# IPSEC : PRÉSENTATION TECHNIQUE

---

**Ghislaine Labouret**

**Hervé Schauer Consultants (HSC)**

142, rue de Rivoli

75001 Paris

FRANCE

http://www.hsc.fr/

**IPsec : présentation technique**

Par Ghislaine LABOURET (Ghislaine.Labouret@hsc.fr)

Version du 16 juin 2000

Mots-clef : sécurité, protocoles, IP, cryptographie, confidentialité, intégrité, authentification, gestion des clefs, AH, ESP, ISAKMP, IKE

# TABLE DES MATIÈRES

# Introduction

Le terme IPsec (*IP Security Protocol*) désigne un **ensemble de mécanismes destinés à protéger le trafic au niveau d'IP** (IPv4 ou IPv6). Les services de sécurité offerts sont **l'intégrité en mode non connecté, l'authentification de l'origine des données, la protection contre le rejeu et la confidentialité** (confidentialité des données et protection partielle contre l'analyse du trafic). Ces services sont fournis au niveau de la couche IP, **offrant donc une protection pour IP et tous les protocoles de niveau supérieur**. Optionnel dans IPv4, IPsec est obligatoire pour toute implémentation de IPv6. Une fois IPv6 en place, il sera ainsi possible à tout utilisateur désirant des fonctions de sécurité d'avoir recours à IPsec.

IPsec est développé par un groupe de travail du même nom à l'IETF (*Internet Engineering Task Force*), groupe qui existe depuis 1992. Une première version des mécanismes proposés a été publiée sous forme de *RFC* en 1995, sans la partie gestion des clefs. Une seconde version, qui comporte en plus la définition du protocole de gestion des clefs IKE, a été publiée en novembre 1998. Mais IPsec reste une norme non figée qui fait en ce moment même l'objet de multiples *Internet drafts*, notamment sur la protection des accès distants. Cette présentation couvre uniquement les *RFC* de novembre 1998.

# 1. Vue d'ensemble

Cette partie présente le principe de fonctionnement d'IPsec, les différents éléments qui le composent, la façon dont ils interagissent et les différentes utilisations possibles.

## *1.1. Architecture d'IPsec*

Pour sécuriser les échanges ayant lieu sur un réseau TCP/IP, il existe plusieurs approches, en particulier en ce qui concerne le niveau auquel est effectuée la sécurisation : niveau applicatif (*mails* chiffrés par exemple), niveau transport (TLS/SSL, SSH…), ou à l'opposé niveau physique (boîtiers chiffrant toutes les données transitant par un lien donné). IPsec, quant à lui, vise à sécuriser les échanges au niveau de la couche réseau.

### 1.1.1. Les mécanismes AH et ESP

Pour cela, IPsec fait appel à deux mécanismes de sécurité pour le trafic IP, les "**protocoles" AH et ESP**, qui viennent s'ajouter au traitement IP classique :

- *Authentication Header* (AH) est conçu pour assurer **l'intégrité et l'authentification** des datagrammes IP sans chiffrement des données (i.e. sans confidentialité).
  Le principe de AH est d'adjoindre au datagramme IP classique un champ supplémentaire permettant à la réception de vérifier l'authenticité des données incluses dans le datagramme.
- *Encapsulating Security Payload* (ESP) a pour rôle premier d'assurer la **confidentialité**, mais peut aussi assurer l'authenticité des données.
  Le principe de ESP est de générer, à partir d'un datagramme IP classique, un nouveau datagramme dans lequel les données et éventuellement l'en-tête original sont chiffrés.

Ces mécanismes peuvent être utilisés seuls ou combinés pour obtenir les fonctions de sécurité désirées. Ils seront décrits plus en détails au chapitre "Les mécanismes de sécurité : AH et ESP".

### 1.1.2. La notion d'association de sécurité

Les mécanismes mentionnés ci-dessus font bien sûr appel à la cryptographie, et utilisent donc un certain nombre de paramètres (algorithmes de chiffrement utilisés, clefs, mécanismes sélectionnés...)

sur lesquels les tiers communicants doivent se mettre d'accord. Afin de gérer ces paramètres, IPsec a recours à la notion d'*association de sécurité* (*Security Association, SA*).

