

Sécurité des Transmissions de Données Réseaux Privés Virtuels

Brie Romain Colpart Grégory Dubois Sébastien

9 janvier 2004

Le sujet :

Vue d'ensemble :

L'utilisation de l'Internet comme réseau d'interconnexion de sociétés s'est aujourd'hui largement répandue. Cette démocratisation est non seulement due à l'émergence de technologies spécifiques à la transmissions de données entre réseaux privés (VPN), mais également aux garanties en termes de sécurité des transferts qui ont été données.

But de l'étude :

Il s'agira tout d'abord, par une recherche bibliographique, de rappeler les problématiques techniques spécifiques aux réseaux privés virtuels. Dans un second temps, il s'agira de mettre en oeuvre un banc de simulation d'un réseau privé virtuel, à base d'IPSec, entre deux systèmes hétérogènes.

Préambule

Ce document est destiné à donner une vue d'ensemble des "Réseaux Privés Virtuels" (Virtual Private Networks, ou VPN), ainsi que d'IPSec, qui est le cadre technique actuel le plus répandu pour leur mise en oeuvre pratique. Ce document n'a pas la prétention de décrire de façon exhaustive les VPN ainsi que leurs protocoles. On se référera aux RFC pour des considérations plus techniques (voir bibliographie).

La version originale de ce document se trouve à l'adresse
<http://www.gcolpart.com/~reg/vpn/>

Vous avez le droit de copier, distribuer et/ou modifier ce document selon les termes de la GNU Free Documentation License, version 1.2 ou n'importe quelle version ultérieure, telle que publiée par la Free Software Foundation. Le texte de la licence se trouve à l'adresse <http://www.gnu.org/copyleft/fdl.html>

Table des matières

1	Les réseaux privés virtuels (VPN)	5
1.1	Introduction au VPN	5
1.2	Définition d'un VPN	6
1.3	Exigences des VPN et leur évolution	7
1.4	Exemples d'utilisation pratique	8
2	IPSec	9
2.1	ISAKMP : Internet Security Association and Key Management Protocol	10
2.1.1	Rôle	10
2.1.2	Principe de fonctionnement	10
2.1.3	La construction par blocs d'ISAKMP	10
2.1.4	Les échanges de ISAKMP :	12
2.2	IKE	13
2.2.1	Main Mode Exchange	13
2.2.2	Aggressive Mode Exchange	14
2.2.3	Quick Mode Exchange	14
2.3	Protocoles AH et ESP	14
2.3.1	Authentication Header	15
2.3.2	Encapsulating Security Payload	15
3	Applications	18
3.1	Installation	18
3.2	Configuration	19
3.2.1	Configuration manuelle des clés	19
3.2.2	Configuration avec un secret partagé	20
3.3	Mise en oeuvre	21
3.3.1	Configuration manuelle des clés	21
3.3.2	Configuration avec un secret partagé	24
3.4	Commentaires généraux	27
4	Conclusion	29

5	Annexe	30
5.1	Annexe 1	30
5.2	Annexe 2	32
6	Bibliographie	37

1 Les réseaux privés virtuels (VPN)

1.1 Introduction au VPN

Dès les débuts de l'informatique la notion de réseau a été présente. Les réseaux locaux sont rapidement devenus de plus en plus nombreux. Une des problématiques a été l'utilisation des ressources d'un réseau local à distance. Les premières solutions viables se sont appuyées sur des liaisons téléphoniques traditionnelles (RTC), puis sur des liaisons du type Numéris, ou encore sur des lignes dédiées. Il reste qu'au final ces solutions présentent un certain nombre d'inconvénients :

Pour les liaisons par lignes spécialisées

- coût très élevé

Pour les liaisons par modems RTC

- lenteur de la connexion (certains services nécessitent un minimum de bande passante, le nombre de connexions simultanées possibles est faible, etc.)

- mise en oeuvre plus lourde (occupation d'une ligne téléphonique, etc.)

Parallèlement à ce besoin d'accéder au réseau interne des entreprises depuis tout lieu sur le globe, une nouvelle technologie de communication s'est développée : l'Internet. En sa qualité de réseaux de réseaux (interconnexion), l'Internet a fait naître l'idée de son utilisation en tant que support de communication, un peu à l'égal d'une connexion téléphonique traditionnelle. La démocratisation des connexions haut débit à Internet a conforté cette situation.

L'utilisation d'Internet pour accéder à distance aux réseaux privés des entreprises s'est donc imposée progressivement. Un des avantages évidents est une réduction substantielle des coûts ¹.

Typiquement, l'un des cas de figure est la création d'un VPN entre deux réseaux locaux reliés à Internet (voir figure 1).

En pratique, la notion de VPN est souvent rattachée à la notion de sécurité du fait de son utilisation par les entreprises. Les données des entreprises devant rester confidentielles, l'utilisation d'un VPN est souvent conjuguée avec un chiffrement des données. En effet, avec un VPN, les données transitent par Internet, réseau public, et sont donc théoriquement accessibles à n'importe qui. Le chiffrement rend donc illisible les données qui circulent. Mais il faut préciser que le chiffrement n'est pas le but principal d'un VPN.

Remarquons qu'une problématique récurrente est la bande passante qui limite les technologies VPN.

¹Connecter deux sites distants de quelques centaines de kilomètres avec une LS (Ligne spécialisée) à 2Mo coûte beaucoup plus cher que la souscription de deux abonnements au haut débit

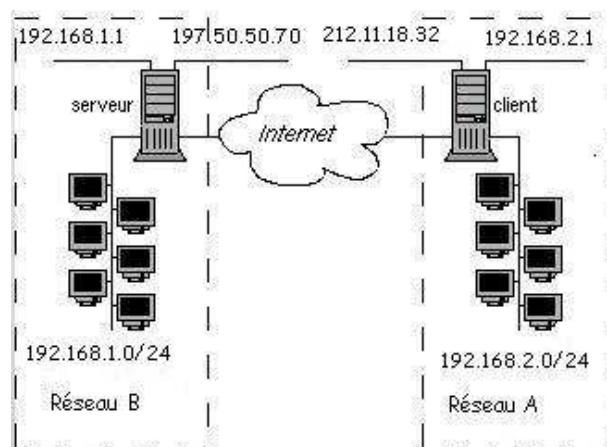


FIG. 1 – Le VPN permet de relier deux réseaux locaux

1.2 Définition d'un VPN

L'acronyme VPN signifie Virtual Private Network, c'est-à-dire Réseau Privé Virtuel.

Lorsqu'on aborde pour la première fois le sujet des réseaux privés virtuels, on s'aperçoit que le nombre de définitions d'un VPN est très élevé.

Une première définition simple et approximative serait :

Un VPN est un réseau privé construit sur l'infrastructure d'un réseau public, comme l'est Internet.

Des réseaux sont dit privés quand on peut distinguer deux types de connexions : les connexions locales et les connexions externes. Les connexions externes ne pouvant accéder qu'à des services très restreints (parfois aucun!) et définis par l'administrateur.

Le terme "Virtual Private" signifie donc que l'on veut permettre l'accès à toutes les ressources locales par des connexions externes. Il s'agit donc de créer un réseau ayant les caractéristiques d'un réseau privé à partir d'une structure qui ne l'est pas elle-même.

Un VPN peut être créé entre deux systèmes, entre deux organisations, entre plusieurs systèmes et une organisation ou entre plusieurs organisations répandues sur Internet, soit encore entre des applications individuelles ou une combinaison de ces possibilités.

Dès lors on aboutit à la définition suivante caractérisant un VPN :

Un VPN est un environnement de communication dans lequel l'accès est contrôlé, afin de permettre des connexions entre une communauté d'intérêt seulement. Son objectif est de fournir aux utilisateurs les conditions d'exploitation, d'utilisation et de sécurité à travers un réseau public identiques à celles disponibles sur un réseau privé.

1.3 Exigences des VPN et leur évolution

Une solution VPN doit permettre aux utilisateurs distants de se connecter aux ressources du réseau local (solution Host-to-LAN). La solution doit aussi permettre à des réseaux locaux distants de s'interconnecter afin de partager les ressources et les informations (solution LAN-to-LAN).

Du point de vue de l'utilisateur aucune différence notable ne doit apparaître dans son utilisation des ressources du système d'information.

