Groupe de travail Réseau
Request for Comments : 4976
Catégorie : sur la voie de la normalisation
Traduction Claude Brière de L'Isle

C. Jennings, Cisco Systems, Inc.
R. Mahy, Plantronics
A. B. Roach, Estacado Systems
septembre 2007

Extensions de relais pour le protocole de relais de session de messages (MSRP)

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.
(La présente traduction incorpore les errata 1269 à 1279, 2993 et 2995)

Résumé
Deux modèles distioncts ont été définis pour porter les messages instantanés. Les messages en mode page sont à part et ne font pas partie d'une session du protocole d'initialisation de session SIP, *Session Initiation Protocol*) tandis que les messages en mode session sont établis au titre d'une session qui utilise SIP. Le protocole de relais de session de messages (MSRP, *Message Session Relay Protocol*) est un protocole pour l'échange d'homologue à homologue presque en temps réel de contenus binaires sans intermédiaires, qui est conçu comme étant signalé en utilisant un protocole de rendez-vous distinct comme SIP. Le présent document introduit la notion d'intermédiaire de relais de message dans MSRP et décrit les extensions nécessaires pour les utiliser.

Table des Matières

1. Introduction et exigences

Il existe un certain nombre de scénarios dans lesquels il est souhiatable d'utiliser un protocole séparé pour la messagerie en vrac. En particulier, il y a un besoin de traiter une séquence de messages comme une session de supports initiée en utilisant SIP [RFC3261], juste comme tout autre type de supports. Le protocole de relais de session de messages (MSRP, *Message Session Relay Protocol*) [RFC4975] est utilisé pour porter une session de messages directement entre deux systèmes d'extrémité sans intermédiaire. Avec MSRP, les messages peuvent être d'une longueur arbitraire et tout le trafic est envoyé sur des transports fiables, sans encombrement.

Le présent document décrit des extensions au cœur du protocole MSRP pour introduire des intermédiaires appelés des relais. Avec ces extensions, les clients MSRP peuvent communiquer directement, ou par un nombre arbitraire de relais. Chaque client est chargé d'identifier chaque relais qui agit en son nom et de lui fournir les accréditifs appropriés. Les clients qui peuvent recevoir directement de nouvelles connexions TCP n'ont pas à mettre en œuvre de nouvelles fonctions pour travailler avec ces relais.

Les buts des extensions de relais MSRP sont énumérés ci-dessous :
- o porter des données MIME binaires arbitraires sans modification ni codage de transfert
- o continuer de prendre en charge le fonctionnement de client à client (sans exiger de serveur de relais)
- o fonctionner à travers un nombre arbitraire de relais pour l'application de la politique
- o fonctionner à travers des relais sous des contôles administratifs différents
- o permettre à chaque client de contrôler quels relais sont traversés en son nom
- o empêcher les messages non sollicités (pourriels) les "relais ouverts", et l'amplification de déni de service (DoS)
- o permettre aux relais d'utiliser une connexion TCP ou TLS [RFC4346] ou à un petit nombre de ces connexions de porter des messages pour plusieurs sessions, receveurs, et envoyeurs
- o permettre d'envoyer de grands messages sur des connexions lentes sans causer de problème de blocage de tête de ligne
- o permettre d'interrompre et reprendre la transmission de grands messages dans les endroits où la connexité du réseau est perdue et ensuite rétablie
- o offrir la notification de l'échec d'un message à tout intermédiaire
- o permettre aux relais de supprimer l'état après un court délai.

2. Conventions et définitions

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].

Voici une liste de plusieurs définitions importantes pour MSRP :

Nœud MSRP : hôte qui met en œuvre les protocoles MSRP comme client ou comme relais.

Client MSRP : nœud MSRP qui est l'envoyeur initial ou la cible finale des messages et de l'état de livraison.

Relais MSRP : nœud MSRP qui transmet les messages et l'état de livraison et peut fournir l'application de politique. Les relais peuvent fragmenter et réassembler les portions de messages.

Message : contenu MIME [RFC2045], [RFC2046] arbitraire qu'un client souhaite envoyer à un autre. Pour les besoins de la

présente spécification, un corps MIME complet par opposition à une portion d'un message complet.

tronçon : portion d'un message complet livrée dans une demande SEND.

de bout en bout : livraison de données du client initiateur au client cible final.

bond : livraison de données entre un nœud MSRP et un nœud adjacent.

3. Vue d'ensemble du protocole

Avec l'introduction de cette extension, MSRP a les deux concepts de clients et de relais. Les clients envoient des messages aux relais et/ou à d'autres clients. Les relais transmettent les messages et l'état de livraison des messages aux clients et aux autres relais. Les clients qui peuvent ouvrir des connexions TCP entre eux sans restrictions de politique peuvent communiquer directement les uns avec les autres. Les clients qui se trouvent derrière des pare-feu ou qui doivent utiliser des intermédiaires pour des raisons de politique peuvent recourir aux services d'un relais. Chaque client est chargé d'obtenir l'aide d'un ou de plusieurs relais pour son côté de la communication.

Les clients qui utilisent un relais commencent par ouvrir une connexion TLS avec un relais, s'authentifient, et récupèrent un URI msrps: (du relais) que le client peut fournir à ses homologues pour recevoir ultérieurement des messages. Il y a plusieurs étapes pour ce faire. D'abord, le client ouvre une connexion TLS à son premier relais et vérifie que le nom figurant dans le certificat correspond au nom du relais auquel il essaie de se connecter. Cette vérification s'effectue selon les procédures définies au paragraphe 9.2. Après avoir vérifié qu'il s'est connecté à l'hôte approprié, le client s'authentifie auprès du relais à l'aide d'une demande AUTH contenant les accréditifs d'authentification appropriés. Dans une réponse AUTH réussie, le relais fournit un URI msrps: associé au chemin de retour au client. Le client peut alors donner cet URI aux autres clients pour la livraison de bout en bout du message.

Quand les clients souhaitent envoyer un message court, ils produisent une demande SEND avec le contenu entier du message. Si des relais sont nécessaires, ils sont inclus dans l'en-tête To-Path. L'URI le plus à gauche dans l'en-tête To-Path est le prochain bond pour livrer une demande ou réponse. L'URI le plus à droite dans l'en-tête To-Path est la cible finale.

Les demandes SEND contiennent des en-têtes qui indiquent comment elles sont acquittées sous forme bond par bond et de bout en bout. Par défaut, les messages SEND font l'objet d'un accusé de réception bond par bond. (Chaque relais qui reçoit une demande SEND accuse réception de la demande avant de transmettre le contenu au relais suivant ou à la cible finale). Toutes les autres demandes font l'objet d'un accusé de réception de bout en bout.

Avec l'introduction des relais, la sémantique subtile des en-têtes To-Path et From-Path devient plus pertinente. Le To-Path, tant dans les demandes que dans les réponses, est la liste des URI qui doivent être visités pour atteindre la cible finale de la demande ou réponse. Le From-Path est la liste des URI qui indiquent comment retourner à l'envoyeur initial de la demande ou réponse. Ces en-têtes diffèrent des en-têtes To et From de SIP, qui ne sont pas "échangés" entre la demande et la réponse. (Noter que parfois une demande est envoyée directement à ou d'un intermédiaire).

Quand un relais transmet une demande, il supprime son adresse de l'en-tête To-Path et l'insère comme premier URI dans l'en-tête From-Path. Par exemple, si le chemin d'Alice à Bob passe par les relais A et B, quand B reçoit la demande, elle contient des en-têtes de chemin qui ressemblent à ce qui suit. (Noter que MSRP ne permet pas de revenir à la ligne. Un "\" dans les exemples montre une continuation de ligne en raison des limitations de longueur de ligne de ce document. Ni la barre oblique inverse ni le CRLF supplémentaire ne sont inclus dans la demande ou réponse réelles).

```
To-Path: msrps://B.exemple.com/bbb;tcp msrps://Bob.exemple.com/bob;tcp
From-Path: msrps://A.exemple.com/aaa;tcp msrps://Alice.exemple.com/alice;tcp
```

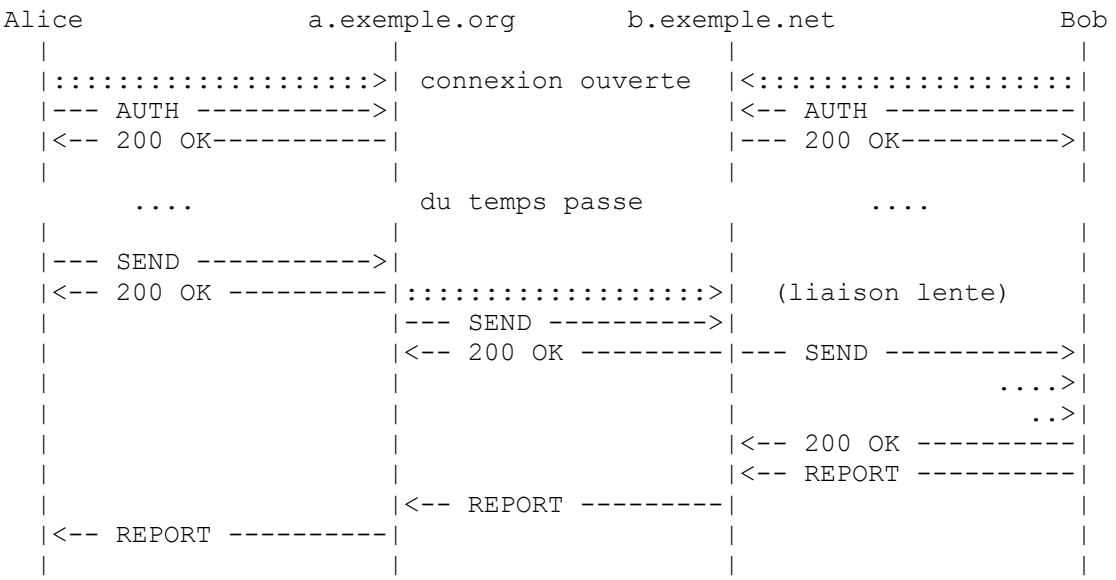
Après la transmission de la demande les en-têtes path ressemblent à ceci :

```
To-Path: msrps://Bob.exemple.com/bob;tcp
From-Path: msrps://B.exemple.com/bbb;tcp msrps://A.exemple.com/aaa;tcp msrps://Alice.exemple.com/alice;tcp
```

