Baker, Cisco
B. Lindell, USC/ISI
M. Talwar, Microsoft
janvier 2000

# Authentification cryptographique RSVP



**Résumé**
Le présent document décrit le format et l'utilisation de l'objet INTEGRITY de RSVP pour fournir l'intégrité bond par bond et l'authentification des messages RSVP.

## 1. Introduction

Le protocole de réservation de ressource (RSVP, *Resource ReSerVation Protocol*) [1] est un protocole d'établissement d'état réparti dans les routeurs et les hôtes, et en particulier pour réserver des ressources afin de mettre en œuvre l'intégration de services. RSVP permet à des usagers particuliers d'obtenir un accès préférentiel aux ressources du réseau, sous le contrôle d'un mécanisme de contrôle d'admission. La permission de faire une réservation va dépendre à la fois de la disponibilité de la ressource demandée le long du chemin des données, et de la satisfaction à des règles de politique.

Pour assurer l'intégrité de ce mécanisme de contrôle d'admission, RSVP requiert la capacité de protéger ses messages contre la corruption et l'espionnage. Le présent document définit un mécanisme de protection de l'intégrité du message RSVP bond par bond. Le schéma proposé transmet un résumé authentifiant du message, calculé en utilisant une clé d'authentification secrète et un algorithme de hachage chiffré. Ce schéma assure la protection contre la falsification et la modification de message. L'objet INTEGRITY de chaque message RSVP est étiqueté avec un numéro de séquence à usage unique. Cela permet au receveur du message d'identifier les répétitions et donc de déjouer les attaques en répétition. Le mécanisme proposé ne prend pas en charge la confidentialité, car les messages restent en clair ; cependant le mécanisme est aussi exportable dans la plupart des pays, ce qui serait impossible si on devait utiliser un algorithme de confidentialité. Noter que ce document utilise les termes "envoyeur" et "receveur" différemment de [1]. Ils sont utilisés ici pour se référer aux systèmes qui se font face à travers un bond RSVP, "l'envoyeur" étant le système qui génère les messages RSVP.

L'algorithme de prévention de répétition de message est assez simple. L'envoyeur génère des paquets avec un accroissement monotone des numéros de séquence. De son côté, le receveur n'accepte les paquets que si ils ont un numéro de séquence supérieur à celui du paquet précédent. Pour commencer ce processus, un receveur prend contact avec l'envoyeur pour obtenir un numéro de séquence initial. Le présent mémoire expose les moyens de relâcher l'ordre strict de livraison des messages ainsi que les techniques pour générer des numéros de séquence à accroissement monotone qui soient robustes à travers des défaillances et redémarrages de l'envoyeur.

Le mécanisme proposé est indépendant d'un algorithme cryptographique spécifique, mais le document décrit l'utilisation du hachage chiffré pour l'authentification de message qui se sert de HMAC-MD5 [7]. Comme il est noté dans [7], il existe des hachages plus forts, tels que HMAC-SHA1, que les mises en œuvre feraient bien de se procurer. Cependant, dans le cas général, [7] suggère que HMAC-MD5 est adéquat pour les besoins courants et a des caractéristiques de performances préférables. [7] offre aussi un code source et des vecteurs d'essai pour cet algorithme, un avantage pour ceux qui veulent vérifier l'interopérabilité. HMAC-MD5 est exigé comme le minimum qui doit être universellement inclus dans les mises en œuvre de RSVP qui fournissent l'authentification cryptographique, avec d'autres propositions facultatives (voir la Section 6 sur les exigences de conformité).

La somme de contrôle RSVP PEUT être désactivée (mise à zéro) lorsque l'objet INTEGRITY est inclus dans le message,

car le résumé de message est une vérification d'intégrité beaucoup plus forte.

### 1.1 Conventions utilisée dans le document

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" dans ces document sont à interpréter comme décrit dans [8].

### 1.2 Pourquoi ne pas utiliser l'en-tête d'authentification IPsec standard ?

La question se pose évidemment de savoir pourquoi on devrait choisir de ne pas utiliser le mécanisme standard d'authentification IPsec [3,5] qui existe déjà. Ce point a été discuté en long et en large dans le groupe de travail, et l'utilisation d'IPsec a été rejetée pour les raisons suivantes.

Les associations de sécurité dans IPsec sont fondées sur une adresse de destination. Il n'est pas évident que les messages RSVP soient bien définis pour des associations de sécurité fondées sur la source ou la destination, car un routeur doit transmettre les messages PATH et PATH TEAR en utilisant la même adresse de source que l'envoyeur figurant dans le SENDER TEMPLATE. Le trafic RSVP peut autrement ne pas suivre exactement le même chemin que le trafic des données. Utiliser des associations fondées sur la source ou la destination exigerait d'ouvrir une nouvelle association de sécurité parmi les routeurs que traverse une réservation.

De plus, il a été noté que les relations de voisinage entre les systèmes RSVP ne sont pas limitées à ceux qui sont face à face à travers un canal de communication. Les relations RSVP à travers des nuages non RSVP, telles que celles décrites au paragraphe 2.9 de [1], ne sont pas nécessairement visibles du système envoyeur. Ces arguments suggèrent l'usage d'une stratégie de gestion de clés fondée sur les associations de routeur RSVP à routeur RSVP plutôt qu'IPsec.