Il doit donc y avoir une interface utilisateur suffisamment instinctive. L'idéal est l'implémentation transparente, à l'image des connexions réseau standard. De plus la solution peut assurer la confidentialité et l'intégrité des données à travers Internet pour répondre aux besoins des entreprises. Le même concept peut être appliqué à des données sensibles traversant le réseau d'entreprise. En conséquence une solution VPN peut offrir les fonctions suivantes :

- *Authenticité (Authentification de l'utilisateur et Intégrité des données)* : la solution doit vérifier l'identité de l'utilisateur et limiter l'accès VPN seulement aux utilisateurs autorisés. Il doit de même pouvoir enregistrer les accès, afin de permettre ensuite de déterminer qui s'est connecté et quand il s'est connecté (trois types principaux : les certificats digitaux, phrases challenge ou radius. voir annexe 1)
- *Chiffrement des données* : les données circulant dans le réseau public (Internet) doivent être illisibles aux clients non autorisés. Le problème réside dans l'échange de la valeur de la clé entre les 2 entités. On le résout grâce au protocole de Diffie-Hellman. La faiblesse de ce type d'échange réside dans la validité de la clé publique. Il s'agit de contrôler l'origine de l'entité qui envoie la clé publique, il faut l'authentifier (certificats). Il est important de noter qu'un chiffrement basé sur une solution matérielle se révèle beaucoup plus rapide que son équivalente logicielle.
- *Administration des clés* : la solution doit générer et mettre à jour les clés pour l'encryption des données pour le client et le serveur.

Il est évident qu'il faudra faire un compromis - suivant les exigences spécifiques d'un cas réel de mise en place - entre la sécurité et la rapidité (taux de transfert) liée au coût.

1.4 Exemples d'utilisation pratique

– Télé-travail

Raccordement de télé-travailleurs ou travailleurs mobiles. Ceux-ci se raccordent aux ressources de l'entreprise par modem ou tout autre moyen de connexion. Cette connexion à distance est utile pour de nombreuses catégories de personnel (commerciaux, développeurs, direction, ...) et pour de nombreuses raisons pratiques (travailler chez soi ou en déplacement).

– Connexion de sites distants/ Externalisation / WAN

Interconnexion de succursales. Des sites distants d'une même entreprise qui partagent les mêmes ressources sans avoir recours à des lignes spécialisées. Ceci peut aussi permettre, mis à part les avantages financiers comparés aux LS, une certaine indépendance vis à vis des opérateurs de télécommunication.

– Transport de la voix

Les possibilités de transport de la voix sur un réseau IP couplées aux deux utilisations précédentes des VPN ouvrent des perspectives tentantes. Avec un simple logiciel de phonie en main la solution VPN de transport de la voix fournit la gratuité des communications inter-site. Historiquement le téléphone rouge, liaison inviolable entre Moscou et Washington, a symbolisé les conversations distantes sécurisées. Le haut débit et la technologie VPN permettent cela pour de faibles coûts même si les bandes passantes peuvent s'avérer encore un peu justes pour une sécurité optimale sur plusieurs appels simultanés.

2 IPSec

À l'heure actuelle, plusieurs solutions VPN existent et sont utilisées sur Internet : IPSec est celle que l'on retrouve le plus souvent, et de nombreux facteurs indiquent que cette situation tend à se généraliser (supporté nativement par de nombreux systèmes d'exploitation, protocole ouvert, etc.). Notre étude se limitera donc à IPSec et aux mécanismes qu'il engendre ; avant d'entrer dans les détails techniques, revenons au principes de base d'IPSec.

IPSec se présente sous la forme d'un ensemble de mécanismes permettant d'initier, au niveau réseau, des connexions entre systèmes distants. Le schéma suivant rappelle les principes de son fonctionnement :

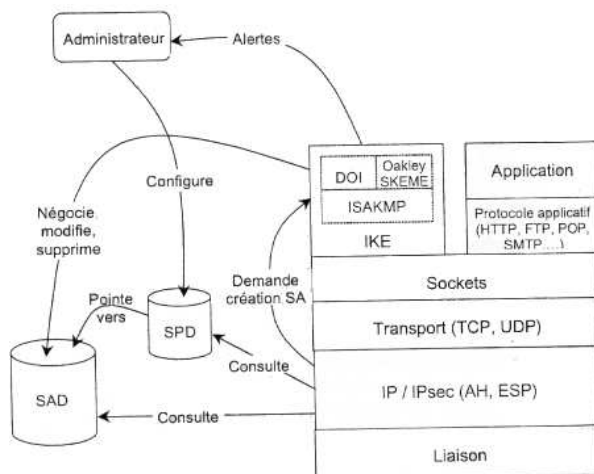


FIG. 2 – Principe d'IPSec

On note à la vue de ce schéma qu'IPSec repose sur le protocole IKE, qui permet une connexion sécurisée entre les entités désirant communiquer, et les protocoles AH et ESP, qui traitent les données utiles de la couche IP afin de les protéger selon la politique choisie ; c'est donc ces protocoles que nous allons étudier en détail.

Avant que les paquets puissent être sécurisés par IPSec, une SA (Associations de Sécurité) doit exister. Elle peut être créée manuellement ou dynamiquement. Le protocole IKE est utilisé pour la création dynamique de cette SA ; il s'agit d'un protocole hybride basé sur les protocoles ISAKMP, Oakley et SKEME : il utilise les bases de ISAKMP, les modes de Oakley et les techniques de partage des clés de SKEME.

2.1 ISAKMP : Internet Security Association and Key Management Protocol

2.1.1 Rôle

Le rôle de ISAKMP est d'établir, de négocier, de modifier ou de supprimer des Associations de Sécurité et leurs attributs.

Ce protocole constitue un cadre générique indépendant des mécanismes en faveur desquels la négociation a lieu et de ceux par lesquels la sécurisation est réalisée ; c'est pour cette dernière raison qu'un document appelé DOI² est nécessaire.

2.1.2 Principe de fonctionnement

ISAKMP se déroule en deux phases :

- Tout d'abord, création de la SA ISAKMP, qui servira à la sécurisation de l'ensemble des échanges futurs : on a donc négociation d'attributs relatifs à la sécurité, authentification des identités des tiers, génération des clés...
- Ensuite, négociation de paramètres de sécurité relatifs à une SA à établir pour un mécanisme donné (par exemple AH ou ESP), via la SA ISAKMP établie en phase 1.

2.1.3 La construction par blocs d'ISAKMP

La forme des messages ISAKMP est un en-tête suivi d'un ensemble de blocs élémentaires chaînés de taille variable. Ces messages sont échangés selon des types d'échanges définis, fournissant un certain nombre de services de sécurité spécifiques (anonymat, *perfect forward secrecy*,...).

Description des champs de la trame :

- Initiator cookie (8 octets) : Cookie du dispositif qui a initié l'établissement, la modification ou la suppression de la SA.
- Responder cookie (8 octets) : Cookie de l'entité qui a répondu à la demande d'initiation de l'établissement, la modification ou la suppression de la SA.

²DOI(*Domain Of Interpretation*) = "domaine d'interprétation". Document définissant les paramètres négociés et les conventions relatives à l'utilisation de ISAKMP dans un cadre précis

16 bit		16 bit		
Initiator cookie				
Responder cookie				
Next Payload	Major Version	Minor Version	Exchange Type	Flags
Message ID				
Message Length				

FIG. 3 – Entête ISAKMP

- Next payload : Indique le type des premières données utiles contenues dans le message (c'est-à-dire du premier bloc ISAKMP).
- Major/Minor version : Indique la version d'ISAKMP utilisée.
- Exchange type : Indique le type d'échange utilisé et définit l'ordre et le nombre des messages et des données utiles de l'échange ISAKMP.
- Flags : Indiquent les différentes options contenues dans le message. Ces options peuvent être : Encryption (chiffrement), Commit ou Authentication only (authentification seule).
- Message ID : Identificateur unique du message.
- Message length : Longueur totale du message.

On notera que les deux cookies présents au début du message servent à définir la SA en cours (la SA ISAKMP n'est en effet pas définie par un SPI).

On distingue 13 types de blocs ISAKMP différents, mais ne seront exposés ici que les types de base :

- Le bloc *Security Association* est utilisé pour négocier les attributs de sécurité. Il contient des champs qui indiquent le contexte de la négociation (DOI et situation).
Un bloc SA est toujours suivi d'un ou plusieurs blocs *Proposal*, classés par ordre de préférence.
- Le bloc *Proposal* contient une proposition numérotée correspondant à une SA : il indique le mécanisme de sécurité à utiliser et le SPI à y associer.
Un bloc *Proposal* est toujours suivi d'un ou plusieurs blocs *Transform*, classés par ordre de préférence.
- Le bloc *Transform* contient un choix de transformation (fonction de hachage, algorithme de chiffrement...) et ses attributs, relatifs au mécanisme ainsi qu'au DOI sélectionnés précédemment.
Un système de numérotation permet de combiner des transformations et/ou d'en formuler des propositions.