**Une association de sécurité IPsec** est une "connexion" simplexe qui fournit des services de sécurité au trafic qu'elle transporte. On peut aussi la considérer comme une **structure de données servant à stocker l'ensemble des paramètres associés à une communication donnée**.

Une SA est **unidirectionnelle** ; en conséquence, protéger les deux sens d'une communication classique requiert deux associations, une dans chaque sens. Les services de sécurité sont fournis par l'utilisation soit de AH soit de ESP. Si AH et ESP sont tous deux appliqués au trafic en question, deux SA (voire plus) sont créées ; on parle alors de paquet (*bundle*) de SA.

Chaque association est identifiée de manière unique à l'aide d'un triplet composé de :
- L'adresse de destination des paquets.
- L'identifiant d'un protocole de sécurité utilisé (AH ou ESP).
- Un *index des paramètres de sécurité* (*Security Parameter Index, SPI*).
  Un SPI est un bloc de 32 bits inscrit en clair dans l'en-tête de chaque paquet échangé ; il est choisi par le récepteur.

Pour gérer les associations de sécurité actives, on utilise une "base de données des associations de sécurité" (*Security Association Database, SAD*). Elle contient tous les paramètres relatifs à chaque SA et sera consultée pour savoir comment traiter chaque paquet reçu ou à émettre.

### 1.1.3. La gestion des clefs et des associations de sécurité

Comme nous l'avons mentionné au paragraphe précédent, les SA contiennent tous les paramètres nécessaires à IPsec, notamment les clefs utilisées. **La gestion des clefs pour IPsec n'est liée aux autres mécanismes de sécurité de IPsec que par le biais des SA**. Une SA peut être configurée manuellement dans le cas d'une situation simple, mais la règle générale est d'utiliser un protocole spécifique qui permet la négociation dynamique des SA et notamment l'échange des clefs de session.

D'autre part, IPv6 n'est pas destiné à supporter une gestion des clefs "en bande", c'est-à-dire où les données relatives à la gestion des clefs seraient transportées à l'aide d'un en-tête IPv6 distinct. Au lieu de cela on utilise un système de gestion des clefs dit "hors bande", où **les données relatives à la gestion des clefs sont transportées par un protocole de couche supérieure tel que UDP ou TCP**. Ceci permet le découplage clair du mécanisme de gestion des clefs et des autres mécanismes de sécurité. Il est ainsi possible de substituer une méthode de gestion des clefs à une autre sans avoir à modifier les implémentations des autres mécanismes de sécurité.

**Le protocole de négociation des SA** développé pour IPsec s'appelle "protocole de gestion des clefs et des associations de sécurité pour Internet" (*Internet Security Association and Key Management Protocol, ISAKMP*). **ISAKMP** est en fait inutilisable seul : c'est un cadre générique qui permet l'utilisation de plusieurs protocoles d'échange de clef et qui peut être utilisé pour d'autres mécanismes de sécurité que ceux de IPsec. Dans le cadre de la standardisation de IPsec, ISAKMP est associé à une partie des protocoles SKEME et Oakley pour donner un protocole final du nom d'**IKE (*Internet Key Exchange*)**. Ces protocoles seront décrits en détail au chapitre "La gestion des clefs" page 15.

### 1.1.4. Politique de sécurité

Les protections offertes par IPsec sont basées sur des choix définis dans une "base de données de politique de sécurité" (*Security Policy Database, SPD*). Cette base de données est établie et maintenue par un utilisateur, un administrateur système ou une application mise en place par ceux-ci. Elle **permet de décider, pour chaque paquet, s'il se verra apporter des services de sécurité, s'il sera autorisé à passer outre ou sera rejeté**.