L'envoi d'un accusé de réception pour les demandes SEND est contrôlé par les en-têtes Success-Report et Failure-Report et fonctionne de la même manière que dans le protocole MSRP de base. Quand un relais reçoit une demande SEND, si Failure-Report a la valeur "oui", cela signifie que le bond précédent utilise un temporisateur et que le relais doit envoyer une réponse à la demande. Si la réponse finale contient une erreur, le bond précédent est chargé d'élaborer le rapport

d'erreur et de le renvoyer à l'envoyeur original du message. La réponse 200 accuse réception de la demande afin que le bond précédent sache qu'il n'est plus responsable de la demande. Si le relais sait qu'il ne sera pas capable de livrer la demande et si le Failure-Report a une valeur autre que "non", il envoie alors un REPORT pour informer l'expéditeur de l'erreur. Si la valeur "Failure-Report" est "oui", une fois que le relais a fini d'envoyer la demande au bond suivant, il lance un temporisateur ; si le temporisateur expire avant qu'une réponse soit reçue du bond suivant, le relais suppose qu'une erreur s'est produite et envoie un REPORT à l'expéditeur. Si le rapport d'échec n'est pas réglé à "oui", le relais n'a pas besoin de lancer ce temporisateur.

L'exemple suivant montre une session MSRP typique. Les demandes AUTH sont expliquées dans un paragraphe ultérieur mais sont laissées dans l'exemple pour que les flux d'appel soient complets.



Les messages SEND et REPORT sont montrés ci-dessous pour illustrer les en-têtes To-Path et From-Path. (Noter que MSRP ne permet par le renvoi à la ligne. Un "\" dans les exemples montre une continuation de ligne due aux limitations de longueur de ligne de ce document. Ni la barre obligue, ni le CRLF supplémentaire ne sont inclus dans la demande ou réponse réelles.)

```
MSRP 6aef SEND
To-Path: msrps://a.exemple.org:9000/kjfjan;tcp msrps://b.exemple.net:9000/aeiug;tcp \
        msrps://bob.exemple.net:8145/foo;tcp
From-Path: msrps://alice.exemple.org:7965/bar;tcp
Success-Report: oui
Byte-Range: 1-*/*
Message-ID: 87652
Content-Type: text/plain

Hé Bob, je vais t'envoyer un fichier mpeg
-------6aef$
```

```
MSRP 6aef 200 OK
To-Path: msrps://alice.exemple.org:7965/bar;tcp
From-Path: msrps://a.exemple.org:9000/kjfjan;tcp
Message-ID: 87652
-------6aef$
```

```
MSRP juh76 SEND
To-Path: msrps://b.exemple.net:9000/aeiug;tcp msrps://bob.exemple.net:8145/foo;tcp
From-Path: msrps://a.exemple.org:9000/kjfjan;tcp msrps://alice.exemple.org:7965/bar;tcp
Success-Report: oui
Message-ID: 87652
Byte-Range: 1-*/*
Content-Type: text/plain
```

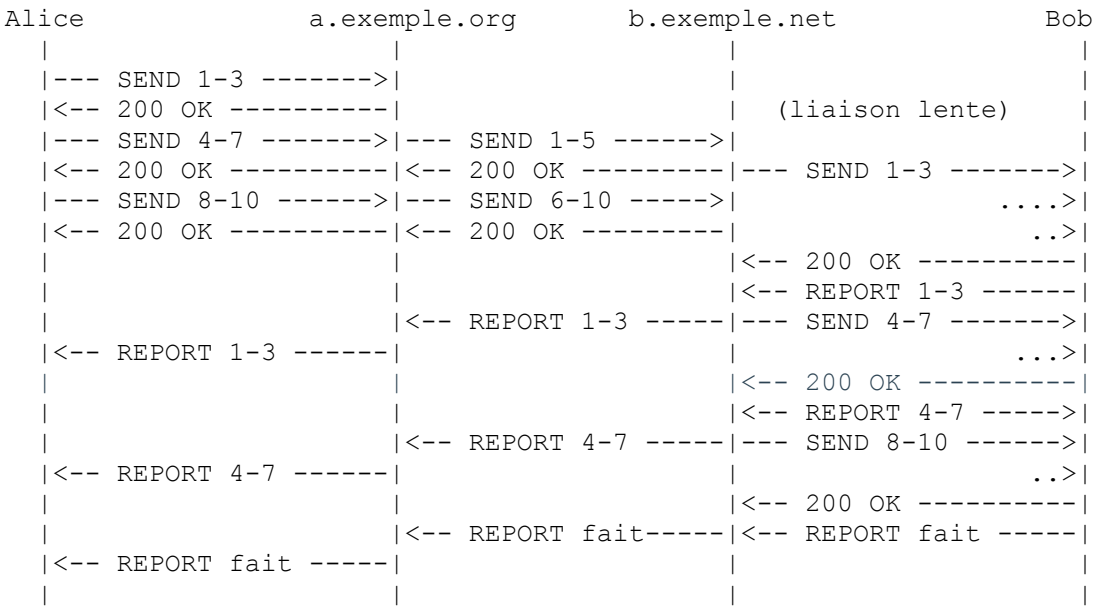
```
Hé Bob,  je vais t'envoyer un fichier mpeg
-------juh76$

MSRP juh76 200 OK
To-Path: msrps://a.exemple.org:9000/kjfjan;tcp
From-Path: msrps://b.exemple.net:9000/aeiug;tcp
Message-ID: 87652
-------juh76$

MSRP xght6 SEND
To-Path: msrps://bob.exemple.net:8145/foo;tcp
From-Path: msrps://b.exemple.net:9000/aeiug;tcp msrps://a.exemple.org:9000/kjfjan;tcp \
           msrps://alice.exemple.org:7965/bar;tcp
Success-Report: oui
Message-ID: 87652
Byte-Range: 1-*/*
Content-Type: text/plain

Hé Bob, je vais t'envoyer un fichier mpeg
-------xght6$

MSRP xght6 200 OK
To-Path: msrps://b.exemple.net:9000/aeiug;tcp
From-Path: msrps://bob.exemple.net:8145/foo;tcp
Message-ID: 87652

MSRP yh67 REPORT
To-Path: msrps://b.exemple.net:9000/aeiug;tcp msrps://a.exemple.org:9000/kjfjan;tcp \
         msrps://alice.exemple.org:7965/bar;tcp
From-Path: msrps://bob.exemple.net:8145/foo;tcp
Message-ID: 87652
Byte-Range: 1-39/39
Status: 000 200 OK
-------yh67$

MSRP yh67 REPORT
To-Path: msrps://a.exemple.org:9000/kjfjan;tcp msrps://alice.exemple.org:7965/bar;tcp
From-Path: msrps://b.exemple.net:9000/aeiug;tcp msrps://bob.exemple.net:8145/foo;tcp
Message-ID: 87652
Byte-Range: 1-39/39
Status: 000 200 OK
-------yh67$

MSRP yh67 REPORT
To-Path: msrps://alice.exemple.org:7965/bar;tcp
From-Path: msrps://a.exemple.org:9000/kjfjan;tcp msrps://b.exemple.net:9000/aeiug;tcp \
           msrps://bob.exemple.net:8145/foo;tcp
Message-ID: 87652
Byte-Range: 1-39/39
Status: 000 200 OK
-------yh67$
```

Quand il envoie de grands contenus, le client peut partager un message en plus petits morceaux ; chaque demande SEND pourrait ne contenir qu'une portion du message complet. Par exemple, quand Alice envoie à Bob un fichier de 4 G bits appelé "file.mpeg", elle envoie plusieurs demandes SEND dont chacune contient une portion du message complet. Les relais peuvent rempaqueter les fragments de message en route. Quand les portions individuelles du message complet arrivent au client de la destination finale, le client receveur peut facultativement envoyer des demndes REPORT qui indiquent l'état de livraison.

Les nœuds MSRP peuvent envoyer des portions individuelles d'un message complet en plusieurs demandes SEND. Quand

les relais reçoivent les tronçons, ils peuvent les réassembler ou les refragmenter pour autant qu'ils envoient les tronçons résultants dans l'ordre. (Les receveurs doivent cependant quand même être prêts à recevoir des tronçons déclassés.) Si l'envoyeur a réglé la valeur du champ d'en-tête Success-Report à "oui", une fois qu'un tronçon ou message complet arrive au client de destination, la destination va envoyer une demande REPORT indiquant qu'un tronçon est arrivé de bout en bout. Cette demande revient sur le chemin inverse de la demande SEND. À la différence de la demande SEND, qui peut être acquittée le long de chaque bond, les demandes REPORT ne reçoivent jamais d'accusé de réception.

L'exemple suivant montre un message re-tronçonnné sur deux relais :



Les relais ne gardent les états de transaction qu'un bref instant pour chaque tronçon. La livraison sur chaque bond devrait ne pas prendre plus de 30 secondes après l'envoi du dernier octet de données. Les applications de clients définissent leurs propres temporisateurs dépendants de la mise en œuvre pour la livraison de message de bout en bout.

Pour la communication de client à client, l'envoyeur d'un message ouvre normalement une nouvelle connexion TCP (avec ou sans TLS) si c'est nécessaire. Les relais réutilisent d'abord les connexions existantes, mais peuvent ouvrir de nouvelles connexions (normalement avec d'autres relais) pour livrer des demandes comme SEND ou REPORT. Les réponses peuvent seulement être envoyées sur des connexions existantes.