Ces trois types de message sont les briques de base de ISAKMP ; ils ne sont pas indépendants et sont emboîtés. Ce message correspond à l'envoi par un

expéditeur de propositions de sécurité pour une liaison avec un tiers.

Les autres blocs (*Key Exchange*, *Identification*, *Certificate*, *Certificate Request*, *Hash*, *Signature*, *Nonce*, *Notification*, *Delete* et *Vendor ID*) servent à transporter un certain nombre de données de sécurité (comme des certificats...) ou des messages informatifs entre les tiers.

Le système de blocs chaînés permet notamment l'indépendance de ISAKMP vis-à-vis de la gestion des clés, puisque le contenu, le nombre et l'agencement variables des blocs permettent une grande souplesse d'utilisation.

2.1.4 Les échanges de ISAKMP :

A partir de ces blocs, ISAKMP définit des types d'échanges qui spécifient l'ensemble des messages (nombre, contenu, ...) répondant à un type d'information à échanger entre les tiers (autrement dit à un type de service à fournir) : authentification mutuelle, anonymat, ...

On distingue 5 types par défaut :

- l'échange de base (*Base Exchange*), permet l'échange simple et rapide de toutes les données nécessaires à la communication, mais ne fournit aucune sécurité supplémentaire (comme l'anonymat...).
- l'échange de protection d'identité (*Identity Protection Exchange*) assure quant à lui l'anonymat des tiers, en n'effectuant leur authentification que sous la protection des systèmes mis en place avec l'échange des données de génération de secret partagé.
- l'échange d'authentification seule (*Authentication Only Exchange*) permet seulement l'authentification des tiers.
- l'échange agressif (*Aggressive Exchange*) assure les mêmes services que l'échange de base, mais en un nombre minimal de messages (il contracte en fait en un seul message les données de négociation de la SA, d'authentification et d'échange de clé); on notera qu'il empêche du même coup l'utilisation de l'échange de clés selon Diffie-Hellman.
- l'échange d'information (*Informational Exchange*) est un message univoque d'information concernant la gestion des SA d'un tiers (modification, suppression, ...).

2.2 IKE

IKE est le protocole de gestion des clés implémenté par IPSec. Il comprend 4 modes, qui gèrent les échanges de paramètres entre les entités souhaitant communiquer ; le but est de créer la SA dans les deux pairs à l'aide des deux phases d'ISAKMP. La première phase est utilisée pour créer une SA IKE - via les échanges identity protect exchange et aggressive exchange d'ISAKMP - , la deuxième pour négocier les paramètres nécessaires à la création de la SA IPSec.

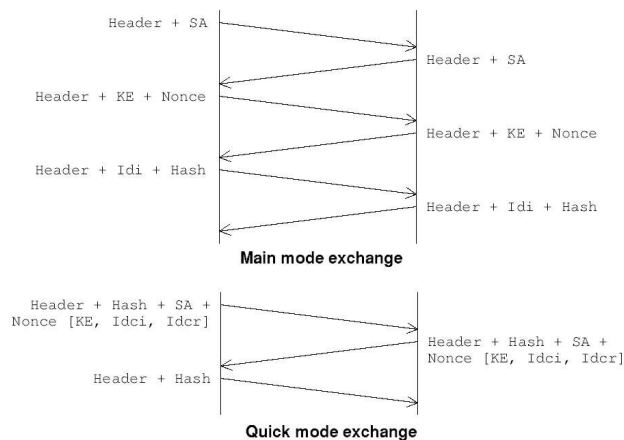


FIG. 4 – échange IKE classique

2.2.1 Main Mode Exchange

Pour l'établissement de la SA IKE, six messages sont utilisés : trois requêtes et trois réponses. Cet échange se déroule en trois étapes :

- échange des paramètres Diffie-Hellman ;
- échange d'aléas ;
- authentification des pairs.

Le premier échange sert à la négociation des paramètres nécessaires à la mise en place de la SA IKE. Le deuxième échange sert à la négociation des valeurs publiques de l'algorithme Diffie-Hellman et des valeurs pseudo-aléatoires contenues dans le bloc d'aléas (Nonce Payload). Lors du dernier échange les deux pairs s'envoient leurs identités respectives et le bloc Hash nécessaire à l'authentification.

2.2.2 Aggressive Mode Exchange

Ce mode utilise directement l'Aggressive Exchange de ISAKMP ; l'échange se déroule donc en seulement trois messages.

2.2.3 Quick Mode Exchange

Une fois la SA IKE établie avec le Main Mode ou l'Aggressive Mode, le Quick Mode est utilisé pour établir une SA pour un autre protocole de sécurité, comme AH ou ESP, sous la protection de la SA IKE précédemment établie. Dans un échange en Quick Mode, les deux pairs négocient les caractéristiques de la SA IPsec à établir, et génèrent les clés correspondantes. La SA IKE protège ces échanges en chiffrant et en authentifiant les messages transmis.

En plus de l'en-tête ISAKMP, du Hash, de la SA, du Nonce et des paramètres optionnels de Diffie-Hellman, les deux pairs peuvent s'échanger des informations concernant leur identité, comme leur adresse IP.

La connexion sécurisée ayant été établie par les protocoles sus-cités, il est alors nécessaire de protéger les données utiles : c'est le rôle des protocoles AH et ESP.

2.3 Protocoles AH et ESP

Afin d'assurer l'intégrité, l'authentification et la confidentialité des données transmises, ainsi qu'optionnellement la protection contre le rejeu, le protocole IPsec utilise deux protocoles distincts : AH (*Authentication Header*) et ESP (*Encapsulating Security Payload*).

Ces protocoles peuvent être appliqués dans deux configurations différentes, selon le type de sécurisation et de connexion utilisés :

- le mode transport, pour lequel seules les données des protocoles de niveau supérieur transportées par le datagramme IP sont protégées. Ce mode est réservé aux connexions entre équipements terminaux (problèmes de routage).
- le mode tunnel, pour lequel l'en-tête IP est également protégé et remplacé par un nouvel en-tête qui permet juste au paquet de traverser le tunnel de sécurité, à la sortie duquel l'en-tête original est rétabli. Ce mode permet une protection plus importante contre l'analyse du trafic, car il masque les adresses de l'expéditeur et du destinataire final.

2.3.1 Authentication Header

Ce protocole ne sera que rapidement évoqué car il est peu utilisé par rapport à ESP, ce dernier étant plus largement utilisé car fournissant des services plus adaptés aux besoins des entreprises.

AH assure simultanément l'authentification³ et l'intégrité⁴ des données transmises via l'ajout d'une ICV⁵, et, de façon optionnelle, la protection contre le rejeu⁶, via l'ajout d'un numéro de séquence.

Il est important de noter que les données d'authentification sont calculées à partir de l'ensemble des champs invariants du datagramme IP final (AH compris), ce qui permet d'étendre l'authentification au SPI⁷ et au numéro de séquence notamment.

2.3.2 Encapsulating Security Payload

Ce protocole peut assurer les services suivants :

- confidentialité (confidentialité des données et protection contre l'analyse du trafic en mode tunnel),
- intégrité des données en mode non connecté,
- authentification de l'origine des données,
- protection contre le rejeu.