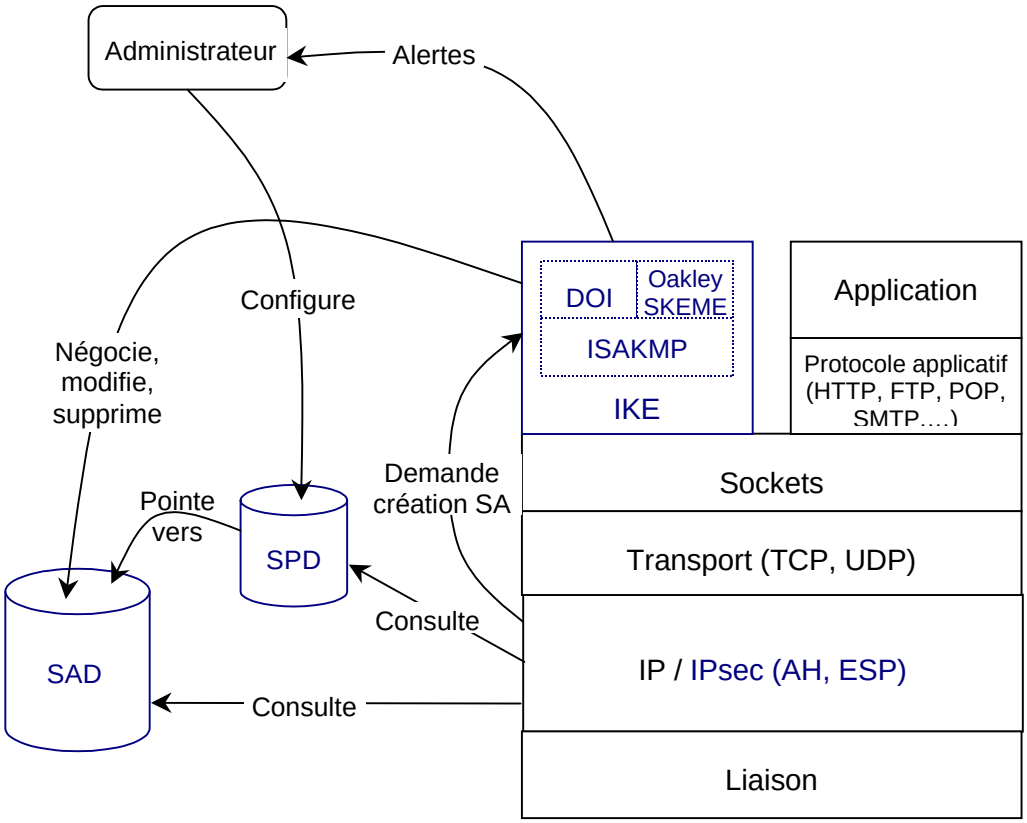
La SPD contient une liste ordonnée de règles, chaque règle comportant un certain nombre de critères qui permettent de déterminer quelle partie du trafic est concernée. **Les critères utilisables sont**

**l'ensemble des informations disponibles par le biais des en-têtes des couches IP et transport**. Ils permettent de définir la granularité selon laquelle les services de sécurité sont applicables et influencent directement le nombre de SA correspondante. Dans le cas où le trafic correspondant à une règle doit se voir attribuer des services de sécurité, la règle indique les caractéristiques de la SA (ou paquet de SA) correspondante : protocole(s), modes, algorithmes requis…

## *1.2. Principe de fonctionnement*

Le schéma ci-dessous représente tous les éléments présentés ci-dessus (en bleu), leurs positions et leurs interactions.




*- Figure 1. Organisation de IPsec -*

On distingue deux situations :

- Trafic sortant
  Lorsque la "couche" IPsec reçoit des données à envoyer, elle commence par consulter la base de donnée des politiques de sécurité (SPD) pour savoir comment traiter ces données. Si cette base lui indique que le trafic doit se voir appliquer des mécanismes de sécurité, elle récupère les caractéristiques requises pour la SA correspondante et va consulter la base des SA (SAD). Si la SA nécessaire existe déjà, elle est utilisée pour traiter le trafic en question. Dans le cas contraire, IPsec fait appel à IKE pour établir une nouvelle SA avec les caractéristiques requises.

- Trafic entrant
  Lorsque la couche IPsec reçoit un paquet en provenance du réseau, elle examine l'en-tête pour savoir si ce paquet s'est vu appliquer un ou plusieurs services IPsec et si oui quelles sont les références de la SA. Elle consulte alors la SAD pour connaître les paramètres à utiliser pour la vérification et/ou le déchiffrement du paquet. Une fois le paquet vérifié et/ou déchiffré, la SPD est consultée pour savoir si l'association de sécurité appliquée au paquet correspondait bien à celle requise par les politiques de sécurité.