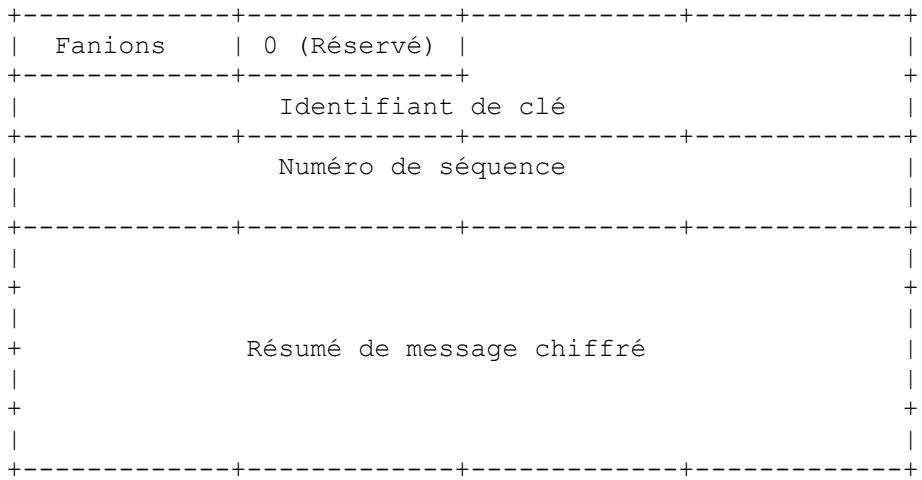
## 2. Structures de données

### 2.1 Format de l'objet INTEGRITY

Un message RSVP consiste en une séquence "d'objets," qui sont des champs codés de type-longueur-valeur avec des objectifs spécifiques. L'information nécessaire pour la vérification d'intégrité bond par bond est portée dans un objet INTEGRITY. Le même type d'objet INTEGRITY est utilisé pour IPv4 et IPv6.

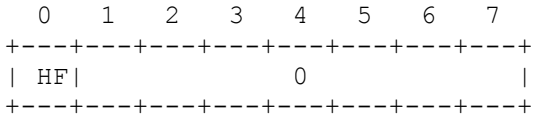
L'objet INTEGRITY a le format suivant :

Objet INTEGRITY de résumé de message chiffré : Classe = 4, C-Type = 1




o Fanions : Un champ de 8 bits avec le format suivant :

Fanions




Actuellement un seul fanion (HF) est défini. Les fanions restants sont réservés pour une utilisation future et DOIVENT être mis à 0.

- Bit 0 : Le fanion de prise de contact (HF, *Handshake Flag*) concerne le mécanisme de prise de contact d'intégrité (paragraphe 4.3). Les envoyeurs de message qui veulent répondre aux message de prise de contact d'intégrité DEVRAIENT mettre ce fanion à 1 tandis que ceux qui vont rejeter les messages de prise de contact d'intégrité DEVRAIENT le mettre à 0.

- Identifiant de clé : Un nombre de 48 bits non signé qui DOIT être unique pour un envoyeur donné. Les identifiants de clé localement uniques peuvent être générés en utilisant une combinaison d'adresse (IP ou MAC ou LIH) de l'interface et du numéro de clé. La combinaison de l'identifiant de clé et de l'adresse IP du système envoyeur identifie de façon univoque l'association de sécurité (paragraphe 2.2).

- Numéro de séquence : Un numéro de séquence unique non signé de 64 bits à accroissement monotone.
  Les valeurs de numéro de séquence peuvent être toute séquence à accroissement monotone qui fournit à l'objet INTEGRITY [de chaque message RSVP] une étiquette qui est unique pour la durée de vie de la clé associée. Les détails sur la génération du numéro de séquence sont présentés à la Section 3.

- Résumé de message chiffré : Le résumé DOIT être un multiple de 4 octets. Pour HMAC-MD5, il sera long de 16 octets.

### 2.2 Association de sécurité

Les systèmes envoyeur et receveur entretiennent une association de sécurité pour chaque clé d'authentification qu'ils partagent. Cette association de sécurité inclut les paramètres suivants :
- Algorithme d'authentification et mode d'algorithme utilisés.
- Clé utilisée avec l'algorithme d'authentification.
- Durée de vie de la clé.
- Interface d'envoi associée et autres critères de choix d'association de sécurité [EXIGÉ au système envoyeur].
- Adresse de source du système envoyeur [EXIGÉ au système receveur].
- Dernier numéro de séquence d'envoyeur utilisé avec cet identifiant de clé [EXIGÉ au système envoyeur].
- Liste des N derniers numéros de séquence reçus avec cet identifiant de clé [EXIGÉ au système receveur].

## 3. Génération des numéros de séquence

Dans cette section, on décrit les méthodes qui pourraient être choisies pour générer les numéros de séquence utilisés dans l'objet INTEGRITY d'un message RSVP. Comme indiqué précédemment, il y a deux propriétés importantes qui DOIVENT être satisfaites par la procédure de génération. La première propriété est que les numéros de séquence sont uniques, ou à utilisation unique, pour la durée de vie de la clé d'intégrité qui est utilisée actuellement. Un receveur peut utiliser cette propriété pour distinguer sans ambiguïté entre un message nouveau et un message répété. La seconde propriété est que les numéros de séquence sont générés en ordre d'accroissement monotone, modulo 2^64. Ceci est exigé pour réduire sensiblement la quantité d'états sauvegardés, car un receveur a seulement besoin de sauvegarder la valeur du plus fort numéro de séquence vu pour éviter une attaque en répétition. Comme le numéro de la séquence de début pourrait être arbitrairement grand, l'opération modulo est nécessaire pour arranger le retour à zéro des numéros de séquence sur la durée de vie de certaines clés. Cette solution est tirée de l'approche retenue pour TCP [9].

Le champ de numéro de séquence est choisi comme étant une quantité de 64 bits non signée. C'est assez grand pour éviter d'être consommé sur la durée de vie de la clé. Par exemple, si la durée de vie d'une clé était prudemment définie pour un an, il y aurait assez de valeurs de numéros de séquence pour envoyer des messages RSVP à un rythme moyen d'environ 585 giga messages par seconde. Un numéro de séquence de 32 bits limiterait ce taux moyen à environ 136 messages par seconde

La capacité à générer des numéros de séquence uniques à accroissement monotone à travers défaillances et redémarrages implique une forme de mémorisation stable, soit locale au niveau de l'appareil, soit à distance sur le réseau. Trois procédures de génération de numéros de séquence sont décrites ci-après.