ESP encapsule les données entre un en-tête et un en-queue, de la forme suivante :

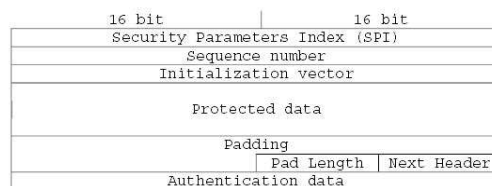


FIG. 5 – en-tête ESP

Description des champs de la trame :

³authentification = vérification de l'origine des données

⁴intégrité = similitude des données avec les données originales

⁵ICV (*Integrity Check Value*) = valeur de vérification d'intégrité (c'est-à-dire scellement ou signature)

⁶rejeu = réexpédition par un attaquant d'un message intercepté au préalable

⁷SPI (*Security Parameter Index*) = index des paramètres de sécurité

- *SPI* : Le SPI correspond à une valeur arbitraire de 32 bits qui, associée à l'adresse IP de destination et au protocole de sécurité (AH) identifie la SA qui doit être utilisé pour authentifier ce paquet. Elle est en général choisie par le système destinataire, lors de l'établissement de la SA. Ce champ est authentifié, mais pas chiffré.
- *Sequence number* : Numéro de séquence de 32 bits auto-incrémenté, employé pour se protéger d'une retransmission du paquet (anti-rejeu). Ce champ est obligatoire et toujours présent même si le récepteur désactive la fonction d'anti-rejeu. Ce champ est authentifié mais pas chiffré, afin de faciliter la détection du rejeu.
- *Initialization vector* : vecteur d'initialisation utile au chiffrement des données ; ce champ est authentifié, mais pas chiffré. Il est généralement placé dans les huit premiers octets du champ Protected data.
- *Protected data* : Ce champ contient les données chiffrées.
- *Padding* : Ce champ sert de bourrage pour le protocole ESP. Certains algorithmes de chiffrement nécessitent que la longueur du datagramme soit un multiple exact d'un nombre fixe d'octets. En fonction de l'algorithme employé, ce champ sera rempli par des 0 jusqu'à l'obtention du multiple désiré.
- *Padding length* : Ce champ donne la longueur du champ bourrage.
- *Next header* : Ce champ indique le protocole de couche supérieur. En mode transport, ce champ indique la première en-tête protégée (TCP ou UDP) ; en mode tunnel, il vaut 4, si l'on travaille avec le protocole IP.
- *Authentication data* : La longueur de ce champ dépend de l'algorithme d'authentification utilisé. Il contient le résultat du calcul d'intégrité du message ICV, suivi d'un remplissage avec des 0, pour que le champ soit multiple de 32 bits.

La confidentialité est donc appliquée dans ESP de la manière suivante :

- l'expéditeur E encapsule dans le champ "Protected data" les données du datagramme original ;
- E ajoute si nécessaire un bourrage ;
- E chiffre le résultat (données, bourrage, champs longueur et en-tête suivant) ;
- E ajoute éventuellement le vecteur d'initialisation.

Si elle a été sélectionnée, l'authentification est toujours appliquée après que les données ont été chiffrées. Cela permet, à la réception, de vérifier la validité du datagramme avant de se lancer dans la coûteuse opération de déchiffrement.

On notera également qu'à l'opposé d'AH, l'authentification est ici uniquement appliquée au "paquet" (en-tête + charge utile + en-queue) ESP, et n'inclut ni l'en-tête de niveau réseau ni le champ d'authentification.

Pour conclure, il est important de noter que la structure par système d'entêtes de ces protocoles permet de les coupler à volonté (dans les limites de la bande passante...) afin d'obtenir le degré de sécurité désiré.

Remarques :

- Il existe toujours un problème de sécurité lors de la création du canal sécurisé, car il est nécessaire de partager un secret avec son correspondant. IPSec y répond de façon assez appropriée en gérant sous IKE les certificats électroniques, Diffie-Hellman...

- La viabilité d'un VPN est du ressort de l'administrateur réseau : c'est à lui qu'incombe la charge de choisir les bonnes politiques de sécurité (via la SPD et la SAD), c'est-à-dire celles qui sont cohérentes avec la politique du réseau utilisant le VPN d'une part, et avec les capacités physiques du matériel dont il dispose d'autre part.

3 Applications

3.1 Installation

Décrivons l'installation d'IPSEC sous Debian GNU/Linux avec un noyau Linux 2.4.23 sur une machine possédant une carte ethernet.

Commençons par installer une Debian GNU/Linux 3.0r2 (www.debian.org)
Nous installons les paquets de base ainsi que quelques programmes utiles (wget, gnupg, tcpdump, make, kernel-package, etc.)

Plaçons nous ensuite dans le répertoire des sources et téléchargeons ensuite un noyau Linux version 2.4.23 (www.kernel.org) :

```
cd /usr/src
wget http://kernel.org/pub/linux/kernel/v2.4/linux-2.4.23.tar.bz2
```

Vérifions l'intégrité du noyau Linux :

```
wget http://kernel.org/pub/linux/kernel/v2.4/linux-2.4.23.tar.bz2.sign
gpg --keyserver wwwkeys.pgp.net --recv-keys 0x517D0F0E
gpg --verify linux-2.4.23.tar.bz2.sign linux-2.4.23.tar.bz2
```

Nous obtenons donc un message : Bonne signature de "Linux Kernel Archives Verification Key <ftpadm@kernel.org>"

Nous pouvons donc décompresser les sources du noyau Linux :

```
tar jxvf linux-2.4.23.tar.bz2
```

Il nous faut également télécharger le patch freeS/WAN pour IPSEC (www.freeswan.org) :

```
wget ftp://ftp.xs4all.nl/pub/crypto/freeswan/freeswan-2.04.tar.gz
md5sum freeswan-2.04.tar.gz
```

Vérifions l'intégrité :

```
wget ftp://ftp.xs4all.nl/pub/crypto/freeswan/freeswan-sigkey.asc
gpg --import freeswan-sigkey.asc
gpg --verify freeswan-2.04.tar.gz.sig freeswan-2.04.tar.gz
```

Nous obtenons donc un message : Bonne signature de "Linux FreeS/WAN Software <Team build@freeswan.org>"

Nous pouvons donc décompresser les sources freeswan :

```
tar zxvf freeswan-2.04.tar.gz
```

Et ensuite patcher et configurer le noyau Linux :

```
cd freeswan-2.04
make menugo
```

On obtient notamment le choix des options suivantes :

```
<*> IP Security Protocol (FreeS/WAN IPSEC)
--- IPsec options (FreeS/WAN)
[*]   IPSEC: IP-in-IP encapsulation (tunnel mode)
[*]   IPSEC: Authentication Header
[*]       HMAC-MD5 authentication algorithm
[*]       HMAC-SHA1 authentication algorithm
[*]   IPSEC: Encapsulating Security Payload
[*]       3DES encryption algorithm
[*]   IPSEC: IP Compression
[*]   IPSEC Debugging Option
```

On peut donc compiler notre noyau Linux. Avec la méthode Debian :

```
cd /usr/src/linux-2.4.23
make-kpkg --append-to-version=ipsec kernel\_image
```

On peut maintenant installer notre nouveau noyau :

```
dpkg -i ../kernel-image-2.4.23ipsec\_10.00.Custom\_i386.deb
```

Installons également les paquets freeswan et ipsec-tools et nous avons une machine prête à servir pour nos tests. Après avoir démarré votre machine avec ce noyau, on peut démarrer/arrêter les services IPsec grâce à :

```
/etc/init.d/ipsec start/stop
```

Une fois les services IPsec démarrés, on obtient une interface ressemblant à :

```
ipsec0
Link encap:Ethernet HWaddr 00:E0:7D:D2:00:10
inet addr:192.168.1.1 Mask:255.255.255.0
UP RUNNING NOARP MTU:16260 Metric:1
RX packets:184 errors:0 dropped:25 overruns:0 frame:0
TX packets:168 errors:0 dropped:16 overruns:0 carrier:0
collisions:0
RX bytes:12318 (12.0 KiB) TX bytes:24616 (24.0 KiB)
```

3.2 Configuration

3.2.1 Configuration manuelle des clés

Lors d'une configuration manuelle, il n'y a aucune négociation. Tous les algorithmes d'authentification, de chiffrement et les clés sont pré-définies.

Cette méthode n'est évidemment pas très sûre, notamment car l'on utilise toujours les mêmes clés pour les échanges. La configuration est dans le fichier `/etc/ipsec.conf` :

```
version 2.0
config setup
    interfaces="ipsec0=eth0"
    #mode debug
    klipsdebug=all
    #Pas d'IKE
    plutodebug=none
    pluto=no
    #identification unique
    uniqueids=yes
#connexion pour le projet
conn projet
    #adresse locale
    left=192.168.1.1
    #adresse destination
    right=192.168.1.3
    #SPI
    spi=0x104
    #Choix des algos
    esp=3des-md5-96
    #cle de chiffrement
    espenckey=
0xa111c235d412345678915a1022d448a44e8e876543211a21
    #cle d'authentification
    espauthkey=0x123456789123456789aabbccdde004b
```

3.2.2 Configuration avec un secret partagé

Il s'agit donc maintenant d'utiliser IKE. L'authentification mutuelle des hôtes afin de définir le clé de session spécifique est réalisée par un secret partagé. L'implémentation de IKE dans Freeswan se nomme Pluto. Voici le fichier de configuration `/etc/ipsec.conf` utilisé :

```
version 2.0
config setup
    interfaces="ipsec0=eth0"
    #mode debug
    klipsdebug=all
```

```

#mode debug pour Pluto
plutodebug=all
#IKE
pluto=yes
#identification unique
uniqueids=yes

#connexion pour le projet
conn projet
#adresse locale
left=192.168.1.1
#adresse destination
right=192.168.1.3
#connexion non chargée au lancement de Fresswan
auto=add
#Authentification par un secret partagé
authby=secret

```

Le secret partagé se trouve dans le fichier `/etc/ipsec.secrets` :

```
192.168.1.1 192.168.1.3: PSK "projetvpnipsec"
```

3.3 Mise en oeuvre

Deux machines sont donc opérationnelles avec une carte réseau chacune. La machine A a l'IP 192.168.1.1 et la machine B 192.168.1.3. Nous utiliserons `tcpdump` pour surveiller les interfaces réseaux.