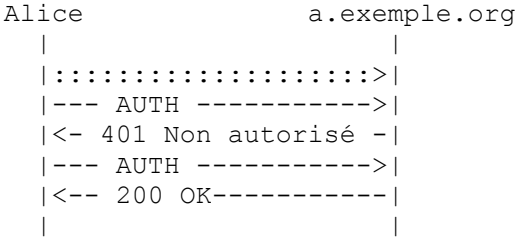
La relation entre MSRP et les protocoles de signalisation (comme SIP) n'est pas changée par le présent document, et est comme décrit dans la [RFC4975]. Un exemple d'échange SDP pour une session MSRP impliquant des relais est montré à la Section 11.

3.1 Vue d'ensemble de Autorisation

Un élément clé de ce protocole est qu'il ne peut pas introduire de relais ouverts, avec tous les problèmes associés qu'ils créent, incluant des attaques de DoS. Un message n'est transmis par un relais que si il va ou vient d'un client qui a authentifié le relais et a été autorisé à relayer des messages sur cette session particulière. À cause de cela, les clients utilisent un message AUTH pour s'authentifier auprès d'un relais et obtenir un URI qui peut être utilisé pour transmettre les messages.

Si un client souhaite utiliser un relais, il envoie une demande AUTH au relais. Le client authentifie le relais en utilisant le certificat TLS du relais. Le client utilise l'authentification par résumé HTTP [RFC2617] pour s'authentifier auprès du relais. Quand l'authentification réussit, le relais retourne une réponse 200 qui contient l'URI que le client peut utiliser dans le chemin MSRP pour le relais.

On montre ci-dessous un flux typique de défi/réponse :



L'URI que le client devrait utiliser est retourné dans le champ d'en-tête Use-Path du 200.

Noter que les URI retournés au client sont effectivement des jetons secrets qui devraient n'être partagés qu'avec les autres clients MSRP dans une session. Pour cette raison, le client NE DOIT PAS réutiliser le même URI pour plusieurs sessions, et doit protéger ces URI contre l'espionnage.

4. Nouveaux éléments de protocole

4.1 Méthode AUTH

Les demandes AUTH sont utilisées par les clients pour créer une bride qu'ils peuvent utiliser pour recevoir les demandes entrantes. Les demandes AUTH contiennent aussi des accréditifs utilisés pour authentifier un client et la politique d'autorisation utilisée pour bloquer les attaques de déni de service.

En réponse à une demande AUTH, une réponse de succès contient un champ d'en-tête Use-Path avec une liste d'URI que le client peut donner à ses homologues pour acheminer les réponses en retour au client.

4.2 Champ d'en-tête Use-Path

Le champ d'en-tête Use-Path contient une liste d'URI fournis par un relais MSRP en réponse à une demande AUTH réussie. Cette liste d'URI peut être utilisée par le client MSRP qui envoie la demande AUTH pour recevoir des demandes MSRP et pour annoncer cette liste d'URI, par exemple, dans une description de session. Les URI dans le champ d'en-tête Use-Path DOIVENT inclure un nom de domaine pleinement qualifié (par opposition à une adresse numérique IP) et un numéro d'accès explicite.

Les URI dans le champ d'en-tête Use-Path sont dans le même ordre que celui que le client qui s'authentifie utilise dans un champ d'en-tête To-Path. Les instructions sur la formation des champs d'en-tête To-Path et des attributs de chemin SDP [RFC4566] à partir des informations dans le champ d'en-tête Use-Path sont fournies au paragraphe 5.1.

4.3 Champ d'en-tête d'authentification HTTP "WWW-Authenticate"

Le champ d'en-tête "WWW-Authenticate" contient un jeton de défi utilisé dans la procédure d'authentification par résumé HTTP (d'après la [RFC2617]). L'usage de l'authentification par résumé HTTP dans MSRP est décrit en détails au paragraphe 5.1.

4.4 Champ d'en-tête d'authentification HTTP "Authorization"

Le champ d'en-tête "Authorization" contient les accréditifs d'authentification pour l'authentification par résumé HTTP (d'après la [RFC2617]). L'usage de l'authentification par résumé HTTP dans MSRP est décrit en détails au paragraphe 5.1.

4.5 Champ d'en-tête d'authentification HTTP "Authentication-Info"

Le champ d'en-tête "Authentication-Info" contient les défis futurs à utiliser pour l'authentification par résumé HTTP (d'après la [RFC2617]). L'usage de l'authentification par résumé HTTP dans MSRP est décrit en détails au paragraphe 5.1.

4.6 Champs d'en-tête relatifs au temps

Le champ d'en-tête Expires dans une demande donne un temps relatif après lequel l'action impliquée par la méthode de la

demande n'a plus d'intérêt. Dans une demande, le champ d'en-tête Expires indique pendant combien de temps l'envoyeur voudrait que la demande reste valide. Dans une réponse, le champ d'en-tête Expires indique pendant combien de temps celui qui répond considère que ces informations sont pertinentes. Précisément, un champ d'en-tête Expires dans une réponse AUTH indique pendant combien de temps les URI fournis vont être valides.

Le champ d'en-tête Min-Expires contient la durée minimum pendant laquelle un serveur va permettre un champ d'en-tête Expires. Il n'est envoyé que dans des réponses 423 "Intervalle hors limites". De même, le champ d'en-tête Max-Expires contient la durée maximum pendant laquelle un serveur va permettre un champ d'en-tête Expires.

5. Comportement du client

5.1 Connexion aux relais agissant au nom du client

Les clients qui veulent utiliser les services d'un relais ou d'une liste de relais doivent envoyer une demande AUTH à chaque relais qui agira en leur nom. (Par exemple, certaines organisations pourraient déployer un relais "intra-org" et un relais "extra-org"). Le relais interne est utilisé pour tunneler les demandes AUTH vers le relais externe. Par exemple, le client va envoyer une demande AUTH à intra-org et recevra en retour un chemin qui peut être utilisé pour la transmission à travers intra-org. Le client enverra ensuite une deuxième demande AUTH destinée à extra-org mais passant par intra-org. Le relais intra-org la transmet à extra-org et extra-org retourne un chemin qui peut être utilisé pour envoyer des messages provenant d'une autre destination à extra-org, à intra-org et ensuite à ce client. Chaque relais authentifie le client. Le client authentifie le premier relais et chaque relais authentifie le relais suivant.

Les clients peuvent être configurés (normalement, par découverte ou provisionnement manuel) avec une liste de relais qu'ils doivent utiliser. Ils DOIVENT être capables d'établir une connexion avec le premier relais et d'envoyer une commande AUTH pour obtenir un URI qui peut être utilisé dans un champ d'en-tête To-Path. Le client peut authentifier son premier relais en consultant le certificat TLS du relais. Le client DOIT s'authentifier auprès de chacun de ses relais en utilisant l'authentification par résumé HTTP [RFC2617] (voir les détails au paragraphe 9.1).

Le relais renvoie un URI, ou une liste d'URI, dans le champ d'en-tête "Use-Path" d'une réponse de succès. Chaque URI DEVRAIT être utilisé pour une seule et unique session. Ces URI sont utilisés par le client dans l'attribut path qui est envoyé dans le SDP pour établir la session, et dans le champ d'en-tête To-Path des demandes sortantes. Pour former le champ d'en-tête To-Path des demandes sortantes, le client prend la liste des URI dans le champ d'en-tête Use-Path après l'authentification la plus externe et ajoute la liste des URI fournie dans l'attribut path de la description de session de l'homologue. Pour former l'attribut de chemin SDP à fournir à l'homologue, le client inverse la liste des URI dans le champ d'en-tête Use-Path (après l'authentification la plus externe) et y ajoute son propre URI.

Par exemple, "A" doit traverser ses propres relais "B" et "C", puis les relais "D" et "E" dans le domaine 2 pour atteindre "F". Le client "A" s'authentifie auprès de ses relais "B" et "C" et reçoit finalement un champ d'en-tête Use-Path contenant "B C". Le client "A" inverse la liste (maintenant "C B") et ajoute son propre URI (maintenant "C B A"), et fournit cette liste à "F" dans un attribut de chemin SDP. Le client "F" envoie sa liste de chemins SDP "D E F", que le client "A" ajoute à la liste Use-Path "B C" qu'il a reçue. Le champ d'en-tête To-Path résultant est "B C D E F".

Use-Path retourné par C : B C
Attribut path : généré par A : C B A
Attribut path : reçu de F : D E F
Champ d'en-tête To-Path généré par A : B C D E F

La demande AUTH initiale envoyée à un relais par un client ne contiendra généralement pas de champ d'en-tête Authorization, car le client n'a pas de défi auquel répondre. En réponse à une demande AUTH qui ne contient pas de champ d'en-tête Autorisation, un relais DOIT répondre par une réponse "401 Non autorisé" contenant un champ d'en-tête WWW-Authenticate. Le champ d'en-tête WWW-Authenticate est formé comme décrit dans la [RFC2617], avec les restrictions et modifications décrites au paragraphe 9.1. Le domaine choisi par le relais MSRP dans un tel défi est une question de politique administrative. Comme un seul relais n'a pas plusieurs espaces de protection dans MSRP, il n'est pas déraisonnable de toujours utiliser le nom d'hôte du relais comme domaine.

À réception d'une réponse 401 à une demande, le client DEVRAIT aller chercher le domaine dans le champ d'en-tête WWW-Authenticate dans la réponse et réessayer la demande, en incluant un champ d'en-tête Authorization avec les accréditifs corrects pour le domaine. Le champ d'en-tête Authorization est formé comme décrit dans la [RFC2617], avec les restrictions et modifications décrites au paragraphe 9.1.