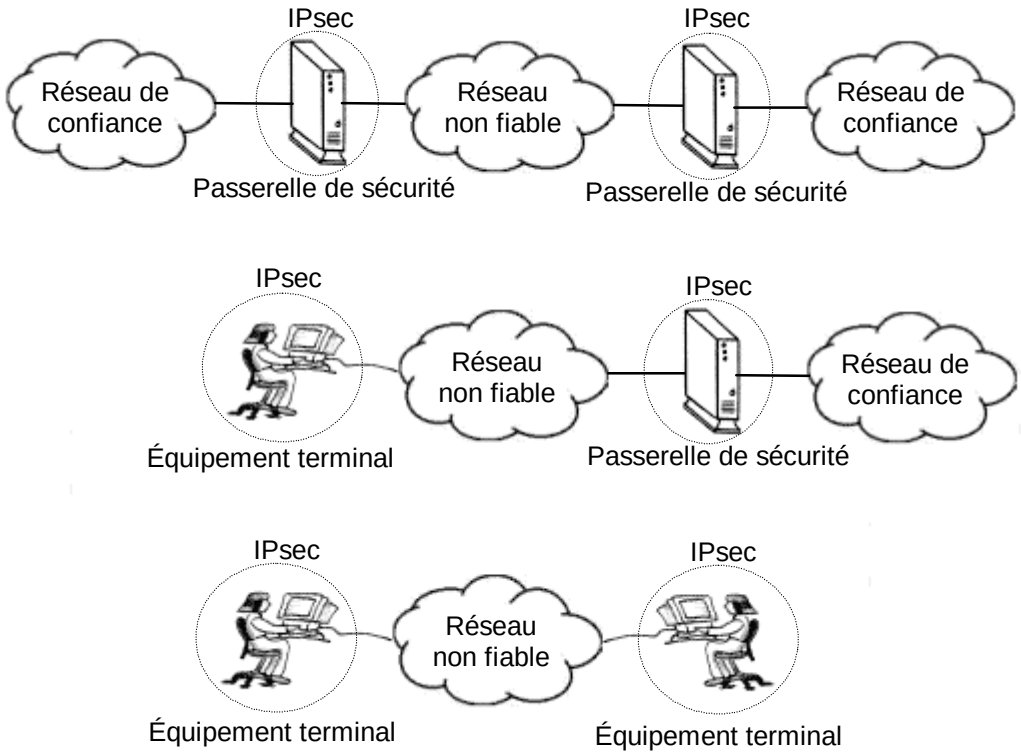
Dans le cas où le paquet reçu est un paquet IP classique, la SPD permet de savoir s'il a néanmoins le droit de passer. Par exemple, les paquets IKE sont une exception. Ils sont traités par IKE, qui peut envoyer des alertes administratives en cas de tentative de connexion infructueuse.

## *1.3. Types d'utilisations possibles*

Après avoir vu comment est constitué IPsec et comment il fonctionne, nous allons maintenant nous intéresser aux différentes façons de l'utiliser. Un point important à retenir est que le fait d'intervenir au niveau réseau rend la sécurisation **totalement transparente pour les applications**.

### 1.3.1. Équipement fournissant IPsec

IPsec peut être utilisé au niveau d'**équipements terminaux** ou au niveau de **passerelles de sécurité** (*security gateway*), permettant ainsi des approches de sécurisation lien par lien comme de bout en bout. Trois configurations de base sont possibles :