3.3.1 Configuration manuelle des clés

On a donc deux machines configurées manuellement avec des clés identiques. Démarrons les services IPsec sur les deux machines :

```
| /etc/init.d/ipsec start
```

Aucun paquet relatif à IPsec ne passe sur les interfaces réseaux. Lançons la connexion projet sur la machine A :

```
| ipsec manual --up projet
```

Il est intéressant de constater que des paquets sont envoyés de A vers B. Tant que la connexion projet n'est pas lancée sur la machine B, les paquets ESP arrivant ne sont pas décodés. Démarrons maintenant la connexion projet sur la machine B et lançons de la machine A un ping vers B. En écoutant

sur l'interface eth0 de la machine B, on voit passer les paquets ESP :

0000	00 e0 4c eb d7 8e 00 e0	7d d2 00 10 08 00 45 00
0010	00 88 11 97 00 00 40 32	e5 58 c0 a8 01 01 c0 a8
0020	01 03 00 00 01 04 00 00	00 01 35 77 ce c7 47 44
0030	b5 70 70 02 02 4a 3a c2	04 a6 59 1d 8b 4f 37 de
0040	d6 5d 01 e5 82 bb 27 82	28 44 e5 5e 0e b2 09 f0
0050	7d 7d e4 e8 98 64 aa 42	df 88 a4 fe 77 e5 20 3e
0060	8b 42 f9 3a 06 53 fc f2	39 c5 37 96 49 e5 b0 30
0070	d5 00 a6 3d 3d 8a bf 45	8e 18 fe 03 11 41 74 55
0080	d6 a5 29 b5 47 0e a5 0d	6e 06 5f 7b 19 ee c5 30
0090	2e a2 8d 22 32 d3	

Analysons cette trame.

En-tête ethernet :

00 e0 4c eb d7 8e : adresse MAC de la machine B
00 e0 7d d2 00 10 : adresse MAC de la machine A
08 00 : type IP

En-tête des datagrammes IP :

4 : IPv4 (version)
5 : 20 bits (taille de l'en-tête)
00 : ECN (type de service)
00 88 : 136 (taille totale)
11 97 : identification
00 00 : flags et déplacement fragmentation
40 : 64 (durée de vie)
32 : ESP (protocole)
e5 58 : somme de contrôle
c0 a8 01 01 : adresse IP de la machine A (source)
c0 a8 01 03 : adresse IP de la machine B (destination)

ESP :

00 00 01 04 : SPI
00 01 35 77 : 1 (numéro de séquence)
ce c7 ... 32 d3 : données

On constate que l'on retrouve bien le SPI 0x104 tel que défini dans le fichier ipsec.conf Le numéro de séquence est incrémenté pour chaque paquet envoyé
En écoutant l'interface ipsec0 on peut voir que ce paquet ESP est bien décodé :

```

0000  00 e0 4c eb d7 8e 00 e0  7d d2 00 10 08 00 45 00
0010  00 54 00 00 40 00 40 01  b7 54 c0 a8 01 01 c0 a8
0020  01 03 08 00 e2 bb ed 33  00 00 3f f2 f8 77 00 0a
0030  04 99 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35
0060  36 37

```

Il s'agit bien du ping de A vers B.

On constate donc qu'il n'y a aucune négociation, et qu'il s'agit directement du protocole ESP, les clés étant déjà sur les machines.

On remarque au passage le coût d'IPSec (dont nous avons parlé précédemment) au travers d'un ping effectué en connexion normale :

```

ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=0.4 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.3 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.3 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.3 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.3 ms

--- 192.168.1.3 ping statistics ---
5 packets transmitted , 5 packets received , 0% packet loss
round-trip min/avg/max = 0.3/0.3/0.4 ms

```

Les temps de réponse de ces ping sont, dans cette configuration, en moyenne trente fois moins élevés qu'un ping en connexion IPsec :

```

ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=10.9 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=8.4 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=10.0 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=8.3 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=10.3 ms

--- 192.168.1.3 ping statistics ---
5 packets transmitted , 5 packets received , 0% packet loss
round-trip min/avg/max = 8.3/9.5/10.9 ms

```

3.3.2 Configuration avec un secret partagé

On a donc deux machines configurées manuellement avec ce secret partagé.

```
| /etc/init.d/ipsec start
```

Aucun paquet relatif à IPsec ne passe sur les interfaces réseaux (à cause du paramètre auto=add). Lançons la connexion projet sur la machine A :

```
| ipsec auto --up projet
| 104 "projet" #1: STATE_MAIN_I1: initiate
| 106 "projet" #1: STATE_MAIN_I2: sent MI2, expecting MR2
| 108 "projet" #1: STATE_MAIN_I3: sent MI3, expecting MR3
| 004 "projet" #1: STATE_MAIN_I4: ISAKMP SA established
| 112 "projet" #2: STATE_QUICK_I1: initiate
| 004 "projet" #2: STATE_QUICK_I2: sent QI2,
| IPsec SA established {ESP=>0xbc431dcf <0x1b6213bb}
```

La connexion s'est donc correctement déroulée. En écoutant sur l'interface eth0 de la machine B, on voit passer les paquets correspondant à IKE. Voici les traces de cet échange :

```
| 18:18:48.338685
| A.500 > B.500: isakmp: phase 1 I ident: [|sa]
| 18:18:48.813416
| B.500 > A.500: isakmp: phase 1 R ident: [|sa]
| 18:18:50.523404
| A.500 > B.500: isakmp: phase 1 I ident: [|ke]
| 18:18:51.147818
| B.500 > A.500: isakmp: phase 1 R ident: [|ke]
| 18:18:52.472836
| A.500 > B.500: isakmp: phase 1 I ident[E]: [|id]
| 18:18:52.793588
| B.500 > A.500: isakmp: phase 1 R ident[E]: [|id]
| 18:18:55.545359
| A.500 > B.500: isakmp: phase 2/others I oakley-quick[E]: [|hash]
| 18:18:57.253421
| B.500 > A.500: isakmp: phase 2/others R oakley-quick[E]: [|hash]
| 18:19:02.239734
| A.500 > B.500: isakmp: phase 2/others I oakley-quick[E]: [|hash]
```

On retrouve bien l'échange IKE défini dans la partie 2.

- Security Association :

Il est intéressant de regarder la première trame dans laquelle la machine A propose sa politique de sécurité :

0000	00	e0	4c	eb	d7	8e	00	e0	7d	d2	00	10	08	00	45	00
0010	00	cc	00	00	40	00	40	11	b6	cc	c0	a8	01	01	c0	a8
0020	01	03	01	f4	01	f4	00	b8	b6	b6	c7	ff	ca	f6	d4	13
0030	0c	19	00	00	00	00	00	00	00	00	01	10	02	00	00	00
0040	00	00	00	00	00	b0	00	00	00	94	00	00	00	01	00	00
0050	00	01	00	00	00	88	00	01	00	04	03	00	00	20	00	01
0060	00	00	80	0b	00	01	80	0c	0e	10	80	01	00	05	80	02
0070	00	01	80	03	00	01	80	04	00	05	03	00	00	20	01	01
0080	00	00	80	0b	00	01	80	0c	0e	10	80	01	00	05	80	02
0090	00	02	80	03	00	01	80	04	00	05	03	00	00	20	02	01
00a0	00	00	80	0b	00	01	80	0c	0e	10	80	01	00	05	80	02
00b0	00	02	80	03	00	01	80	04	00	02	00	00	00	20	03	01
00c0	00	00	80	0b	00	01	80	0c	0e	10	80	01	00	05	80	02
00d0	00	01	80	03	00	01	80	04	00	02						

Analysons cette trame. L'en-tête IP précise bien qu'il s'agit du protocole de transport UDP (11)

Entête UDP :

01 f4 : 500 (port source)
 01 f4 : 500 (port destination)
 00 b8 : 184 (taille)
 b6 b6 : somme de contrôle