### 3.1 Numéros de séquence simples

L'approche la plus directe est de générer un numéro de séquence unique en utilisant un compteur de messages. Chaque fois qu'un message est transmis pour une clé donnée, le compteur de numéros de séquence est incrémenté. La valeur courante du compteur est sauvegardée en continu ou périodiquement sur une mémorisation stable. Après un redémarrage, le compteur est récupéré en utilisant cette mémorisation stable. Si le compteur était sauvegardé de façon périodique sur une mémoire stable, le compte serait récupéré en augmentant la valeur sauvegardée de façon à être supérieure à toute valeur possible du compteur au moment de la défaillance. Cela peut être calculé en connaissant l'intervalle auquel le compteur était sauvegardé sur la mémorisation stable et en incrémentant la valeur mémorisée de cette quantité.

### 3.2 Numéros de séquence fondés sur une horloge en temps réel

La plupart des appareils n'auront probablement pas la capacité de sauvegarder les compteurs de numéros de séquence sur des mémoires stables pour chaque clé. Une solution universelle est de fonder les numéros de séquence sur la mémorisation stable d'une horloge en temps réel. De nombreux appareils de calcul ont un module d'horloge en temps réel qui comporte une mémorisation stable de l'horloge. Ces modules comportent généralement une forme quelconque de mémoire non volatile pour conserver les informations d'horloge en cas d'une défaillance d'alimentation.

Dans cette approche, on pourrait utiliser une valeur d'horodatage fondée sur NTP comme numéro de séquence. La période de retournement d'un horodatage NTP est d'environ 136 ans, soit beaucoup plus que la durée de vie raisonnable d'une clé. De plus, la granularité de l'horodatage NTP est assez fine pour permettre la génération d'un message RSVP chaque 200 picosecondes pour une clé donnée. De nombreux modules d'horloge en temps réel n'ont pas la résolution d'un horodatage NTP. Dans ces cas, les bits de moindre poids de l'horodatage peuvent être générés en utilisant un compteur de message qui est remis à zéro à chaque tic d'horloge. Par exemple, lorsque l'horloge en temps réel donne une résolution de 1 seconde, les 32 bits de moindre poids du numéro de séquence peuvent être générés en utilisant un compteur de messages. Les 32 bits restants sont remplis avec les 32 bits de moindre poids de l'horodatage. En supposant que le temps de récupération après défaillance prend plus longtemps qu'un tic de l'horloge en temps réel, le compteur de messages pour les bits de moindre poids peut être remis à zéro en toute sécurité après un redémarrage.

### 3.3 Numéros de séquence fondés sur une horloge du réseau

Si l'appareil ne contient aucune mémorisation stable des compteurs de numéro de séquence ou d'une horloge en temps réel, il pourrait récupérer l'horloge en temps réel du réseau en utilisant NTP. Une fois que l'horloge a été récupérée à la suite d'un redémarrage, la procédure de génération de numéro de séquence sera identique à celle décrite ci-dessus.

## 4. Traitement de message

Les mises en œuvre DEVRAIENT permettre la spécification d'interfaces à sécuriser soit pour l'envoi des messages, soit pour leur récéption, ou les deux. L'envoyeur doit s'assurer que tous les messages RSVP envoyés sur les interfaces d'envoi sécurisées comportent un objet INTEGRITY, généré en utilisant la clé appropriée. Les receveurs vérifient si les messages RSVP, excepté du type "Mise en cause d'intégrité" (paragraphe 4.3) qui arrivent sur une interface de réception sécurisée contiennent l'objet INTEGRITY. Si l'objet INTEGRITY est absent, le receveur élimine le message.

Les associations de sécurité sont à sens unique – les clés qu'un système envoyeur utilise pour signer ses messages peuvent être différentes des clés que son receveur utilise pour les signer. Et donc, chaque association est associée à un unique système envoyeur et (éventuellement) plusieurs systèmes receveurs.

Chaque envoyeur DEVRAIT avoir des associations de sécurité (et des clés) distinctes par interface (ou LIH *(Logical Interface Handle,* traitement d'interface logique) d'envoi sécurisée. Bien que les administrateurs puissent configurer tous les routeurs et hôtes d'un sous-réseau (ou pour cette affaire, de leur réseau) en utilisant une seule association de sécurité, les mises en œuvre DOIVENT supposer que chaque envoyeur peut envoyer en utilisant une association de sécurité distincte

sur chaque interface sécurisée. Chez l'envoyeur, le choix de l'association de sécurité se fonde sur l'interface à travers laquelle le message est envoyé. Ce choix PEUT inclure des critères supplémentaires, comme l'adresse de destination (lors d'un envoi de message en envoi individuel, sur un LAN de diffusion avec un grand nombre d'hôtes) ou les identités d'utilisateur chez l'envoyeur ou les receveurs [2]. Finalement, tous les receveurs de message prévus devraient participer à cette association de sécurité. Le flottement des chemins dans un nuage non RSVP peut être cause que des messages pour le même receveur soient envoyés sur des interfaces différentes à des moments différents. Dans de tels cas, les receveurs devraient participer à toutes les associations de sécurité possibles qui peuvent être choisies pour les interfaces à travers lesquelles le message peut être envoyé.

Les receveurs choisissent les clés sur la base de l'identifiant de clé et de l'adresse IP du système d'envoi. L'identifiant de clé est inclus dans l'objet INTEGRITY. L'adresse du système d'envoi peut être obtenue de l'objet RSVP_HOP, ou si il n'est pas présent (comme c'est le cas avec les messages PathErr et ResvConf) de l'adresse IP de source. Comme l'identifiant de clé est unique pour un envoyeur, cette méthode identifie la clé de façon univoque.

Le mécanisme d'intégrité modifie légèrement les règles de traitement des messages RSVP, à la fois quand on inclut l'objet INTEGRITY dans un message envoyé sur une interface d'envoi sécurisée et quand on accepte un message reçu sur une interface de réception sécurisée. Ces modifications sont détaillées ci-dessous.