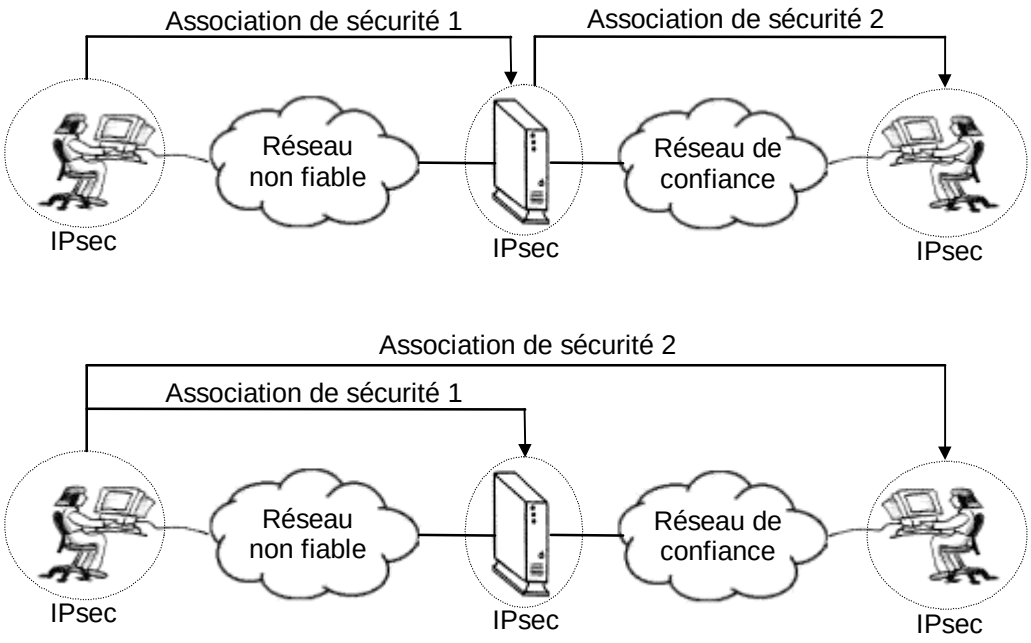

*- Figure 2. Différentes configurations possibles suivant l'équipement mettant IPsec en œuvre -*

La première situation est celle où l'on désire relier des réseaux privés distants par l'intermédiaire d'un réseau non fiable, typiquement Internet. Les deux passerelles de sécurité permettent ici d'établir un **réseau privé virtuel** (en anglais *Virtual Private Network, VPN*).

La deuxième situation correspond au cas où l'on **désire fournir un accès sécurisé au réseau interne pour des postes nomades**. Le réseau non fiable peut être Internet, le réseau téléphonique…

Enfin, dans la troisième situation, deux tiers désirent communiquer de façon sécurisée mais n'ont aucune confiance dans le réseau qui les sépare.

On peut également imaginer des configurations plus complexes où plusieurs associations de sécurité, apportant éventuellement des services de sécurité différents, se succéderaient ou se superposeraient partiellement :




*- Figure 3. Exemples de double utilisation d'IPsec -*

Dans les exemples ci-dessus, la première association peut servir à assurer les services de sécurité requis par la politique de sécurité externe (authentification et confidentialité par exemple), et la seconde à assurer les services requis par la politique de sécurité interne (authentification vis-à-vis de l'hôte final par exemple).

### 1.3.2. Modes de fonctionnement

Pour chacun des mécanismes de sécurité d'IPsec, il existe deux modes : le mode transport et le mode tunnel.

**En mode transport, seules les données en provenance du protocole de niveau supérieur et transportées par le datagramme IP sont protégées**. Ce mode n'est utilisable que sur des équipements terminaux ; en effet, en cas d'utilisation sur des équipements intermédiaires, on courrait le risque, suivant les aléas du routage, que le paquet atteigne sa destination finale sans avoir traversé la passerelle sensée le déchiffrer.

**En mode tunnel, l'en-tête IP est également protégé** (authentification, intégrité et/ou confidentialité) et remplacé par un nouvel en-tête. Ce nouvel en-tête sert à transporter le paquet jusqu'à la fin du tunnel, où l'en-tête original est rétabli. Le mode tunnel est donc utilisable à la fois sur des équipements terminaux et sur des passerelles de sécurité. Ce mode permet d'assurer une **protection plus importante contre l'analyse du trafic**, car il masque les adresses de l'expéditeur et du destinataire final.