ISAKMP :

c7 ff ca f6 d4 13 0c 19 : initiator cookie
 00 00 00 00 00 00 00 00 : responder cookie
 01 : Security Association (next payload)
 10 : version 1.0 de isakmp
 02 : main mode (exchange type)
 00 : flags
 00 00 00 00 : message ID
 00 00 00 b0 : 176 (taille)

Security Association Payload :

00 : next payload
 00 94 : 148 (taille)
 00 00 00 01 : DOI IPSEC
 00 00 00 01 : IDENTITY

Proposal Payload :

00 : next payload
 00 88 : 136 (taille)

00 : proposal number
01 : ISAKMP 1 (protocol ID)
00 : SPI size
04 : nombre de propositions de Transform Payload

Transform Payload :

03 : next transform
00 20 : 32 (taille)
00 : transform number
01 : KEY IKE (transform ID)
00 00
80 0b : life-type (champ type d'unités)
00 01 : secondes (type d'unités)
80 0c : life-duration (champ durée de vie)
0e 10 : 3600 (durée de vie)
80 01 : encryption-algorithm (champ type de chiffrement)
00 05 : 3DES-CBC (type de chiffrement)
80 02 : hash-algorithm (type fonction de hashage)
00 02 : SHA (type de fonction de hashage)
80 03 : authentication-method (champ méthode d'authentification)
00 01 : PSK (méthode d'authentification)
80 04 : group-description
00 02 : valeur du groupe

Il y a ensuite les 3 autres propositions différents de Transform Payload. Le champ transform number est incrémenté pour chaque proposition. Elles proposent une fonction de hashage différente (MD5) ou une valeur de groupe différente.

Les trames de cet échange sont placées dans l'annexe 2.

La machine A envoie donc un premier paquet IKE en main mode (le responder cookie est donc nul). On constate que la machine A fait 4 propositions à la machine B.

Ensuite, la machine B répond à la machine A (en reprenant l'initiator cookie et donnant un responder cookie). La machine B envoie le Transform Payload qui lui convient le mieux. Dans notre exemple, la machine B va retenir la proposition d'une durée de vie d'une heure, d'un chiffrement 3DES-CBC et d'une fonction de hashage MD5 notamment.

- Key Exchange :

La machine A envoie un paquet pour procéder à l'échange de la clé de session. Les champs initiator cookie et responder cookie sont toujours les mêmes. Le champ next payload vaut 04 ce qui signifie échange de clé. Les champs les plus intéressants sont :

Key Exchange Payload :
0a : aucun (next payload)
00 c4 : 196 (taille)
21 dd .. a1 a8 : données de Diffie-Hellman pour l'échange de la clé

Nonce Payload :
00 : aucun (next payload)
00 14 : 20 (taille)
78 fe .. a6 97 : nonce data (valeurs pseudo-aléatoires)

La machine B répond et envoie de la même façon des données de Diffie-Hellman pour l'échange de la clé.

- Identification :

Ensuite, la machine A envoie un paquet pour procéder à l'identification. Le champ next payload vaut 05 ce qui signifie identification.

Le champ flags contient donc essentiellement Encrypted Payload (32 Bytes).

- Hash :

On entre dans la phase 2 d'IKE. Le champ Exchange type vaut maintenant 20 ce qui correspond à au Quick mode.

L'échange des paramètres est chiffré et nous pouvons simplement constater que la machine A envoie 320 bits de données chiffrées (encrypted payload), que la machine B envoie 288 bits de données chiffrées et que finalement la machine A envoie 24 bits de données chiffrées. Cet échange sécurisé correspond à la fabrication des clés de session.

La phase d'échange des paramètres de sécurité grâce à IKE est terminée. L'échange de données entre les machines A et B utilise IPSEC. Dans notre exemple, un ping entre A et B utilise donc le protocole ESP. Les trames sont donc du même type que la trame analysée dans la configuration manuelle des clés.

3.4 Commentaires généraux

Les noyaux Linux 2.6 ont désormais le support d'IPSec en natif. La configuration n'est plus similaire à FreeS/WAN mais se rapproche des systèmes *BSD : FreeBSD, NetBSD et OpenBSD. Nul doute, qu'IPsec est devenu le standard pour les VPN, lesquels ont un bel avenir devant eux.

En ce qui concerne l'installation et la configuration d'IPSec sous d'autres plateformes, les *BSD ont le support IPSec en natif ainsi que MS-Windows. Nous avons testé rapidement l'installation et la configuration sous FreeBSD 5.1 et OpenBSD 3.4 et nous n'avons pas rencontré de problèmes particuliers.

Par contre, malgré la théorie d'interopérabilité, nous avons eu quelques problèmes entre une machine Debian GNU/Linux et FreeBSD 5.1 mais nous n'avons pas eu le temps de nous attarder dessus.

4 Conclusion

Les technologies de transmission des données de type VPN vont être de plus en plus utilisées pour de nombreuses raisons (développement du télé-travail, démocratisation du haut débit, émergence des technologies de connexions sans fil, etc.). Les outils IPSec s'imposent comme l'un des standards les plus utilisés. Par nos applications, nous avons constatés que la mise en place de telles solutions est assez aisée et peu coûteuse.

Les difficultés consistent à déterminer la politique de sécurité de l'entreprise, et que cette politique soit compatible avec les moyens dont elle dispose (bande passante, matériel). Il est important de bien étudier cette politique au risque d'avoir un système non utilisable ou inutile !

Les problématiques techniques des technologies VPN sont les limites de la bande passante. Une connexion distante à un réseau local est limitée. Certains services, comme un simple transfert de fichiers, ont donc un faible débit lors d'une connexion VPN par rapport aux mêmes services sur un réseau local réel. Le développement de connexions Internet avec des débits plus élevés estompera petit à petit ce problème.

5 Annexe

5.1 Annexe 1

Pour qu'il y ait authentification, il faut fournir la preuve de son identité auprès de son interlocuteur. Il existe plusieurs technologies dont voici les 3 principales :

Les certificats digitaux : Un certificat est constitué d'une clé publique et d'un certain nombre de champs d'identification, le tout signé par un tiers certificateur. De plus, un certificat contient des informations de gestion (numéro de série, date d'expiration,...). Ils se basent sur les recommandations X509 et permettent de façon sûre d'authentifier une personne. On fournit à une autorité de certification les informations et celle-ci retourne un certificat digital.

Ces certificats sont composés de 2 parties : les informations concernant l'entité (nom, clé publique, adresse physique) et un résumé chiffré de ces informations. Le résumé de ces informations est effectué par un algorithme de hachage tel MD5 ou SHA-1, qui retourne un numéro unique qui est ensuite chiffré. Lorsqu'un certificat est transmis à une entité qui peut vérifier l'authenticité d'une autre, elle procède en 4 étapes :

elle sépare les informations de l'entité et le résumé chiffré

elle déchiffre le résumé chiffré

elle recalcule un résumé en utilisant le même algorithme (MD5)

elle compare le résumé calculé par ses soins et le résumé déchiffré : si les résultats correspondent, l'authenticité est prouvée.

Avec la figure ci-dessous on peut mieux comprendre ce phénomène :

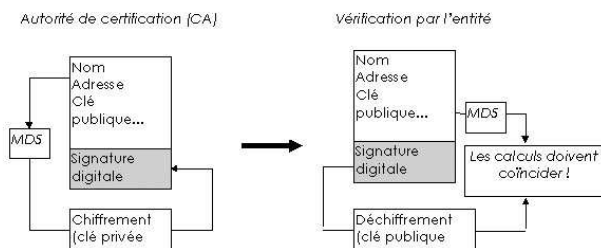


FIG. 6 – Authentification

L'autorité de certification peut être de 2 types : soit propriétaire et fournie par le constructeur ou bien externe. C'est alors une société tiers à qui l'on délègue la gestion de sa PKI (Public Key Infrastructure).

Phrase challenge : Le processus est similaire à celui utilisé dans le cas des certificats digitaux. La différence réside en l'absence d'autorité de certification ; les entités doivent elles mêmes générer leurs certificats digitaux. La signature est alors chiffrée par une phrase challenge commune aux 2 entités. Il faut donc que celle-ci soit entrée dans tous les équipements désirant communiquer

Radius : Ce système utilise un serveur d'authentification RADIUS. Lors d'une demande de connexion d'un client sur un équipement VPN, ce dernier demande le mot de passe et l'identifiant RADIUS du client. Ensuite, l'équipement VPN utilise sa clé secrète pour vérifier l'authentification auprès du serveur RADIUS.