### 4.1 Génération de message

Pour un message RSVP envoyé sur une interface d'envoi sécurisée, le message est créé comme décrit dans [1], avec les exceptions suivantes :

(1) Le champ Somme de contrôle RSVP est réglé à zéro. Si nécessaire, une somme de contrôle RSVP peut être calculée lorsque le traitement de l'objet INTEGRITY est achevé.
(2) L'objet INTEGRITY est inséré à l'endroit approprié, et sa localisation dans le message est mémorisée pour utilisation ultérieure.
(3) L'interface d'envoi et les autres critères appropriés (comme mentionné ci-dessus) sont utilisés pour déterminer la clé d'authentification et l'algorithme de hachage à utiliser.
(4) Les fanions inutilisés et le champ réservé dans l'objet INTEGRITY DOIVENT être réglés à 0. Le fanion de prise de contact (HF) devrait être réglé conformément aux règles spécifiées au paragraphe 2.1.
(5) Le numéro de séquence d'envoi DOIT être mis à jour pour assurer un nombre unique, à croissance monotone. Il est alors placé dans le champ Numéro de séquence de l'objet INTEGRITY.
(6) Le champ Résumé de message chiffré est réglé à zéro.
(7) L'identifiant de clé est placé dans l'objet INTEGRITY.
(8) Un résumé d'authentification du message est calculé en utilisant la clé d'authentification conjointement avec l'algorithme de hachage de clé. Lorsque l'algorithme HMAC-MD5 est utilisé, le calcul du hachage est décrit dans [7].
(9) Le résumé est écrit dans le champ Résumé cryptographique de l'objet INTEGRITY.

### 4.2 Réception de message

Lorsque le message est reçu sur une interface de réception sécurisée, et n'est pas du type "Mise en cause d'intégrité", il est traité de la façon suivante :

(1) Le champ Somme de contrôle RSVP est sauvegardé et le champ est ensuite réglé à zéro.
(2) Le champ Résumé cryptographique de l'objet INTEGRITY est sauvegardé et le champ est ensuite réglé à zéro.
(3) Le champ Identifiant de clé et l'adresse du système d'envoi sont utilisés pour déterminer de façon univoque la clé d'authentification et l'algorithme de hachage à utiliser. Le traitement de ce paquet peut être retardé lorsque le système de gestion de clés (Section 12) est interrogé sur ces informations.
(4) Un nouveau résumé chiffré est calculé en utilisant l'algorithme indiqué et la clé d'authentification.
(5) Si le résumé calculé ne correspond pas au résumé reçu, le message est éliminé sans autre traitement.
(6) Si le message est du type "Réponse d'intégrité", vérifier que l'objet CHALLENGE correspond identiquement à la mise en cause d'origine. Si il correspond, sauvegarder le numéro de séquence dans l'objet INTEGRITY comme étant le plus grand numéro de séquence reçu à ce moment.

Autrement, pour tous les autres messages RSVP, le numéro de séquence est validé pour empêcher les attaques en répétition et les messages avec des numéros de séquence invalides sont ignorés par le receveur.

Lorsque un message est accepté, le numéro de séquence de ce message pourrait mettre à jour une valeur mémorisée correspondant au plus grand numéro de séquence reçu à ce moment. Chaque message suivant doit alors pour être accepté

avoir un numéro de séquence supérieur (modulo 2^64). Cette règle de traitement simple empêche les attaques en répétition de message, mais elle doit être modifiée pour tolérer la livraison de messages déclassés dans certaines limites. Par exemple, si plusieurs messages étaient envoyés dans une salve (dans un rafraîchissement périodique généré par un routeur, ou par suite d'une fonction d'élimination) ils pourraient être réordonnés et alors les numéros de séquence ne seraient plus reçus en ordre croissant.

Une mise en œuvre DEVRAIT permettre la configuration administrative qui établit la tolérance du receveur à l'égard de la livraison des message déclassés. Une approche simple permettrait aux administrateurs de spécifier la fenêtre de messages correspondant au pire cas de comportement de réarrangement. Par exemple, on pourrait spécifier que les paquets réordonnés dans une fenêtre de 32 messages seraient acceptés. Si aucun réarrangement ne peut intervenir, la fenêtre est réglée à un.

Le receveur doit mémoriser une liste de tous les numéros de séquence vus au sein de la fenêtre de réarrangement. Un numéro de séquence reçu est valide si (a) il est supérieur au numéro de séquence maximum reçu ou (b) il est un numéro de séquence passé qui se tient dans la fenêtre de réarrangement et n'est pas enregistré dans la liste. L'acceptation d'un numéro de séquence implique de l'ajouter à la liste et de retirer un numéro de l'extrémité inférieure de la liste. Les messages reçus avec des numéros de séquence qui se tiennent en dessous de l'extrémité inférieure de la liste ou sont marqués vus dans la liste sont éliminés.

Quand un message "Mise en cause d'intégrité" est reçu sur une interface d'envoi sécurisée, il est traité de la façon suivante :

(1) Un message "Réponse d'intégrité" est formé en utilisant l'objet CHALLENGE reçu dans le message de mise en cause.

(2) Le message est renvoyé au receveur, sur la base de l'adresse IP de source du message de mise en cause, en utilisant les étapes de "génération de message" données plus haut. Le choix de la clé d'authentification et de l'algorithme de hachage à utiliser est déterminé par l'identifiant de clé fourni dans le message de mise en cause.

### 4.3 Prise de contact d'intégrité au redémarrage ou à l'initialisation du receveur

Pour obtenir le numéro de séquence de début pour une clé d'authentification active, le receveur PEUT initier une prise de contact d'intégrité avec l'envoyeur. Cette prise de contact consiste en une mise en cause par le receveur et la réponse de l'envoyeur, et peut être initiée durant un redémarrage ou retardée jusqu'à ce qu'arrive un message signé avec cette clé.