Quand un client souhaite utiliser plus d'un relais, il DOIT envoyer une demande AUTH à chaque relais qu'il souhaite utiliser. Considérons un client A, qui souhaite que les messages s'écoulent de A au premier relais, R1, puis à un second relais, R2. Ce client va faire un AUTH normal avec R1. Il va ensuite faire une transaction AUTH avec R2 qui est acheminée à travers R1. Le client va former ce message AUTH en réglant le To-Path à msrps://R1;tcp msrps://R2;tcp. R1 va transmettrs cette demande jusqu'à R2.

Quand il envoie une demande AUTH, le client PEUT ajouter un champ d'en-tête Expires pour demander un URI MSRP qui n'est valide que pour l'intervalle fourni (un nombre entier de secondes). Le serveur va inclure un champ d'en-tête Expires dans une réponse de succès indiquant pendant combien de temps son URI dans le Use-Path va être valide. Noter que chaque serveur peut retourner un temps d'expiration indépendant.

Noter que MSRP ne permet par le renvoi à la ligne. Un "\" dans les exemples montre une continuation de ligne due aux limitations de longueur de ligne dans le présent document. Ni la barre oblique inverse ni le CRLF supplémentaire ne sont inclus dans les demandes ou réponses réelles.

(Alice ouvre une connexion TLS à intra.exemple.com et envoie une demande AUTH pour initier le processus d'authentification.)

```
MSRP 49fh AUTH
To-Path: msrps://alice@intra.exemple.com;tcp
From-Path: msrps://alice.exemple.com:9892/98cjs;tcp
-------49fh$
```

(Le relais d'Alice met au défi la demande AUTH.)

```
MSRP 49fh 401 Non autorisé
To-Path: msrps://alice.exemple.com:9892/98cjs;tcp
From-Path: msrps://alice@intra.exemple.com;tcp
WWW-Authenticate: Digest realm="intra.exemple.com", qop="auth", \ nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093"
-------49fh$
```

(Alice répond au défi.)

```
MSRP 49fi AUTH
To-Path: msrps://alice@intra.exemple.com;tcp
From-Path: msrps://alice.exemple.com:9892/98cjs;tcp
Authorization: Digest username="Alice",
            realm="intra.exemple.com", \
            nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093", \
            qop=auth, nc=00000001, cnonce="0a4f113b", \
            response="6629fae49393a05397450978507c4ef1"
-------49fi$
```

(Le relais d'Alice confirme qu'Alice est un utilisateur autorisé. Au titre de la politique locale, il inclut un champ d'en-tête "Authentication-Info" avec une nouvelle valeur de nom occasionnel pour traiter les futures demandes AUTH.)

```
MSRP 49fi 200 OK
To-Path: msrps://alice.exemple.com:9892/98cjs;tcp
From-Path: msrps://alice@intra.exemple.com;tcp
Use-Path: msrps://intra.exemple.com:9000/jui787s2f;tcp
Authentication-Info: nextnonce="40f2e879449675f288476d772627370a",\
                rspauth="7327570c586207eca2afae94fc20903d", cnonce="0a4f113b", nc=00000001, qop=auth
Expires: 900
-------49fi$
```

(Alice envoie maintenant une demande AUTH à son relais "external" par son relais "internal", en utilisant l'URI qu'elle

vient d'obtenir ; la demande AUTH fait l'objet d'un défi.)

```
MSRP mnbvw AUTH
To-Path: msrps://intra.exemple.com:9000/jui787s2f;tcp msrps://extra.exemple.com;tcp
From-Path: msrps://alice.exemple.com:9892/98cjs;tcp
-------mnbvw$

MSRP m2nbvw AUTH
To-Path: msrps://extra.exemple.com;tcp
From-Path: msrps://intra.exemple.com:9000/jui787s2f;tcp msrps://alice.exemple.com:9892/98cjs;tcp
-------m2nbvw$

MSRP m2nbvw 401 Unauthorized
To-Path: msrps://intra.exemple.com:9000/jui787s2f;tcp msrps://alice.exemple.com:9892/98cjs;tcp
From-Path: msrps://extra.exemple.com;tcp
WWW-Authenticate: Digest realm="extra.exemple.com", qop="auth", \
                  nonce="Uumu8cAV38FGsEF31VLevIbNXj9HWO"
-------m2nbvw$

MSRP mnbvw 401 Unauthorized
To-Path: msrps://alice.exemple.com:9892/98cjs;tcp
From-Path: msrps://intra.exemple.com:9000/jui787s2f;tcp msrps://extra.exemple.com;tcp
WWW-Authenticate: Digest realm="extra.exemple.com", qop="auth", \
                                        nonce="Uumu8cAV38FGsEF31VLevIbNXj9HWO"
-------mnbvw$
```

(Alice répond au défi avec ses accréditifs et est ensuite autorisée à utiliser le relais "external").

```
MSRP m3nbvx AUTH
To-Path: msrps://intra.exemple.com:9000/jui787s2f;tcp msrps://extra.exemple.com;tcp
From-Path: msrps://alice.exemple.com:9892/98cjs;tcp
Authorization: Digest username="Alice",
               realm="extra.exemple.com", \
               nonce="Uumu8cAV38FGsEF31VLevIbNXj9HWO", \
               qop=auth, nc=00000001, cnonce="85a0dca8", \
               response="cb06c4a77cd90918cd7914432032e0e6"
-------m3nbvx$

MSRP m4nbvx AUTH
To-Path: msrps://extra.exemple.com;tcp
From-Path: msrps://intra.exemple.com:9000/jui787s2f;tcp msrps://alice.exemple.com:9892/98cjs;tcp
Authorization: Digest username="Alice",
               realm="extra.exemple.com", \
               nonce="Uumu8cAV38FGsEF31VLevIbNXj9HWO", \
               qop=auth, nc=00000001, cnonce="85a0dca8", \
               response="cb06c4a77cd90918cd7914432032e0e6"
-------m4nbvx$

MSRP m4nbvx 200 OK
To-Path: msrps://intra.exemple.com:9000/jui787s2f;tcp msrps://alice.exemple.com:9892/98cjs;tcp
From-Path: msrps://extra.exemple.com;tcp
Use-Path: msrps://intra.exemple.com:9000/mywdEe1233;tcp
Authentication-Info: nextnonce="bz8V080GEA2sLyEDpITF2AZCq7gIkc", \
                     rspauth="72f109ed2755d7ed0d0a213ec653b3f2", \
                     cnonce="85a0dca8", nc=00000001, qop=auth
Expires: 1800
-------m4nbvx$

MSRP m3nbvx 200 OK
To-Path: msrps://alice.exemple.com:9892/98cjs;tcp
From-Path: msrps://intra.exemple.com:9000/jui787s2f;tcp msrps://extra.exemple.com;tcp
```

```
Use-Path: msrps://extra.exemple.com:9000/jui787s2f;tcp \
          msrps://extra.exemple.com:9000/mywdEe1233;tcp
Authentication-Info: nextnonce="bz8V080GEA2sLyEDpITF2AZCq7gIkc", \
                     rspauth="72f109ed2755d7ed0d0a213ec653b3f2", \
                     cnonce="85a0dca8", nc=00000001, qop=auth
Expires: 1800
-------m3nbvx$
```

5.2 Envoi des demandes

La procédure de formation les demandes SEND et REPORT est identique pour les clients, qu'il y ait ou non des relais. Les procédures spécifiques sont décrites dans la Section 7 du protocole MSRP de base.

Comme d'habitude, une fois que l'URI du prochain bond est déterminé, le client DOIT trouver l'adresse, l'accès et le transport appropriés à utiliser, puis vérifier s'il existe déjà une connexion appropriée avec la cible du prochain bond. Si c'est le cas, le client DOIT envoyer la demande via la connexion la plus appropriée. L'adéquation peut être déterminée par divers facteurs tels que la charge mesurée et la politique locale, mais dans la plupart des mises en œuvre simples, une connexion sera adéquate si elle existe et est active.

5.3 Réception des demandes

La procédure pour recevoir les demandes est identique pour les clients, que des relais soient impliqués ou non.

5.4 Gestion des connexions

Les clients devraient ouvrir une connexion chaque fois qu'ils souhaitent livrer une demande et qu'il n'existe pas de connexion appropriée. Pour les connexions aux relais, le client devrait laisser une connexion ouverte jusqu'à ce qu'aucune session ne l'ait utilisée pendant une période de temps définie localement, qui est par défaut de 5 minutes pour les relais étrangers et d'une heure pour les relais du client.

6. Comportement des relais

6.1 Traitement des connexions entrantes

Quand un relais reçoit une connexion entrante sur un accès configuré pour TLS, il inclut une CertificateRequest du client dans le même enregistrement que celui dans lequel il envoie son ServerHello. Si le client TLS fournit un certificat, le serveur le vérifie et continue si le certificat est valide et a pour racine une autorité de confiance. Si le client TLS ne fournit pas de certificat, le serveur suppose que le client est un point d'extrémité MSRP et invoque l'authentification par résumé. Une fois qu'un canal TCP ou TLS est négocié, le serveur attend jusqu'à 30 secondes la réception d'une demande MSRP sur le canal. Si aucune demande n'est reçue dans ce délai, le serveur ferme la connexion. Si aucune demande réussie n'est envoyée pendant cette période probatoire, le serveur ferme la connexion. De même, si plusieurs demandes infructueuses sont envoyées pendant la période probatoire et si aucune demande n'est envoyée avec succès, le serveur DEVRAIT fermer la connexion.

6.2 Comportement générique de demande

À réception d'une nouvelle demande, les relais vérifient d'abord la validité de la demande. Les relais examinent ensuite le premier URI dans le champ d'en-tête To-Path et suppriment cet URI s'il correspond à un URI correspondant au relais. Si la demande n'est pas adressée au relais, celui-ci abandonne immédiatement la connexion correspondante sur laquelle la demande a été reçue.

6.3 Réception des demandes AUTH

Quand un relais reçoit une demande AUTH, la première chose qu'il fait est d'authentifier et autoriser le bond précédent et le client à l'extrémité distante. Si il n'y a pas d'autre relais entre ce relais et le client, ce sont alors la même chose.