### *Résumé et bibliographie*

Ipsec est un système de sécurisation des échanges au niveau IP qui repose sur deux mécanismes (ou protocoles), AH (*Authentication Header*) et ESP (*Encapsulating Security Payload*). Les paramètres nécessaires à l'utilisation de ces protocoles sont gérés à l'aide d'associations de sécurité (*Security Association, SA*), une association regroupant les paramètres servant à protéger une partie donnée du trafic. Les SA sont stockées dans la base de donnée des associations de sécurité (*Security Association Database, SAD*) et gérées à l'aide du protocole IKE (*Internet Key Exchange*). Les protections offertes par IPsec sont basées sur des choix définis dans la base de données de politique de sécurité (*Security Policy Database, SPD*). Cette liste de règles permet de décider, pour chaque paquet, s'il se verra apporter des services de sécurité, s'il sera autorisé à passer outre ou sera rejeté.

IPsec peut être utilisé au niveau d'équipements terminaux ou au niveau de passerelles de sécurité, permettant ainsi des approches de sécurisation lien par lien comme de bout en bout. IPsec peut donc être utilisé, notamment, pour la création de réseaux privés virtuels ou la sécurisation des accès distants. Enfin, IPsec comporte deux modes, le mode transport qui protège juste les données transportées et le mode tunnel qui protège en plus l'en-tête IP.

Le document de base pour comprendre le fonctionnement d'IPsec et le document intitulé "*Security Architecture for the Internet Protocol*", dont la version la plus récente est la [RFC 2401].

# 2. Les mécanismes de sécurité : AH et ESP

Ainsi que nous l'avons déjà mentionné dans le chapitre précédent, deux mécanismes de base permettent d'assurer l'ensemble des fonctions de sécurité fournies par IPsec. Ce sont :

- *Authentication Header* (AH), qui est conçu pour assurer **l'intégrité et l'authentification** des datagrammes IP sans chiffrement des données (i.e. sans confidentialité).
- *Encapsulating Security Payload* (ESP), dont le rôle premier est d'assurer la **confidentialité**.

Ces mécanismes peuvent être utilisés seuls ou combinés pour obtenir les fonctions de sécurité désirées.

Les mécanismes IPsec ne sont **liés à aucun algorithme cryptographique spécifique**, donc ces algorithmes peuvent être modifiés sans affecter les autres éléments d'une implémentation. Pour assurer l'interopérabilité, des algorithmes cryptographiques par défaut sont toutefois indiqués. Par ailleurs, les propriétés de l'algorithme utilisé influenceront les fonctions de sécurité fournies.

Nous allons commencer par rappeler les quelques principes relatifs aux services et mécanismes de sécurité dans le cadre de la protection des échanges, avant de détailler le fonctionnement des protocoles AH et ESP.

## *2.1. Rappels sur les services de sécurité et les mécanismes associés*

### 2.1.1. Confidentialité

La confidentialité est un service de sécurité qui consiste à s'assurer **que seules les personnes autorisées peuvent prendre connaissance d'un ensemble de données**. Le mécanisme qui permet d'obtenir ce service est généralement le **chiffrement des données** concernées à l'aide d'un algorithme cryptographique. Dans le cadre du chiffrement d'échanges réseau, on utilise toujours, pour des raisons de performance, des algorithmes de chiffrement symétriques.

Si seules les données transportées sont chiffrées, un espion peut tout de même observer des caractéristiques extérieures du trafic transitant sur un réseau afin de tenter d'en tirer des informations : fréquence des transmissions, identités des tiers communicants, quantités de données transférées. Associées à des informations de nature différente (date de rendez-vous, actualité...) ces éléments peuvent permettre aux adversaires de faire des déductions intéressantes. On parle de

**protection contre l'analyse du trafic** lorsqu'on tente d'empêcher l'analyse du trafic en cachant les adresses source et destination, la taille des paquets, la fréquence des échanges...

### 2.1.2. Authentification et intégrité

On distingue deux types d'authentification :

- L'**authentification d'un tiers** est l'action qui consiste à prouver son identité.
  Ce service est généralement rendu par l'utilisation d'un "échange d'authentification" qui implique un certain dialogue entre les tiers communiquants.
- L'**authentification de l'origine des données** sert à prouver que les données reçues ont bien été émises par l'émetteur déclaré.