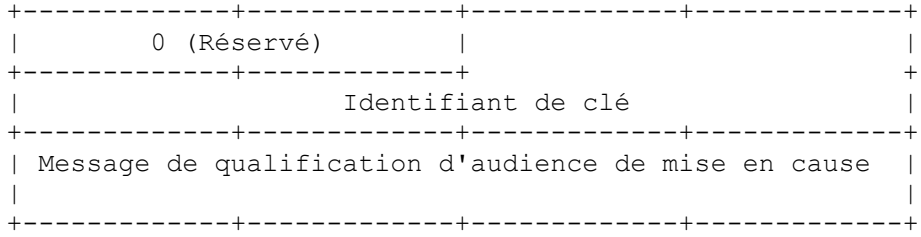
Une fois que le receveur a décidé d'initier une prise de contact d'intégrité pour une clé d'authentification particulière, il identifie l'envoyeur en utilisant l'adresse du système d'envoi configurée dans l'association de sécurité correspondante. Le receveur envoie alors un message Mise en cause d'intégrité RSVP à l'envoyeur. Ce message contient l'identifiant de clé pour identifier la clé de l'envoyeur et DOIT avoir un message de qualification d'audience *(cookie)* de mise en cause unique qui se fonde sur un secret local pour empêcher de le deviner (voir le paragraphe 2.5.3 de [4]). Il est suggéré que le message de qualification d'audience soit un hachage MD5 d'un secret local et un horodatage pour fournir l'unicité (voir la Section 9).

Un message de mise en cause d'intégrité RSVP va porter un type de message de 11. Le format du message est le suivant :

<Message de mise en cause d'intégrité> ::= <En-tête commun> <CHALLENGE>

L'objet CHALLENGE a le format suivant :

Objet CHALLENGE : Classe = 64, C-Type = 1




L'envoyeur accepte la "Mise en cause d'intégrité" sans faire de vérification d'intégrité. Il retourne un message RSVP

"Réponse d'intégrité" qui contient l'objet CHALLENGE d'origine. Il comporte aussi un objet INTEGRITY, signé avec la clé spécifiée par l'identifiant de clé inclus dans la "Mise en cause d'intégrité".

Un message RSVP Réponse d'intégrité va porter un type de message de 12. Le format du message est le suivant :

<Message de réponse d'intégrité> ::= <En-tête commun> <INTEGRITY> <CHALLENGE>

Le message "Réponse d'intégrité" n'est accepté par le receveur (celui qui met en cause) que si l'objet CHALLENGE retourné correspond à celui envoyé dans le message "Mise en cause d'intégrité". Cela empêche la répétition de vieux messages "Réponse d'intégrité". Si la correspondance est établie, le receveur sauvegarde le numéro de séquence de l'objet INTEGRITY comme dernier numéro de séquence reçu avec l'identifiant de clé inclus dans le CHALLENGE.

Si une réponse n'est pas reçue dans un délai donné, la mise en cause est répétée. Lorsque la prise de contact d'intégrité s'achève avec succès, le receveur commence à accepter les messages normaux de signalisation RSVP provenant de cet envoyeur et ignore tous les autres messages "Réponse d'intégrité".

Le fanion de prise de contact (HF) est utilisé pour donner aux mises en œuvre la souplesse de ne pas inclure le mécanisme de prise de contact d'intégrité. En établissant ce fanion à 1, les envoyeurs de messages qui mettent en œuvre la prise de contact d'intégrité se distinguent eux-mêmes de ceux qui ne le font pas. Les receveurs NE DEVRAIENT PAS tenter de prendre contact avec des envoyeurs dont l'objet INTEGRITY a HF = 0.

Une prise de contact d'intégrité peut n'être pas nécessaire dans tous les environnements. Une utilisation courante de l'intégrité RSVP sera entre des routeurs de domaines qui échangent du trafic, qui vont vraisemblablement traiter un flux continu de messages RSVP du fait des effets d'agrégation. Lorsque un routeur redémarre après une défaillance, des messages RSVP valides provenant d'envoyeurs de trafic vont probablement arriver dans un bref délai. Si on suppose que des messages répétés sont injectés dans le flux de messages RSVP valides, il peut n'y avoir qu'une petite fenêtre d'opportunité pour une attaque de répétition avant qu'un message valide soit traité. Ce message valide va établir le plus grand numéro de séquence vu avec une valeur supérieure à tout nombre déjà mémorisé avant la défaillance, empêchant d'autres répétitions.

D'un autre côté, ne pas utiliser la prise de contact d'intégrité pourrait permettre l'exposition à des attaques en répétition si il y a une longue période de silence de la part d'un envoyeur donné, suivie du redémarrage d'un receveur. Donc, qu'un receveur effectue ou non une prise de contact d'intégrité avec les envoyeurs qui acceptent de répondre aux messages de "Mise en cause d'intégrité" DEVRAIT être une décision administrative, ainsi que d'accepter des messages provenant d'envoyeurs qui refusent de faire ainsi. Ces décisions se fonderont sur des hypothèses se rapportant à un environnement de réseau particulier.

## 5. Gestion des clés

Il est vraisemblable que l'IETF va définir une norme de protocole de gestion de clés. Il est très souhaitable d'utiliser ce protocole de gestion de clés pour distribuer les clés d'authentification RSVP dans les mises en œuvre de RSVP communicantes. Un tel protocole fournirait l'échelonnabilité et réduirait de façon significative la charge administrative humaine. L'identifiant de clé peut être utilisé comme l'attache entre RSVP et un tel futur protocole. Les protocoles de gestion de clé ont une longue histoire de fautes subtiles qui ont souvent été découvertes longtemps après la première description publique du protocole. Pour éviter d'avoir à changer toutes les mises en œuvre de RSVP si une telle faute devait être découverte, les techniques intégrées de gestion de clé ont été délibérément omises de la présente spécification.

### 5.1 Procédures de gestion des clés

Chaque clé a une durée de vie associée qui est enregistrée dans tous les systèmes (envoyeur et receveurs) configurés avec cette clé. Le concept d'une "durée de vie de clé" exige simplement que les heures la plus tôt (KeyStartValid) et la plus tard (KeyEndValid) auxquelles la clé est valide soient programmables d'une façon que comprend le système. Certains mécanismes de génération de clé, tels que Kerberos ou certains schémas de clés publiques, peuvent produire directement des clés éphémères. Dans ce cas, la durée de vie de la clé est implicitement définie au titre de la clé.