5.2 Annexe 2

Voici les trames IKE dans la configuration avec un secret partagé :

```
0000 00 e0 4c eb d7 8e 00 e0 7d d2 00 10 08 00 45 00
0010 00 cc 00 00 40 00 40 11 b6 cc c0 a8 01 01 c0 a8
0020 01 03 01 f4 01 f4 00 b8 b6 b6 c7 ff ca f6 d4 13
0030 0c 19 00 00 00 00 00 00 00 00 01 10 02 00 00 00
0040 00 00 00 00 00 b0 00 00 00 94 00 00 00 01 00 00
0050 00 01 00 00 00 88 00 01 00 04 03 00 00 20 00 01
0060 00 00 80 0b 00 01 80 0c 0e 10 80 01 00 05 80 02
0070 00 01 80 03 00 01 80 04 00 05 03 00 00 20 01 01
0080 00 00 80 0b 00 01 80 0c 0e 10 80 01 00 05 80 02
0090 00 02 80 03 00 01 80 04 00 05 03 00 00 20 02 01
00a0 00 00 80 0b 00 01 80 0c 0e 10 80 01 00 05 80 02
00b0 00 02 80 03 00 01 80 04 00 02 00 00 00 20 03 01
00c0 00 00 80 0b 00 01 80 0c 0e 10 80 01 00 05 80 02
00d0 00 01 80 03 00 01 80 04 00 02
```

18 :18 :48.338685 A.500 > B.500 : isakmp : phase 1 I ident : [—sa] (DF)

```
0000 00 e0 7d d2 00 10 00 e0 4c eb d7 8e 08 00 45 00
0010 00 6c 00 00 40 00 40 11 b7 2c c0 a8 01 03 c0 a8
0020 01 01 01 f4 01 f4 00 58 02 10 c7 ff ca f6 d4 13
0030 0c 19 b4 da 95 f2 d0 03 d5 da 01 10 02 00 00 00
0040 00 00 00 00 00 50 00 00 00 34 00 00 00 01 00 00
0050 00 01 00 00 00 28 00 01 00 01 00 00 00 20 00 01
0060 00 00 80 0b 00 01 80 0c 0e 10 80 01 00 05 80 02
0070 00 01 80 03 00 01 80 04 00 05
```

18 :18 :48.813416 B.500 > A.500 : isakmp : phase 1 R ident : [—sa] (DF)

```
0000 00 e0 4c eb d7 8e 00 e0 7d d2 00 10 08 00 45 00
0010 01 10 00 00 40 00 40 11 b6 88 c0 a8 01 01 c0 a8
0020 01 03 01 f4 01 f4 00 fc 7e 35 c7 ff ca f6 d4 13
0030 0c 19 b4 da 95 f2 d0 03 d5 da 04 10 02 00 00 00
0040 00 00 00 00 00 f4 0a 00 00 c4 21 dd 7f 0f 02 c5
0050 85 e1 ef dc d3 34 03 f6 b4 26 02 29 ad 12 df f4
0060 aa 11 13 54 38 5e 70 e2 6d e4 a9 e9 e2 42 7a 5b
0070 53 18 fb 43 ec 81 40 4f df 0f 15 8a 76 5c cc 3a
0080 bd 03 49 a1 9f c2 07 d9 8f 52 2f ef 03 cd 94 a1
0090 a3 f7 a4 9d 0a 01 2b 7d df 46 73 8d 98 4a 1c 29
00a0 23 f5 2f 2b 7d 0e b9 ef 97 46 e2 d8 c5 f4 c9 9e
00b0 f2 5a d3 f0 26 e9 fc 10 97 84 d1 74 42 0f 9a 37
00c0 c7 5d 73 60 11 ba a1 10 df 81 fa 95 af 7a 87 8c
```

00d0	e4	43	98	64	96	49	6f	d5	ee	60	42	e2	7c	31	b5	10
00e0	df	28	91	0d	5a	c7	06	bd	f0	94	a5	b6	ae	44	07	2c
00f0	6a	2b	78	08	9b	f0	88	11	9b	f0	5d	28	33	05	ac	70
0100	9e	53	23	ab	2f	54	79	09	a1	a8	00	00	00	14	78	fe
0110	e5	07	b6	cd	4a	0c	1e	6c	7c	ef	e6	c6	a6	97		

18 :18 :50.523404 A.500 > B.500 : isakmp : phase 1 I ident : [—ke] (DF)

0000	00	e0	7d	d2	00	10	00	e0	4c	eb	d7	8e	08	00	45	00
0010	01	10	00	00	40	00	40	11	b6	88	c0	a8	01	03	c0	a8
0020	01	01	01	f4	01	f4	00	fc	24	d7	c7	ff	ca	f6	d4	13
0030	0c	19	b4	da	95	f2	d0	03	d5	da	04	10	02	00	00	00
0040	00	00	00	00	00	f4	0a	00	00	c4	f5	51	3c	4c	8a	75
0050	8e	ce	09	75	24	62	d5	70	8f	11	35	68	be	cf	a5	f1
0060	86	ef	27	1d	b8	39	d5	f1	32	ac	5a	b3	f4	9b	c8	49
0070	c7	ed	86	46	9f	af	c6	50	ff	36	a3	12	ea	01	6f	62
0080	79	f7	0e	4a	91	c3	7a	da	a6	75	20	e9	ba	02	84	e4
0090	01	03	ec	07	3f	f3	bb	a5	c6	44	5e	91	2f	4b	9c	49
00a0	f8	23	d3	fb	46	a9	90	3d	00	97	90	20	7c	ae	1f	5f
00b0	49	92	6f	1f	74	50	9b	7c	6a	61	55	e5	42	38	de	e9
00c0	76	b0	a8	6a	f6	7c	dc	a1	df	1c	8a	73	33	99	1e	68
00d0	03	d8	b6	db	b8	87	b8	92	c6	db	fa	9d	b5	0c	65	cb
00e0	b6	06	5b	bf	c9	03	7c	9c	9a	87	59	12	d6	54	13	b8
00f0	79	b2	a0	a2	7c	b9	02	1a	3f	16	5c	56	4c	07	c5	37
0100	bc	f3	aa	52	10	ab	81	cf	9b	b3	00	00	00	14	a2	8d
0110	0b	d5	f6	9c	75	16	5f	54	32	b8	4e	ed	7f	37		

18 :18 :51.147818 B.500 > A.500 : isakmp : phase 1 R ident : [—ke] (DF)

0000	00	e0	4c	eb	d7	8e	00	e0	7d	d2	00	10	08	00	45	00
0010	00	58	00	00	40	00	40	11	b7	40	c0	a8	01	01	c0	a8
0020	01	03	01	f4	01	f4	00	44	88	30	c7	ff	ca	f6	d4	13
0030	0c	19	b4	da	95	f2	d0	03	d5	da	05	10	02	01	00	00
0040	00	00	00	00	00	3c	ec	c1	b4	38	0d	67	d6	28	9a	84
0050	2e	61	ee	c3	8f	f1	50	ca	ad	4b	12	21	5a	06	d0	dc
0060	96	cf	0e	2d	d8	9f										

18 :18 :52.472836 A.500 > B.500 : isakmp : phase 1 I ident[E] : [—id] (DF)