Quand le bond précédent est un relais, l'authentification est faite avec TLS en utilisant l'authentification mutuelle. Si le client TLS a présenté un certificat d'hôte, le relais vérifie que le subjectAltName dans le certificat du client TLS correspond au nom d'hôte dans le premier URI From-Path. Si le client TLS ne fournit pas de certificat d'hôte, le relais suppose que le client TLS est un client MSRP et lui envoie un défi.

L'autorisation est une affaire de politique locale au relais. De nombreux relais vont choisir d'autoriser tous les relais qui peuvent être authentifiés, éventuellement en conjonction avec un mécanisme de liste noire. Les relais destinés à ne fonctionner que dans une fédération limitée peuvent choisir d'autoriser seulement les relais dont l'identité apparaît dans une liste provisionnée. D'autres politiques d'autoriqation peuvent aussi être appliquées.

Quand le bond précédent est un client, le bond précédent est identique à l'identité du client. Le relais vérifie les accréditifs (nom d'utilisateur et mot de passe) fournis par le client dans l'en-tête Authorization et vérifie si ce client est autorisé à utiliser le relais. Si le client n'est pas autorisé, le relais renvoie une réponse 403. Si le client a demandé un délai d'expiration particulier dans un champ d'en-tête Expires, le relais doit vérifier que ce délai est acceptable et, si il ne l'est pas, renvoyer une erreur contenant un champ d'en-tête Min-Expires ou Max-Expires, comme approprié.

Ensuite, le relais va générer un URI MSRP qui permet aux messages d'être transmis vers ou depuis ce bond précédent. Si le bond précédent était un relais authentifié par TLS mutuel, l'URI DOIT être alors valide pour acheminer les messages à travers toute connexion que le relais a avec le bond précédent. Si le bond précédent est un client, l'URI ne doit alors être valide que pour passer par la même connexion que celle par laquelle la demande AUTH a été reçue. Si la connexion du client est fermée puis rouverte, l'URI DOIT être invalidé.

Si la demande AUTH contient un champ d'en-tête Expires, le relais DOIT s'assurer que l'URI est invalidé après le délai d'expiration. L'URI DOIT contenir au moins 64 bits de matériel de chiffrement aléatoire afin qu'il ne puisse pas être deviné par des attaquants. Si il est demandé à un relais d'envoyer un message pour lequel l'URI n'est pas valide, le relais doit éliminer le message et PEUT envoyer un REPORT indiquant que l'URI AUTH était mauvais.

Une réponse AUTH réussie renvoie un champ d'en-tête Use-Path contenant un URI MSRP que le client peut utiliser. Elle renvoie aussi un champ d'en-tête Expires qui indique la durée de validité de l'URI (exprimée en nombre entier de secondes).

Si un relais reçoit plusieurs demandes AUTH infructueuses d'un client qui lui est directement connecté via TLS, le relais DEVRAIT mettre fin à la connexion correspondante. De même, si un relais transmet à la même destination plusieurs demandes AUTH infructueuses provenant d'un client qui lui est directement connecté via TLS, le relais DEVRAIT mettre fin à la connexion correspondante. La détermination d'un échec d'AUTH à distance peut se faire par l'observation d'une demande AUTH contenant un champ d'en-tête Authorization qui déclenche une réponse 401 sans indication "stale=TRUE". Ces mesures préventives ne s'appliquent qu'à une connexion entre un relais et un client ; un relais NE DEVRAIT PAS utiliser un nombre excessif d'échecs de requêtes AUTH comme raison pour mettre fin à une connexion avec un autre relais.

6.4 Transmission

Avant que toute demande soit transmise, le relais DOIT vérifier que le premier URI dans le champ d'en-tête To-Path correspond à un URI que ce relais a créé et distribué dans le champ d'en-tête Use-Path d'une demande AUTH. Ensuite, il vérifie que 1) le prochain bond est le prochain bond vers le client qui a obtenu cet URI, ou 2) le bond précédent était le bond précédent correct en provenance du client qui a obtenu cet URI.

Comme les valeurs transact-id ne sont pas autorisées à entrer en conflit sur une connexion donnée, un relais devra généralement construire une nouvelle valeur transact-id pour toute demande qu'il transmet.

6.4.1 Transmission des demandes SEND

Si une demande SEND entrante contient un champ d'en-tête Failure-Report d'une valeur de "oui", un relais MSRP qui reçoit cette demande SEND DOIT répondre immédiatement avec une réponse finale. Une réponse de classe 200 indique le succès de la réception d'un fragment de message mais ne signifie pas que le message a été transmis au prochain bond. La réponse finale au SEND DOIT être seulement envoyée au bond précédent, qui pourrait être un relais MSRP ou l'envoyeur original de la demande SEND.

Si la valeur du champ d'en-tête Failure-Report est "oui", alors le relais DOIT lancer un temporisateur pour détecter si la transmission au prochain bond échoue. Le temporisateur démarre quand le dernier octet du message a été envoyé au prochain bond. Si après 30 secondes le prochain bond n'a pas envoyé de réponse, le relais DOIT alors construire un REPORT avec un code d'état de 408 pour indiquer qu'une erreur de fin de temporisation est survenue dans l'envoi du message, et envoyer le REPORT à l'envoyeur d'origine du message.

Si la valeur du champ d'en-tête Failure-Report est "oui" ou "partiel", et si il y a un problème dans le traitement de la

demande SEND ou si une réponse d'erreur est reçue pour cette demande SEND, le relais DOIT alors répondre avec une réponse d'erreur appropriée dans un REPORT à la source originale du message.

Le relais MSRP PEUT recouper le fragment de message reçu dans la demande SEND en fragments plus petits et les transmettre au prochain bond dans des demandes SEND séparées. Il PEUT aussi combiner les fragments de message reçus avant ou après cette demande SEND, et les transmettre dans une seule demande SEND au prochain bond identifié dans le champ d'en-tête To-Path. Le relais MSRP NE DOIT PAS combiner des fragments de message provenant de demandes SEND ayant des valeurs différentes dans le champ d'en-tête Message-ID.

Le relais MSRP PEUT choisir si il refragmente le message, ou combine les fragments du message, ou envoie le message comme il est, sur la base d'une politique administrée, ou de la vitesse du réseau du dernier bond, ou tout autre mécanisme.

Si le relais MSRP a connaissance de la gamme d'octets qu'il va transmettre au prochain bond, il DEVRAIT de façon appropriée mettre à jour la valeur du champ d'en-tête Byte-Range dans la demande SEND.

Avant de transmettre la demande SEND au prochain bond, le relais MSRP DOIT inspecter le premier URI dans le champ d'en-tête To-Path. Si il indique ce relais, le relais retire l'URI du champ d'en-tête To-Path et l'insère dans le champ d'en-tête From-Path avant tout autre URI. Si il n'indique pas ce relais, il y a eu une erreur dans la transmission à un bond précédent. Dans ce cas, le relais DEVRAIT éliminer le message, et si la valeur du champ d'en-tête Failure-Report est réglée à "oui", le relais DEVRAIT générer un rapport d'échec.

6.4.2 Transmission des demandes non SEND

Un relais MSRP qui reçoit une demande autre qu'une demande SEND (incluant de nouvelles méthodes inconnues du relais) suit d'abord les règles de validation et d'autorisation pour toutes les demandes. Ensuite, le relais déplace son URI du début des champs d'en-tête To-Path au début du champ d'en-tête From-Path et transmet la demande au prochain bond. Si il a déjà une connexion avec le prochain bond, il DEVRAIT utiliser cette connexion et non former une nouvelle connexion. Si aucune connexion convenable n'existe, le relais ouvre une nouvelle connexion.

Les demandes avec une méthode inconnue sont transmises comme si elles étaient des demandes REPORT. Un nœud MSRP PEUT être configuré à bloquer les méthodes inconnues pour des raisons de sécurité.

6.4.3 Traitement des réponses

Les relais qui reçoivent une réponse vérifient d'abord que le premier URI dans le To-Path correspond à lui-même ; sinon, la réponse DEVRAIT être éliminée. De même, si le message ne peut pas être analysé, le relais DOIT éliminer la réponse. Ensuite, le relais détermine si il y a des URI supplémentaires dans le To-Path. (Pour les réponses aux demandes SEND, il n'y aura pas d'URI supplémentaires, tandis que les réponses aux demandes AUTH ont des URI supplémentaires redirigeant la réponse sur le client.)

Si la réponse correspond à une transaction existante, alors cette transaction est achevée et tous les temporisateurs qui fonctionnennent sur elle peuvent être supprimés. Si la response n'est pas de classe 200, et si la demande d'origine était une demande SEND qui avait un champ d'en-tête Failure-Report d'une valeur autre que "non", alors le relais DOIT envoyer un REPORT indiquant la nature de la défaillance. Le code de réponse reçu par le relais est utilisé pour former la ligne d'état dans le REPORT qu'envoie le relais.

Si il y a des URI supplémentaires dans le champ d'en-tête To-Path, le relais DOIT alors déplacer son URI du champ d'en-tête To-Path, insérer son URI devant tout autre URI dans le champ d'en-tête From-Path, et transmettre la réponse au prochain URI dans le champ d'en-tête To-Path. Le relais envoie la réponse sur la meilleure connexion qui correspond au prochain URI dans le champ d'en-tête To-Path. Si cette connexion est fermée, alors la réponse est éliminée en silence.

6.5 Gestion des connexions

Les relais devraient garder les connexions ouvertes aussi longtemps que possible. Si une connexion n'a pas été utilisée pendant un délai significatif (plus d'une heure) elle PEUT être close. Si le relais se trouve à bout de ressources et ne peut plus établir de nouvelles connexions, il DEVRAIT commencer à clore les connexions existantes. Il PEUT choisir de clore les connexions dont l'utilisation est la plus ancienne.