En général, aucune clé n'est utilisée en dehors de sa durée de vie (mais voir au paragraphe 5.3). Des mécanismes possibles de gestion de la durée de vie des clés incluent le protocole de l'heure du réseau (NTP) et des horloges matérielles de l'heure du jour.

Pour maintenir la sécurité, il est conseillé de changer la clé d'authentification RSVP de façon régulière. Il devrait être possible de changer la clé d'authentification RSVP sans perte d'état RSVP ou refus de service de réservation, et sans exiger que des gens changent toutes les clés d'un coup. Cela exige qu'une mise en œuvre RSVP prenne en charge la mémorisation et l'utilisation de plus d'une clé d'authentification RSVP active en même temps. Donc, l'envoyeur et le receveur devraient tous deux avoir plusieurs clés actives pour une association de sécurité donnée.

Comme les clés sont partagées entre un envoyeur et (éventuellement) plusieurs receveurs, il y a une zone d'incertitude autour du moment où intervient le changement de clés durant laquelle certains systèmes peuvent encore utiliser la vieille clé et d'autres peuvent être déjà passés à la nouvelle clé. La taille de cette zone d'incertitude est en rapport avec la synchronisation d'horloge des systèmes. Les administrateurs devraient configurer le recoupement entre le moment d'expiration de la vieille clé (KeyEndValid) et la validité de la nouvelle clé (KeyStartValid) à au moins deux fois la taille de cet intervalle d'incertitude. Cela va permettre à l'envoyeur de faire le changement de clés au point médian de cet intervalle et d'être sûr que tous les receveurs vont maintenant accepter la nouvelle clé. Pendant la durée du recoupement des durées de vie des clés, un receveur doit être prêt à authentifier les messages en utilisant l'une ou l'autre clé.

Durant un changement de clé, il sera nécessaire que chaque receveur fasse une prise de contact avec l'envoyeur en utilisant la nouvelle clé. Comme indiqué plus haut, un receveur a le choix de prendre l'initiative d'une prise de contact durant le changement ou de la retarder jusqu'à réception d'un message utilisant cette clé.

### 5.2 Exigences de la gestion des clés

Les exigences pour une mise en œuvre sont les suivantes :

- Il est fortement souhaitable d'une éventuelle faille de la sécurité dans un protocole de l'Internet ne compromette pas automatiquement les autres protocoles de l'Internet. La clé d'authentification de la présente spécification NE DEVRAIT PAS être mémorisée en utilisant des protocoles ou algorithmes qui sont connus pour être fautifs.

- Une mise en œuvre DOIT prendre en charge la mémorisation et l'utilisation de plus d'une clé en même temps, à la fois pour les systèmes envoyeurs et receveurs.

- Une mise en œuvre DOIT associer une durée de vie spécifique (c'est-à-dire, KeyStartValid et KeyEndValid) à chaque clé et à l'identifiant de clé correspondant.

- Une mise en œuvre DOIT prendre en charge la distribution manuelle de clé (par exemple, l'utilisateur privilégié tape manuellement la clé, la durée de vie de la clé, et l'identifiant de clé sur la console). La durée de vie peut être infinie.

- Si plus d'un algorithme est pris en charge, la mise en œuvre DOIT exiger que l'algorithme soit spécifié pour chaque clé au moment où les autres informations de clé sont entrées.

- Les clés qui sont périmées PEUVENT être automatiquement supprimées par la mise en œuvre.

- La suppression manuelle des clés actives DOIT aussi être acceptée.

- La mémorisation des clés DEVRAIT persister à travers un redémarrage système, à chaud ou à froid, pour faciliter le fonctionnement.

### 5.3 Cas pathologique

Il est possible que la dernière clé pour une association de sécurité donnée soit arrivée à expiration. Lorsque cela arrive, il n'est pas acceptable de revenir à une condition de non authentification, et il n'est pas conseillé d'interrompre les réservations en cours. Donc, le système devrait envoyer une notification "expiration de la dernière clé d'authentification" au gestionnaire de réseau et traiter la clé comme ayant une durée de vie infinie jusqu'à ce que sa durée de vie soit étendue, que la clé soit supprimée par la gestion du réseau, ou qu'une nouvelle clé soit configurée.

## 6. Exigences de conformité

Pour se conformer à la présente spécification, une mise en œuvre DOIT prendre en charge tous ces aspects. L'algorithme d'authentification HMAC-MD5 défini dans [7] DOIT être mis en œuvre par toutes les applications qui revendiquent la conformité. Une mise en œuvre conforme PEUT aussi accepter d'autres algorithmes d'authentification tels que l'algorithme de hachage sécurisé (SHA) du NIST. La distribution manuelle de clés telle que décrite ci-dessus DOIT être acceptée par toutes les mises en œuvre conformes. Toutes les mises en œuvre DOIVENT accepter le changement de clé en douceur décrit à la section "Procédures de gestion de clés".

Les mises en œuvre DEVRAIENT accepter une norme de protocole de gestion de clé pour une distribution sécurisée des clés d'authentification RSVP telle qu'un protocole de gestion de clé normalisé par l'IETF.

## 7. Génération de clés d'authentification RSVP par Kerberos

Kerberos[10] PEUT servir pour générer la clé d'authentification RSVP utilisée pour générer une signature dans l'objet Integrity envoyé d'un envoyeur RSVP à un receveur. La génération de clés Kerberos évite l'utilisation de clés partagées entre les envoyeurs et receveurs RSVP tels que les hôtes et les routeurs. Kerberos permet l'utilisation de relations chiffrées de tiers de confiance entre principaux de sécurité (envoyeurs et receveurs RSVP) lorsque le centre de distribution de clés Kerberos (KDC) établit une clé de session éphémère qui sera ensuite partagée entre envoyeur et receveurs RSVP. Dans le cas de diffusion groupée, tous les receveurs d'un messages RSVP en diffusion groupée DOIVENT partager une seule clé avec le KDC (par exemple, les receveurs sont effectivement le même principal de sécurité par rapport à Kerberos).