0000	00	e0	7d	d2	00	10	00	e0	4c	eb	d7	8e	08	00	45	00
0010	00	58	00	00	40	00	40	11	b7	40	c0	a8	01	03	c0	a8
0020	01	01	01	f4	01	f4	00	44	8f	29	c7	ff	ca	f6	d4	13
0030	0c	19	b4	da	95	f2	d0	03	d5	da	05	10	02	01	00	00
0040	00	00	00	00	00	3c	7c	62	1f	54	e8	87	c9	44	7e	19
0050	bf	19	a6	2d	2e	7c	76	32	c6	4f	77	b4	fa	10	c7	02

```
0060 ad cd 4b 91 af d9
```

```
18 :18 :52.793588 B.500 > A.500 : isakmp : phase 1 R ident[E] : [—id] (DF)
```

```
0000 00 e0 4c eb d7 8e 00 e0 7d d2 00 10 08 00 45 00
0010 01 78 00 00 40 00 40 11 b6 20 c0 a8 01 01 c0 a8
0020 01 03 01 f4 01 f4 01 64 27 43 c7 ff ca f6 d4 13
0030 0c 19 b4 da 95 f2 d0 03 d5 da 08 10 20 01 b6 0b
0040 72 9e 00 00 01 5c ef c5 1e 30 27 f1 c7 d8 b4 54
0050 c8 e1 b9 87 7c f8 ff bb 9f 98 be 19 f8 1c 0b 9f
0060 57 f8 be 03 8b 68 d0 c1 ff 2f 46 c7 3f 04 00 cb
0070 2f 90 14 9d 98 7e f5 af a3 f1 03 32 37 05 06 eb
0080 f2 91 b8 53 64 46 cb 85 af 22 00 51 9c f7 28 42
0090 a9 7f 9b 99 b9 4c 45 55 73 24 bb 57 5e 8d f9 eb
00a0 69 04 93 f6 07 84 f2 44 49 f4 7f ff 2c e1 7a c9
00b0 a1 8b 14 ae b8 7d d5 60 b9 e4 a5 70 47 85 a1 dc
00c0 ee a7 ba 61 86 82 47 0f 19 ce a2 09 7a fd aa c8
00d0 b0 91 73 b5 d8 ce 4a e0 34 6d 79 13 e9 11 15 9f
00e0 49 2b a2 6d 0a 5d 25 43 34 68 4a 76 c9 c9 ec 9e
00f0 f7 29 6d 62 22 49 f9 a3 9a 76 04 ba 46 37 31 c5
0100 da c4 57 40 c5 2d 92 52 b6 c6 74 f2 88 c5 dd 1a
0110 36 6c 66 5b e9 36 85 bf 55 57 43 7d 48 03 72 5a
0120 4f 62 b1 5b 10 7f c2 e7 d7 83 79 1b 02 0f a6 0f
0130 4f c6 60 3c 16 ee 8c 49 e9 90 a6 6e 4d cc fc 62
0140 2f f1 67 a8 12 3f 67 a7 c9 69 d1 80 81 04 8d 32
0150 32 7e e5 9c 53 8d 5f 38 28 be 40 b1 6e e9 5b ba
0160 24 d9 6a 3f 8e 93 4d 84 fe a3 f7 b0 e6 6c 03 8a
0170 4a fc 04 56 56 e4 3c 67 8b 8a 5b 5e 43 c2 9e c9
0180 64 b2 42 2e c6 96
```

```
18 :18 :55.545359 A.500 > B.500 : isakmp : phase 2/others I oakley-quick[E] :
[—hash] (DF)
```

```
0000 00 e0 7d d2 00 10 00 e0 4c eb d7 8e 08 00 45 00
0010 01 58 00 00 40 00 40 11 b6 40 c0 a8 01 03 c0 a8
0020 01 01 01 f4 01 f4 01 44 c5 8b c7 ff ca f6 d4 13
0030 0c 19 b4 da 95 f2 d0 03 d5 da 08 10 20 01 b6 0b
0040 72 9e 00 00 01 3c 4c f3 e4 89 ff 98 a2 1b aa 1d
0050 e9 44 66 46 50 d2 be 8a 1c 70 ad 75 69 35 04 e8
0060 e9 52 c2 4d 68 24 4a 5e bc 58 04 e2 0e b1 0e dc
0070 65 6c 59 61 eb 1d a5 4c ae 98 8b 14 6b 25 70 3e
0080 a7 e4 b9 10 bc 0a a6 a4 a6 b8 06 b6 7c 2d 52 3a
0090 70 ad 17 8c 01 57 c1 8b 55 a8 27 fe 4e 91 90 97
00a0 42 30 98 02 6e 86 95 3f b3 f6 81 a7 4d 8b f9 4d
00b0 86 c1 46 9a 40 8b 3c fd 1b 8f 18 35 be 40 36 73
```

```

00c0  2e d2 c4 0e 89 1e 0c 2d  d6 65 5d 19 52 e2 55 8c
00d0  75 ea ee 8c dc 49 2c c1  05 eb fc ff 83 68 32 ad
00e0  16 9d 06 79 fb d0 8a 4d  af 2c af 34 12 85 85 e8
00f0  71 04 0f f6 de a7 d0 ee  96 8d 33 5e 86 92 f1 c0
0100  1c 11 6a f8 0c 66 b0 52  af bf cf 76 ee c6 54 0f
0110  83 a5 a8 01 a1 e8 94 9e  79 62 8e ea f6 6b 53 18
0120  2d dc bf 0e bc 5e 61 b1  dd 49 e5 8a 7e 35 3c 02
0130  e8 27 7e 50 ae dc ee 52  ce 61 c4 45 cb a8 44 08
0140  a1 b3 b4 3a e1 49 e2 2b  e4 4d 23 b4 d8 b3 48 22
0150  e6 cb 22 76 65 d1 e2 65  b4 4d 52 75 28 bd 93 b5
0160  64 45 07 a3 4e 17

```

18:18:57.253421 B.500 > A.500 : isakmp : phase 2/others R oakley-quick[E] :
[—hash] (DF)

```

0000  00 e0 4c eb d7 8e 00 e0  7d d2 00 10 08 00 45 00
0010  00 50 00 00 40 00 40 11  b7 48 c0 a8 01 01 c0 a8
0020  01 03 01 f4 01 f4 00 3c  dd f0 c7 ff ca f6 d4 13
0030  0c 19 b4 da 95 f2 d0 03  d5 da 08 10 20 01 b6 0b
0040  72 9e 00 00 00 34 a7 ab  17 c3 3d c7 94 c3 54 0a
0050  91 81 44 03 07 40 bf 50  6e 9e 3e 33 b6 9e

```

18:19:02.239734 A.500 > B.500 : isakmp : phase 2/others I oakley-quick[E] :
[—hash] (DF)

```

0000  00 e0 4c eb d7 8e 00 e0  7d d2 00 10 08 00 45 00
0010  00 88 7e e6 00 00 40 32  78 09 c0 a8 01 01 c0 a8
0020  01 03 bc 43 1d cf 00 00  00 01 60 d0 32 d5 92 f1
0030  a9 3f 3d cc 35 6c 67 46  32 c0 57 cd da c3 5b 05
0040  2a b1 6e 8c 77 5c b3 53  41 f6 3e 49 16 a4 11 29
0050  88 5d 66 3d 24 e6 21 2d  5b df 10 a3 ac fb 1d b9
0060  b6 84 a6 14 0c e2 ad 53  fe 73 86 82 49 17 ca 41
0070  b4 22 65 74 e1 e2 97 a7  4a 70 d4 57 aa 13 cf 70
0080  52 8f 68 7e 4f 0b 5f 27  30 54 3e da a3 75 36 8d
0090  ee dc 1a 60 7c 0f

```

Et au final, voici la première trame ESP qui a suivi la négociation IKE.

18:19:16.659494 B > A : ESP(spi=0x1b6213bb,seq=0x1)

Table des figures

1	Le VPN permet de relier deux réseaux locaux	6
2	Principe d'IPSec	9
3	Entête ISAKMP	11
4	échange IKE classique	13
5	en-tête ESP	15
6	Authentification	30

6 Bibliographie

- TUTORIAL VPN (Christian Tettamanti)
http://docpacks.tcom.ch/data/VPN/Tutorial_VPN.pdf
- IPSEC : PRÉSENTATION TECHNIQUE (Ghislaine Labouret)
<http://www.hsc.fr/ressources/articles/ipsec-tech/index.html.fr>
- Magazine MISC numéro 10 (novembre-décembre 2003)
- Modélisation des performances du protocole IPsec (Benoit Joseph)
<http://benoit-joseph.mine.nu/tfe/>
- The official IPsec Howto for Linux
<http://www.ipsec-howto.org/>
- Linux FreeS/WAN
<http://www.freeswan.org/>
- Debian GNU/Linux
<http://www.debian.org>
- Les Réseaux Privés Virtuels (Jean-Marie Guillemot)
<http://www.guill.net/index.php?cat=2&ccm=10>
- Installation/configuration OpenVPN (Guillaume LEHMANN)
<http://lehmann.free.fr/InstallOpenVPN.html>
- Les RFC :
IPSEC : <ftp://ftp.rfc-editor.org/in-notes/rfc2411.txt>
IKE : <ftp://ftp.rfc-editor.org/in-notes/rfc2409.txt>
ISAKMP : <ftp://ftp.rfc-editor.org/in-notes/rfc2408.txt>
ESP : <ftp://ftp.rfc-editor.org/in-notes/rfc2406.txt>
AH : <ftp://ftp.rfc-editor.org/in-notes/rfc2402.txt>
IPSEC DOI for ISAKMP : <ftp://ftp.rfc-editor.org/in-notes/rfc2407.txt>
OAKLEY : <ftp://ftp.rfc-editor.org/in-notes/rfc2412.txt>