7. Syntaxe formelle

La spécification de syntaxe suivante utilise la forme Backus-Naur augmenté (ABNF) décrite dans la [RFC4234].

Cet ABNF importe toutes les définitions de l'ABNF de la RFC 4975.

```
header =/ Expires / Min-Expires / Max-Expires / Use-Path / WWW-Authenticate / Authorization / Authentication-Info
                                        ; ceci s'ajoute à la règle de la RFC 4975

mAUTH = %x41.55.54.48           ; AUTH en majuscules
method  =/ mAUTH                ; ceci s'ajoute à la règle de la RFC 4975

WWW-Authenticate    = "WWW-Authenticate:" SP "Digest" SP digest-param *("," SP digest-param)
 ; realm, nonce, et qop digest-params sont exigés ;

digest-param = ( realm / nonce / opaque / stale / algorithm / qop-options / auth-param )

realm  = "realm=" realm-value
realm-value = quoted-string

auth-param = token "=" ( token / quoted-string )

nonce = "nonce=" nonce-value
nonce-value = quoted-string
opaque = "opaque=" quoted-string
stale = "stale=" ( "true" / "false" )
algorithm = "algorithm=" ( "MD5" / token )
qop-options = "qop=" DQUOTE qop-list DQUOTE
qop-list = qop-value *( "," qop-value )
qop-value = "auth" / token

Authorization = "Authorization:" SP credentials

credentials = "Digest" SP digest-response *( "," SP digest-response)
 ; les éléments digest-response obligatoires sont : username, realm, nonce, response, cnonce, et message-qop ;

digest-response = ( username / realm / nonce / response / algorithm / cnonce / opaque / message-qop /
                    nonce-count / auth-param )

username = "username=" username-value
username-value = quoted-string
message-qop = "qop=" qop-value
cnonce = "cnonce=" cnonce-value
cnonce-value = nonce-value
nonce-count = "nc=" nc-value
nc-value = 8LHEX
response = "response=" request-digest
request-digest = DQUOTE 32LHEX DQUOTE
LHEX  = DIGIT / %x61-66         ; a-f en minuscules

Authentication-Info =  "Authentication-Info:" SP ainfo *("," ainfo)
ainfo = nextnonce / message-qop / response-auth / cnonce / nonce-count
nextnonce = "nextnonce=" nonce-value
response-auth = "rspauth=" response-digest
response-digest = DQUOTE *LHEX DQUOTE

Expires = "Expires:" SP 1*DIGIT
Min-Expires = "Min-Expires:" SP 1*DIGIT
Max-Expires = "Max-Expires:" SP 1*DIGIT
```

Use-Path = "Use-Path:" SP MSRP-URI *(SP MSRP-URI)

8. Découverte de relais MSRP

Quand il résout un URI MSRP qui contient un numéro d'accès explicite, un nœud MSRP suit les règles de la Section 6 de la spécification MSRP de base. Les URI MSRP échangés dans SDP et dans les champs d'en-têtes To-Path et From-Path dans les demandes non AUTH DOIVENT avoir un numéro d'accès explicite. (Le seul message dans la présente spécification qui puisse avoir un URI MSRP sans numéro d'accès explicite est dans le champ d'en-tête To-Path dans une demande AUTH.) De même, si le composant d'autorité d'un URI msrps: contient une adresse IPv4 ou une référence IPv6, un numéro d'accès DOIT être présent.

Les règles suivantes permettent aux clients MSRP de découvrir les relais MSRP plus facilement dans les demandes AUTH. Si le composant d'autorité contient un nom de domaine et si un numéro d'accès explicite est fourni, on tente de chercher un enregistrement d'adresse valide (A ou AAAA) pour le nom de domaine. On se connecte en utilisant TLS sur le transport par défaut (TCP) avec le numéro d'accès fourni.

Si un nom de domaine est fourni sans numéro d'accès, effectuer une recherche de SRV DNS [RFC2782] pour le service '_msrps' et le transport '_tcp' au nom de domaine, et suivre l'algorithme de choix d'enregistrement de service (SRV, *Service Record*) défini dans la présente spécification pour choisir l'entrée. (Un service '_msrp' n'est pas défini, car les demandes AUTH ne sont envoyées que sur TLS.) Si aucun SRV n'est trouvé, essayer une recherche d'adresse (A ou AAAA) pour le nom de domaine. Se connecter en utilisant TLS sur le transport par défaut (TCP) avec le numéro d'accès par défaut (2855). Noter que les demandes AUTH DOIVENT seulement être envoyées sur un canal protégé par TLS. Une recherche de SRV dans le domaine exemple.com pourrait retourner :

```
;; dans exemple.com.       Pri Wght Accès  Cible
  _msrps._tcp   IN SRV   0    1     9000   serveur1.exemple.com.
  _msrps._tcp   IN SRV   0    2     9000   serveur2.exemple.com.
```

Si on met en œuvre un réservoir de relais, il est RECOMMANDÉ que chaque membre du réservoir de relais ait une entrée de SRV. Si un membre du réservoir a plusieurs adresses IP (par exemple, une adresse IPv4 et une adresse IPv6) chacune de ces adresses DEVRAIT être enregistrée dans le DNS comme un enregistrement A ou AAAA séparé correspondant à une seule cible.

9. Considérations sur la sécurité

Cette section décrit d'abord les mécanismes de sécurité disponibles utilisables dans MSRP. Le modèle de menaces est présenté ensuite. Finalement, on fait la liste des exigences de mise en œuvre relatives à la sécurité.

9.1 Utilisation de l'authentification HTTP

Les demandes AUTH DOIVENT être authentifiées. Le mécanisme d'authentification décrit dans la présente spécification utilise l'authentification par résumé HTTP. L'authentification par résumé HTTP est effectuée comme décrit dans la [RFC2617], avec les restrictions et modifications suivantes :

- Les clients NE DOIVENT PAS tenter d'utiliser l'authentification de base, et les relais NE DOIVENT PAS demander ou accepter l'authentification de base.

- L'utilisation d'une valeur de qop de auth-int n'a pas de sens pour MSRP. La protection de l'intégrité est fournie par l'utilisation de TLS. Par conséquent, les relais MSRP NE DOIVENT PAS indiquer un qop de auth-int dans un défi.

- L'interaction entre l'algorithme MD5-sess et le mécanisme nextnonce est sous spéifiée dans la [RFC2617] ; par conséquent, les relais MSRP NE DOIVENT PAS envoyer de défis indiquant l'algorithme MD5-sess.

- Les clients DEVRAIENT considérer l'espace de protection au sein d'un domaine comme ayant la portée de la portion d'autorité de l'URI, sans considération du contenu de la portion chemin de l'URI. En conséquence, les relais NE DEVRAIENT PAS envoyer de paramètre "domain" dans l'en-tête "WWW-Authenticate", et les clients DOIVENT l'ignorer si il est présent.

- Les clients et relais DOIVENT inclure un paramètre qop dans tous les en-têtes "WWW-Authenticate" et "Authorization". Noter que la valeur du paramètre qop dans un en-tête "WWW-Authenticate" est entre guillemets, mais la valeur du paramètre qop dans un en-tête "Authorization" ou "Authentication-Info" n'est pas entre guillemets.

- Les clients DOIVENT envoyer les paramètres cnonce et nonce-count dans tous les en-têtes "Authorization".

- L'URI de demande à utiliser pour calculer H(A2) est l'URI le plus à droite dans le champ d'en-tête To-Path.

- Les relais DOIVENT inclure les paramètres rspauth, cnonce, nc, et qop dans un en-tête "Authentication-Info" pour toutes les réponses "200 OK" à une demande AUTH.

Noter que le BNF dans la RFC 2617 a un certain nombre d'erreurs. En particulier, la valeur du paramètre uri DOIT être entre guillemets ; de plus, les paramètres dans l'en-tête Authentication-Info DOIVENT être séparés par des virgules. Le BNF dans le présent document est correct, comme le sont les exemples dans la [RFC2617].

L'utilisation des paramètres nextnonce et nc est prise en charge comme décrit dans la [RFC2617], qui fournit des lignes directrices sur la façon et le moment où ils devraient être utilisés. Une lègère modification aux lignes directrices de la RFC 2617 est que les mises en œuvre de relais devraient noter que les demandes AUTH ne peuvent pas être traitées en parallèle ; par conséquent, il n'y a pas d'impact négatif sur le débit quand les relais utilisent le mécanisme nextnonce.

Voir au paragraphe 5.1 des informations sur les procédures d'authentification du client.

9.2 Utilisation de TLS

TLS est utilisé pour authentifier les relais auprès des envoyeurs et pour assurer l'intégrité et la confidentialité des en-têtes transportés. Les clients et relais MSRP DOIVENT mettre en œuvre TLS. Les clients DOIVENT envoyer les informations de hello étendu de ClientExtendedHello TLS pour l'indication du nom du serveur comme décrit dans la [RFC4366]. Une suite de chiffrement TLS de TLS_RSA_WITH_AES_128_CBC_SHA [RFC3268] DOIT être prise en charge (d'autres suites de chiffrement PEUVENT également être prises en charge). Un relais DOIT agir comme serveur TLS et présenter un certificat avec son identité dans le SubjectAltName en utilisant le type de choix de dnsName. Les connexions de relais à relais DOIVENT utiliser TLS avec authentification mutuelle. Les communications de client à relais DOIVENT utiliser TLS pour les demandes et les réponses AUTH.

Le SubjectAltName du certificat reçu d'un relais DOIT correspondre à la partie Nom d'hôte de l'URI, et le certificat DOIT être valide selon la [RFC3280], y compris avoir une date valide et être signé par une autorité de certification acceptable. Après validation que c'est bien le cas, l'appareil qui a initié la connexion TLS peut supposer qu'il s'est connecté au bon relais.