Les informations de clés déterminées par l'envoyeur PEUVENT spécifier l'utilisation de Kerberos à la place des clés partagées configurées, comme mécanisme d'établissement de clé entre l'envoyeur et le receveur. L'identité Kerberos du receveur est établie au titre de la configuration d'interface de l'envoyeur ou elle peut être établie par d'autres mécanismes. Lors de la génération du premier message RSVP pour un identifiant de clé spécifique, l'envoyeur demande un ticket de service Kerberos et obtient en retour une clé de session éphémère et un ticket Kerberos de la part du KDC. L'envoyeur encapsule le ticket et l'identité de l'envoyeur dans un objet Identity Policy [2]. L'envoyeur inclut l'objet Policy dans le message RSVP. La clé de session est alors utilisée par l'envoyeur comme clé d'authentification RSVP à l'étape (3) du paragraphe 4.1, et est mémorisée comme informations de clé associées à l'identifiant de clé.

À réception d'un message RSVP, le receveur restitue le ticket Kerberos à partir de l'objet Identity Policy, déchiffre le ticket et restitue la clé de session à partir du ticket. La clé de session est la même clé qu'utilisée par l'envoyeur et utilisée comme clé à l'étape (3) du paragraphe 4.2. Le receveur mémorise la clé pour l'utiliser dans le traitement ultérieur des messages RSVP.

Les tickets Kerberos ont une durée de vie et l'envoyeur NE DOIT PAS utiliser les tickets qui sont arrivés à expiration. Un nouveau ticket DOIT être demandé et utilisé par l'envoyeur pour le receveur avant l'arrivée à expiration du ticket.

### 7.1 Optimisation de l'authentification fondée sur Kerberos

Les tickets Kerberos sont relativement longs (> 500 octets) et il n'est pas nécessaire d'envoyer un ticket dans chaque message RSVP. La clé de session éphémère peut être mise en antémémoire par l'envoyeur et par le receveur et elle peut être utilisée pendant la durée de vie du ticket Kerberos. Dans ce cas, l'envoyeur a seulement besoin d'inclure le ticket Kerberos dans le premier message généré. Les messages RSVP ultérieurs utilisent l'identifiant de clé pour restituer la clé de l'antémémoire (et facultativement les autres informations d'identité) au lieu de passer les tickets de l'envoyeur au receveur dans chaque message RSVP.

Un receveur peut n'avoir pas mis en antémémoire l'état de clé avec un identifiant de clé associé du fait d'un réamorçage ou de changements d'acheminement. Si la politique du receveur indique l'utilisation des clés Kerberos pour les vérifications d'intégrité, le receveur peut renvoyer un messages Mise en cause d'intégrité à l'envoyeur. À réception d'un message Mise en cause d'intégrité, un envoyeur DOIT envoyer un objet Identity qui comporte le ticket Kerberos dans le message Réponse d'intégrité, permettant par là de restituer et mémoriser la clé de session à partir du ticket Kerberos pour les vérifications d'intégrité ultérieures.

## 8. Remerciements

Le présent document dérive directement du travail similaire effectué pour OSPF et RIP version 2, conjointement par Ran Atkinson et Fred Baker. Une révision rédactionnelle significative a été effectuée par Bob Braden, apportant une formulation plus claire. Des commentaires significatifs ont été formulés par Steve Bellovin, qui comprend réellement ce sujet. Matt Crawford et Dan Harkins ont aidé à la révision du document.

## 9. Références

*(Les liens sur les numéros pointent sur la version anglaise d'origine, ceux du corps du titre sur la traduction française.)*

[1] R. Braden, éd., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Protocole de réservation de ressource (RSVP) -- version 1, spécification fonctionnelle", RFC2205, septembre 1997. (*MàJ par* RFC2750, RFC3936, RFC4495) *(P.S.)*

[2] S. Yadav et autres, "Représentation d'identité pour RSVP", RFC2752, janvier 2000. (*Obsolète, voir* RFC3182) *(P.S.)*

[3] S. Kent et R. Atkinson, "Architecture de sécurité pour le protocole Internet", RFC2401, novembre 1998. (*Obsolète, voir* RFC4301)

[4] D. Maughan, M. Schertler, M. Schneider et J. Turner, "Association de sécurité Internet et protocole de gestion de clés (ISAKMP)", RFC2408, novembre 1998. *(Obsolète, voir la RFC4306)*

[5] S. Kent et R. Atkinson, "En-tête d'authentification IP", RFC2402, novembre 1998. (*Obsolète, voir* RFC4302, 4305)

[6] S. Kent et R. Atkinson, "Encapsulation de charge utile de sécurité IP (ESP)", RFC2406, novembre 1998. *(Obsolète, voir RFC4303)*

[7] H. Krawczyk, M. Bellare et R. Canetti, "HMAC : Hachage de clés pour l'authentification de message", RFC2104, février 1997.

[8] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", RFC2119, BCP 14, mars 1997.

[9] J. Postel (éd.), "Protocole de commande de transmission – Spécification du protocole du programme Internet DARPA", RFC0793, (STD 7), septembre 1981.

[10] J. Kohl et C. Neuman, "Service Kerberos d'authentification de réseau (v5)", RFC1510, septembre 1993. *(Obsolète, voir RFC4120)*

## 10. Considérations pour la sécurité

Ce mémoire tout entier décrit et spécifie un mécanisme d'authentification pour RSVP dont il est estimé qu'il est sûr contre les attaques actives et passives.

La qualité de la sécurité fournie par ce mécanisme dépend de la force de l'algorithme d'authentification mis en œuvre, de la force de la clé utilisée, et de la mise en œuvre correcte du mécanisme de sécurité dans toutes les mises en œuvre RSVP communicantes. Ce mécanisme dépend aussi du maintien de la confidentialité des clés d'authentification RSVP par toutes les parties. Si l'une quelconque de ces suppositions se révèle incorrecte ou si les procédures sont insuffisamment sûres, aucune sécurité réelle ne sera fournie à l'utilisateur de ce mécanisme.