Le présent document ne définit pas de procédures pour l'utilisation de l'authentification mutuelle entre un client MSRP et un relais MSRP. L'authentification des clients est assurée par la méthode AUTH selon les procédures décrites aux paragraphes 5.1 et 6.3. D'autres spécifications peuvent définir l'utilisation de l'authentification mutuelle TLS aux fins d'authentification des utilisateurs associés aux clients MSRP. Sauf s'ils opèrent dans le cadre de ces autres spécifications, les clients MSRP DEVRAIENT présenter une liste de certificats vide (si une liste est demandée par le relais MSRP) et les relais MSRP DEVRAIENT ignorer tout certificat présenté par le client.

Ce comportement est défini spécifiquement pour permettre la compatibilité future avec les spécifications qui définissent l'utilisation de TLS pour l'authentification du client.

Note : Quand des relais sont impliqués dans une session, TCP sans TLS n'est utilisé que quand un utilisateur qui n'utilise pas de relais se connecte directement au relais d'un utilisateur qui utilise des relais. Dans ce cas, le client n'a pas de moyen d'authentifier le relais autrement qu'en utilisant les URI qui forment un secret partagé de la même façon que ces URI sont utilisés quand aucun relais n'est impliqué.

9.3 Modèle de menaces

Ce paragraphe examine le modèle de menace et le mécanisme général qui doit être mis en place pour sécuriser le protocole. Le paragraphe suivant décrit les détails de comment le mécanisme du protocole satisfait les exigences générales.

MSRP permet à deux clients d'homologue à homologue d'échanger des messages. Chaque homologue peut sélectionner un ensemble de relais pour effectuer certaines opérations de politique pour lui. Cet ensemble combiné de relais est appelé l'ensemble de chemins. Un canal extérieur à MSRP doit toujours exister, tel que le provisionnement hors bande ou un protocole de rendez-vous explicite comme SIP, qui peut négocier en toute sécurité l'établissement de la session MSRP et communiquer l'ensemble de chemins aux deux clients. Un client peut faire confiance à un relais pour certains types de décisions d'acheminement et de politique, mais il peut ou non faire confiance au relais pour tout le contenu de la session. Par exemple, un relais de confiance pour rechercher des virus devra probablement être autorisé à voir tout le contenu de la session. Un relais qui aide à traiter la traversée du traducteur d'adresses réseau (NAT, *Network Address Translator*) du fournisseur d'accès Internet ne sera probablement pas de confiance pour le contenu de la session, mais sera de confiance pour transmettre correctement les messages.

Les clients font implicitement confiance aux relais par lesquels ils envoient et reçoivent des messages pour respecter l'acheminement indiqué dans ces messages, dans les limites des contraintes du protocole MSRP. Les clients doivent également avoir confiance que les relais qu'ils utilisent n'insèrent pas de nouveaux messages en leur nom et ne modifient pas les messages envoyés aux ou par les clients. Il convient de noter que certains relais sont en mesure d'amener un client à mal acheminer un message en modifiant de manière malveillante un Use-Path *(chemin d'utilisation)* renvoyé par un relais situé plus bas dans la chaîne. Cependant, il ne s'agit pas d'une menace supplémentaire pour la sécurité, car ces mêmes relais peuvent également décider d'abord de mal acheminer un message. Si on fait confiance au relais pour acheminer les messages, il est raisonnable de lui faire confiance pour ne pas modifier la valeur du champ d'en-tête Use-Path. Si le relais ne peut pas être de confiance pour l'acheminement des messages, alors il ne peut pas être utilisé.

Dans certaines circonstances, les relais doivent faire confiance à d'autres relais pour ne pas modifier les informations entre eux et le client qu'ils représentent. Par exemple, si un client passe par le relais A pour atteindre le relais B, et si le relais B enregistre les messages envoyés par le client, le relais B peut être tenu d'authentifier que les messages qu'il enregistre proviennent du client et n'ont pas été modifiés ou falsifiés par le relais A. Cela peut se faire en demandant au client de signer le message.

Les clients doivent être capables d'authentifier que le relais avec lequel ils communiquent est celui en qui ils ont confiance. De même, les relais doivent être capables d'authentifier que le client est bien celui à qui ils sont autorisés à envoyer des informations. Les clients doivent avoir la possibilité de s'assurer que les informations entre le relais et le client sont protégées en intégrité et confidentielles pour les éléments autres que le relais et les clients. Pour simplifier le nombre d'options, le trafic entre relais est toujours protégé en intégrité et chiffré, que le client le demande ou non. Les clients n'ont aucun moyen d'indiquer aux relais la puissance des mécanismes cryptographiques à utiliser entre relais, si ce n'est que les clients choisissent des relais qui sont administrés de manière à exiger un niveau de sécurité adéquat.

Le système doit également empêcher les messages d'être dirigés vers des relais qui ne sont pas censés les voir. Pour éviter que les relais soient utilisés dans des attaques par déni de service (DoS), les relais ne transmettent jamais de messages à moins d'avoir une relation de confiance avec le client qui envoie ou qui reçoit le message ; en outre, ils ne transmettent un message que s'il provient du client avec lequel ils ont une relation de confiance ou s'il est destiné à ce client. Si un relais a une relation de confiance avec le client qui est la destination du message, il ne devrait pas envoyer le message ailleurs qu'au client qui est la destination.

Quelques termes utilisés dans cette discussion : SClient est le client qui envoie un message et RClient est le client qui reçoit un message. SRelay est un relais auquel l'envoyeur fait confiance et RRelay est un relais auquel le receveur fait confiance. Le message ira de SClient à SRelay1 à SRelay2 à RRelay2 à RRelay1 à RClient.

9.4 Mécanisme de sécurité

La confidentialité et la protection de la vie privée à partir d'éléments qui ne sont pas dans l'ensemble de chemins sont fournies en utilisant TLS sur tous les transports. Les relais utilisent toujours TLS. Un client peut utiliser TCP non protégé pour MSRP d'homologue à homologue, mais chaque fois qu'un client communique avec son relais, il DOIT utiliser TLS.

Le relais s'authentifie auprès des clients en utilisant TLS (mais n'a pas à faire un TLS mutuel). De plus, l'utilisation du paramètre rspauth dans l'en-tête Authentication-Info fournit une authentification limitée des relais auxquels le client n'est pas directement connecté. Les clients s'authentifient auprès du relais en utilisant l'authentification par résumé HTTP. Les relais s'authentifient les uns les autres en utilisant l'authentification mutuelle TLS.

En utilisant le chiffrement des extensions sécurisées multi objets de messagerie Internet (S/MIME, *Secure/Multipurpose Internet Mail Extensions*) [RFC3851], les clients peuvent protéger leurs contenus réels de message afin que les relais ne puissent pas voir les contenus. La signature de bout en bout est aussi possible avec S/MIME.

La partie complexe est de s'assurer que les relais ne peuvent pas recevoir pour instruction d'envoyer des messages à un endroit où ils ne devraient pas les envoyer. Pour cela, le client s'authentifie auprès du relais et le relais retourne un jeton. Les messages qui contiennent ce jeton peuvent être relayés si ils viennent du client qui a obtenu le jeton ou si ils ont été transmis au client qui a obtenu le jeton. Les jetons sont les URI que le relais place dans le champ d'en-tête Use-Path. Les jetons contiennent du materiel aléatoire (défini au paragraphe 6.3) afin qu'ils ne puissent pas être devinés par des attaquants. Les jetons doivent être protégés afin qu'ils ne soient jamais vus que par les éléments de l'ensemble d'acheminement ou d'autres éléments qu'au moins une des parties tient pour être de confiance. Si un tiers découvre le jeton que RRelay2 utilise pour envoyer des messages au RClient, alors ce tiers peut envoyer autant de messages qu'il veut au RRelay2 et il va les transmettre au RClient. Le tiers ne peut pas causer leur transmission ailleurs qu'au RClient, éliminant les problèmes de relais ouvert. SRelay1 ne va transmette le message que si il contient un jeton valide.

Quand le SClient va pour obtenir un jeton de SRelay2, cette demande est relayée par SRelay1. SRelay2 authentifie que c'est bien le SClient qui demande le jeton, mais il génère un jeton qui n'est valable que pour transmettre des messages à destination ou en provenance de SRelay1. SRelay2 sait qu'il est connecté à SRelay1 grâce au protocole TLS mutuel.

Les jetons sont portés dans la portion ressource des URI MSRP. La durée de validité des jetons est négociée à l'aide de l'en-tête Expire de la demande AUTH. Les clients doivent renégocier les jetons en utilisant un nouvel échange offre/réponse [RFC3264] (par exemple, un re-invite SIP) avant que les jetons expirent.

Noter que ce système repose sur les relais en tant que nœuds de confiance, agissant au nom des utilisateurs authentifiés auprès d'eux. Il n'existe aucun mécanisme de sécurité pour empêcher les relais sur le chemin d'insérer de faux messages, de manipuler le contenu des messages, d'envoyer des messages dans une session à une partie autre que celle spécifiée par l'envoyeur, ou de les copier à un tiers. Cependant, le lien univoque entre les identifiants de session et les sessions permet d'atténuer les dommages qui peuvent être causés par des relais malhonnêtes en limitant les destinations vers lesquelles des messages falsifiés ou modifiés peuvent être envoyés aux deux parties impliquées dans la session, et ce uniquement pendant la durée de la session. En outre, l'utilisation du chiffrement S/MIME peut être employée pour limiter l'utilité de la redirection des messages. Enfin, les clients peuvent utiliser des signatures S/MIME pour garantir l'authenticité des messages qu'ils envoient, ce qui permet, dans certaines circonstances, de détecter la manipulation des relais ou la falsification des messages.

Les clients ne sont pas les seuls acteurs du réseau à devoir faire confiance aux relais pour agir de manière non malveillante. Si un relais n'a pas de connexion TLS directe avec le client au nom duquel il agit (c'est-à-dire qu'il y a un ou plusieurs relais intermédiaires) il est à la merci de ces relais intermédiaires pour transmettre avec précision les messages envoyés au client et en provenance de celui-ci. Si une garantie plus forte de l'origine authentique d'un message est nécessaire (par exemple, le relais procède à l'enregistrement des messages au titre d'une obligation légale) les utilisateurs de ce relais peuvent alors recevoir l'instruction de leurs administrateurs d'utiliser des signatures S/MIME détachées pour tous les messages envoyés par leur client. Le relais peut appliquer cette politique en renvoyant une réponse 415 à toute demande SEND utilisant un type MIME de niveau supérieur autre que "multipart/signed". Un tel relais peut choisir de prendre des décisions de politique (telles que l'interruption des sessions et/ou la suspension de l'autorisation de l'utilisateur) si ces signatures ne correspondent pas au contenu du message.

10. Considérations relatives à l'IANA

10.1. New MSRP Method

La présente spécification définit une nouvelle méthode MSRP, à ajouter au sous registre des méthodes sous le registre des paramètres MSRP : AUTH. Voir au paragraphe 5.1 les détails de la méthode AUTH.

10.2 Nouveaux champs d'en-tête MSRP

La présente spécification définit plusieurs nouveaux champs d'en-tête MSRP, à ajouter au sous registre des champs d'en-tête sous le registre des paramètres MSRP :

- Expires
- Min-Expires
- Max-Expires
- Use-Path
- WWW-Authenticate
- Authorization
- Authentication-Info

10.3 Nouveaux codes de réponse MSRP

La présente spécification définit un nouveau code d'état MSRP, à ajouter au sous registre des codes d'état sous le registre des paramètres MSRP :

La réponse 401 indique qu'une demande AUTH ne contenait pas d'accréditifs, un valeur de nom occasionnel périmée, ou des accréditifs invalides. La réponse inclut un champ d'en-tête "WWW-Authenticate" contenant un défi (parmi d'autres champs) ; voir les détails au paragraphe 9.1. La phrase de réponse par défaut pour cette réponse est "non autorisé".

11. Exemple de SDP avec plusieurs bonds

La présente Section montre un exemple de SDP qui pourrait se produire dans un message SIP pour établir une session MSRP entre Alice et Bob où Bob utilise un relais. Alice fait une offre avec un chemin pour Alice.

```
c=IN IP4 a.exemple.com
m=message 1234 TCP/MSRP *
a=accept-types: message/cpim text/plain text/html
a=path:msrp://a.exemple.com:1234/agic456;tcp
```

Dans cette offre, Alice souhaite recevoir des messages MSRP à a.exemple.com. Elle veut utiliser TCP comme transport pour la session MSRP. Elle peut accepter des corps de message message/cpim, text/plain, et text/html dans les demandes SEND. Elle n'a pas besoin de relais pour établir la session MSRP.

À cette offre, la réponse de Bob pourrait ressembler à :

```
c=IN IP4 bob.exemple.com
m=message 1234 TCP/TLS/MSRP *
a=accept-types: message/cpim text/plain
a=path:msrps://relay.exemple.com:9000/hjdhfha;tcp \ msrps://bob.exemple.com:1234/fuige;tcp
```

Ici, Bob souhaite recevoir les messages MSRP à bob.exemple.com. Il ne peut accepter que les corps de message message/cpim et text/plain dans les demandes SEND et a rejeté le contenu text/html offert par Alice. Il souhaite utiliser le relais appelé relay.exemple.com pour la session MSRP.

12. Remerciements

Tous nos remerciements à Avshalom Houri, Hisham Khartabil, Robert Sparks, Miguel Garcia, Hans Persson, et Orit Levin, qui ont fourni des notes de relecture détaillées et du texte utile. Merci aux membres suivants du groupe de travail SIMPLE pour les discussions animées sur le mode session : Chris Boulton, Ben Campbell, Juhee Garg, Paul Kyzivat, Allison Mankin, Aki Niemi, Pekka Pessi, Jon Peterson, Brian Rosen, Jonathan Rosenberg, et Dean Willis.

13. Références

13.1 Références normatives

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

[RFC2617] J. Franks et autres, "Authentification HTTP : Authentification d'accès de base et par résumé", juin 1999. *(DS.)*

[RFC2782] A. Gulbrandsen, P. Vixie et L. Esibov, "Enregistrement de ressource DNS pour la spécification de la localisation des services (DNS SRV)", février 2000.

[RFC3261] J. Rosenberg et autres, "SIP : Protocole d'initialisation de session", juin 2002. (*Mise à jour par* 3265, 3853,

4320, 4916, 5393, 6665, 8217, 8760)

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

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

[RFC3851] B. Ramsdell, "Spécification du message d'extensions de messagerie Internet multi-objets/sécurisé (S/MIME) version 3.1", juillet 2004. *(Obsolète, voir* RFC5751)

[RFC4234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", octobre 2005. *(Remplace* RFC2234, *remplacée par* RFC5234)

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

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

[RFC4566] M. Handley, V. Jacobson et C. Perkins, "SDP : Protocole de description de session", juillet 2006. *(P.S. ; remplacée par RFC8866)*

[RFC4975] B. Campbell et autres, "Protocole de relais de session de message (MSRP)", septembre 2007. *(P.S ; MàJ par* RFC7977, RFC8873)

13.2 Références pour information

[RFC2045] N. Freed et N. Borenstein, "Extensions de messagerie Internet multi-objets (MIME) Partie 1 : Format des corps de message Internet", novembre 1996. *(D. S., MàJ par* 2184, 2231, 5335.)

[RFC2046] N. Freed et N. Borenstein, "Extensions de messagerie Internet multi-objets (MIME) Partie 2 : Types de support", novembre 1996. *(D. S., MàJ par* 2646, 3798, 5147, 6657, 8098)

[RFC3264] J. Rosenberg et H. Schulzrinne, "Modèle d'offre/réponse avec le protocole de description de session (SDP)", juin 2002. *(P.S. ; MàJ par* RFC8843, 9143)

Appendice A. Considérations de mise en œuvre

Ce texte n'est pas nécessaire pour mettre en œuvre MSRP d'une manière interopérable, mais il reste utile comme discussion sur la mise en œuvre pour la communauté. Il est purement un détai de mise en œuvre.

Note : L'idée a été proposée qu'un relais renvoie un URI de base que le client puisse utiliser pour construire d'autres URI, mais cela permet à des tiers qui ont eu une session avec le client de connaître les URI que le relais utilisera pour la transmission après la fin de la session avec le tiers. En effet, cela révèle les URI secrets aux tiers, ce qui compromet la sécurité de la solution, c'est pourquoi cette approche n'est pas utilisée.

Une solution de remplacement de cette approche consiste pour le relais à renvoyer un URI divisé en une portion indice et une portion secrète. Le client peut chiffrer son identifiant et ses propres données opaques avec la partie secrète, et les enchaîner avec la partie indice pour créer une pluralité d'URI valides. Quand le relais reçoit un de ces URI, il pourrait utiliser l'indice pour rechercher le secret approprié, déchiffrer la portion client et vérifier qu'elle contient l'identifiant du client. Le relais peut alors transmettre la demande. Le client n'a pas besoin d'envoyer une demande AUTH pour chaque URI qu'il utilise. Il s'agit là d'un détail de mise en œuvre qui n'entre pas dans le domaine d'application du présent document.

Il est possible de mettre en œuvre les exigences de transmission dans un réservoir sans que le relais ne sauvegarde aucun état. Une mise en œuvre possible qu'un relais pourrait utiliser est décrite dans le reste de cette section. Lorsqu'un relais

démarre, il pourrait choisir un mot de passe (K) et une valeur d'initialisation (IV) cryptographiquement aléatoires de 128 bits. Si le relais était en fait un réservoir de serveurs avec le même nom DNS, toutes les machines du réservoir devraient partager le même K. Quand une demande AUTH est reçue, le relais forme une chaîne qui contient l'heure d'expiration de l'URI, une indication de si le bond précédent était mutuellement authentifié TLS ou non, et s'il l'était, le nom du bond précédent, et s'il ne l'était pas, l'identifiant de la connexion qui a reçu la demande AUTH. Cette chaîne est complétée par un octet de valeur 0x80, puis par zéro ou plusieurs octets de valeur 0x00 jusqu'à ce que la longueur de la chaîne soit un multiple de 16 octets. Une nouvelle IV aléatoire est sélectionnée (elle doit changer parce qu'elle constitue le sel) et la chaîne complétée est chiffrée à l'aide d'AES-CBC avec une clé de K. L'IV, les données chiffrées et un SPI (indice de paramètre de sécurité) qui change chaque fois que K change sont codés en base 64 et forment la partie "ressource" de l'URI de la demande. Le SPI permet à la clé d'être modifiée et au système de savoir quel K devrait être utilisé. Plus tard, lorsque le relais recevra cet URI, il pourra le déchiffrer, vérifier que l'heure actuelle est antérieure à l'heure d'expiration et vérifier que le message provient de la connexion ou de l'emplacement spécifié dans l'URI ou qu'il y est destiné. La protection de l'intégrité n'est pas nécessaire car il est extrêmement improbable que des données aléatoires déchiffrées aboutissent à un emplacement valide qui soit le même que celui auquel le message était acheminé ou d'où il provenait. Quand on met en œuvre quelque chose comme cela, on devrait être attentif à ne pas utiliser un schéma comme EBE qui permettrait que des portions de jetons chiffrés soient coupées/collées dans d'autres URI.

Adresse des auteurs

Cullen Jennings
Cisco Systems, Inc.
170 West Tasman Dr.
MS: SJC-21/2
San Jose, CA 95134
USA
mél : fluffy@cisco.com

Rohan Mahy
Plantronics
345 Encincal Street
Santa Cruz, CA 95060
USA
mél : rohan@ekabal.com

Adam Roach
Estacado Systems
17210 Campbell Rd.
Suite 250
Dallas, TX 75252
USA
mél : adam@estacado.net

Déclaration complète de droits de reproduction

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