Alors que le message de prise de contact "Réponse d'intégrité" est vérifié en intégrité, le message de prise de contact "Mise en cause d'intégrité" ne l'est pas. Cela a été fait intentionnellement pour éviter le cas où les deux routeurs qui échangent du trafic n'ont pas de numéro de séquence de départ pour la clé de l'autre. Par conséquent, ils vont l'un et l'autre continuer d'envoyer des messages de prise de contact "Mise en cause d'intégrité" qui vont être éliminés par l'autre extrémité. De plus, n'exiger que la vérification d'intégrité de la réponse élimine la dépendance à une association de sécurité dans la direction opposée.

Ceci laisse cependant la possibilité qu'un intrus génère de fausses mises en cause de prise de contact avec certains messages de qualification d'audience. Il pourrait alors conserver la réponse et tenter de la répéter à l'égard d'un receveur en cours de récupération. Si il se trouvait assez chanceux pour avoir deviné le message de qualification d'audience utilisé par le receveur au moment de la récupération, il pourrait utiliser la réponse conservée. Cette réponse serait acceptée, car elle est correctement signée, et aurait un plus petit numéro de séquence pour l'envoyeur parce qu'elle vient d'un vieux message. Cela ouvre le receveur aux répétitions. Cela semble quand même très difficile à exploiter. Cela exige non seulement de

deviner le message de qualification d'audience (qui se fonde sur un secret connu en local) à l'avance, mais aussi d'être capable de se faire passer pour le receveur pour générer une "Mise en cause d'intégrité" de prise de contact avec l'adresse IP appropriée, et de ne pas se faire prendre.

La confidentialité n'est pas fournie par ce mécanisme. Si la confidentialité est exigée, IPsec ESP [6] peut être la meilleure approche, bien qu'elle soit sujette aux mêmes critiques que l'authentification IPsec, et ne serait donc applicable qu'à des environnements spécifiques. La protection contre les analyses de trafic n'est également pas fournie. Des mécanismes comme le chiffrement brut de liaison peuvent être utilisés lorsque la protection contre les analyses de trafic est nécessaire.

## 11. Adresse des auteurs

Fred Baker
Cisco Systems
519 Lado Drive
Santa Barbara, CA 93111
téléphone : (408) 526-4257
mél : fred@cisco.com

Bob Lindell
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
téléphone : (310) 822-1511
mél : lindell@ISI.EDU

Mohit Talwar
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
téléphone : +1 425 705 3131
mél : mohitt@microsoft.com

## 12. Appendice 1 Interface de gestion des clés

Cet appendice décrit une interface générique pour la gestion de clés. Cette description est à un niveau abstrait ce qui veut dire que les mises en œuvre pourraient devoir introduire de petites variations pour l'interface réelle.

Au début de l'exécution, RSVP va utiliser cette interface pour obtenir l'ensemble actuel des clés pertinentes pour l'envoi et la réception des messages. Durant l'exécution, RSVP peut demander des clés spécifiques étant donnés un identifiant de clé et une adresse de source, découvrir les clés nouvelles créées, et être informé des clés qui ont été supprimées. L'interface fournit à la fois un style d'interrogation et d'invocation asynchrone pour une applicabilité plus large.

### 12.1 Structures des données

Les informations sur les clés sont retournées en utilisant la structure de données KeyInfo suivante :

```
KeyInfo {
        Type de clé (envoi ou réception)
        Identifiant_de_clé
        Clé
        Type et mode d'algorithme d'authentification
        DébutDeValiditéDeClé
        FinDeValiditéDeClé
        État (Actif ou Supprimé)
        Interface sortante (seulement en envoi)
        Autres critères de choix d'association de sécurité sortante (en envoi seulement, facultatif)
        Adresse du système d'envoi (seulement en réception)
   }
```

### 12.2 Tableau de clé par défaut

Cette fonction retourne une liste de structures de données KeyInfo correspondant à toutes les clés qui sont configurées pour l'envoi et la réception des messages RSVP et ont un état Actif. Cette fonction est habituellement appelée au début de l'exécution mais il n'y a pas de limite au nombre de fois qu'elle peut être invoquée.

```
KM_DefaultKeyTable() -> KeyInfoList
```

### 12.3 Interrogations sur les clés inconnues reçues

Lorsque un message arrive avec une paire Identifiant de clé/Adresse de système d'envoi inconnue, RSVP peut utiliser cette fonction pour interroger le système de gestion de clés sur la clé appropriée. L'état de l'élément retourné, s'il en est, doit être Actif.

KM_GetRecvKey( Objet INTEGRITY, Adresse_de_source ) -> KeyInfo

### 12.4 Appel des mises à jour

Cette fonction retourne une liste de structures de données KeyInfo qui correspondent à tous les changements incrémentaires qui ont été faits au tableau des clés par défaut ou aux clés demandées depuis la dernière invocation à KM_KeyTablePoll, à KM_DefaultKeyTable, ou à KM_GetRecvKey. L'état de certains éléments dans la liste retournée peut être réglé à Supprimé.

KM_KeyTablePoll() -> KeyInfoList

### 12.5 Interface d'invocation asynchrone

Plutôt que d'invoquer de façon répétée la KM_KeyTablePoll(), une mise en œuvre peut choisir d'utiliser un modèle d'événement asynchrone. Cette fonction enregistre l'intérêt pour les changements de clé pour un identifiant de clé donné ou pour toutes les clés si aucun identifiant de clé n'est spécifié. La fonction d'invocation est appelée chaque fois qu'un changement est apporté à une clé.

KM_KeyUpdate ( Fonction [, IdentifiantDeClé ] )

où la fonction d'invocation est paramétrée comme suit :

Fonction ( KeyInfo )

## 13. Déclaration complète de droits de reproduction



Remerciement
Le financement de la fonction d'éditeur des RFC est actuellement assuré par la Internet Society.