L'intégrité est un service de sécurité qui consiste à s'assurer que seules les personnes autorisées pourront modifier un ensemble de données. Dans le cadre de communications, ce service consiste à permettre la détection de l'altération des données durant le transfert. On distingue deux types d'intégrité :

- L'**intégrité en mode non connecté** permet de détecter des modifications sur un datagramme individuel, mais pas sur l'ordre des datagrammes.
- L'**intégrité en mode connecté** permet en plus de détecter la perte de paquets ou leur réordonnancement.

Lorsque l'on communique avec une autre personne au travers d'un canal peu sûr, on aimerait que **le destinataire puisse s'assurer que le message émane bien de l'auteur auquel il est attribué et qu'il n'a pas été altéré pendant le transfert**. Les services correspondant sont l'authentification de l'origine des données et l'intégrité.

Les **fonctions de hachage à sens unique** permettent d'assurer **l'intégrité des données** : appliquée à un ensemble de données, une telle fonction génère un bloc de taille plus petite appelée empreinte. Toute modification des données entraîne une modification de l'empreinte, et il est très difficile de générer un message ayant la même empreinte que l'original. Si l'on dispose d'un canal sûr (mais plus coûteux) en parallèle du canal de communication normal, on peut communiquer l'empreinte des messages par l'intermédiaire de ce canal. À la réception, le destinataire recalcule l'empreinte et la compare à l'original pour vérifier que les données n'ont pas été modifiées.

Sans canal sûr, le problème se complique : si l'on transfère l'empreinte sur un canal de communication non sûr, un intercepteur peut modifier les données puis recalculer l'empreinte. Il convient donc de trouver une méthode pour s'assurer que seul l'expéditeur est capable de calculer l'empreinte. Pour cela, on peut utiliser, par exemple, une fonction de hachage à sens unique qui fonctionne de plus avec une clef secrète ou privée. On remarquera que, ce faisant, on fournit également l'authentification de l'origine des données. Inversement, si on désire fournir l'authentification de l'origine des données et que l'on utilise pour cela un moyen qui ne garantit pas l'intégrité des données authentifiées, un intrus peut modifier le message et donc faire accepter comme authentifiées des données qu'il a choisies. C'est pourquoi **intégrité et authentification de l'origine des données sont généralement fournies conjointement par un même mécanisme**. Le terme "authenticité" désigne l'intégrité jointe à l'authentification des données. Pas abus de langage, le terme "authentification" est également couramment utilisé pour désigner en fait authentification et intégrité.

Les deux mécanismes permettant d'assurer l'authenticité des données transmises sont le scellement et la signature.

Le **scellement** consiste à adjoindre au message un sceau ou code d'authentification de message (*Message Authentication Code*, MAC), qui est le résultat d'une fonction de hachage à sens unique à clef secrète. L'empreinte dépend à la fois des données et de la clef ; elle n'est donc calculable que par les personnes connaissant la clef.

La **signature numérique** assure également l'authenticité des données et fournit en plus la non-répudiation : l'émetteur ne peut pas nier avoir émis un message qu'il a signé. Ce dernier point

différencie la signature des codes d'authentification de message, et a pour conséquence que la plupart des algorithmes de signature utilisent la cryptographie à clef publique. Dans le cadre de la protection d'échanges réseau, on utilise toujours, pour des raisons de performance, un mécanisme de scellement.

Enfin, la **protection contre le rejeu** est une forme partielle d'intégrité en mode connecté qui permet de contrer les attaques au cours desquelles un adversaire envoie un message intercepté précédemment, en espérant qu'il sera accepté comme valide par le destinataire. Elle est généralement assurée par l'utilisation d'un numéro de séquence.

### *Résumé et bibliographie*

Dans le cadre de la protection des échanges réseau, les principaux services de sécurité sont :

- La confidentialité, qui est assurée par le chiffrement des données.
- L'authenticité (authentification + intégrité), qui est assurée par l'ajout d'un code d'authentification de message à chaque paquet transféré.

Les différents services et mécanismes intervenant dans la sécurité des réseaux sont référencés de façon formelle dans la norme [ISO 7498-2].

Pour plus de détails sur les notions cryptographiques mentionnées ici, on pourra se référer à la bibliographie sur la cryptologie.

## *2.2. Authentication Header (AH)*

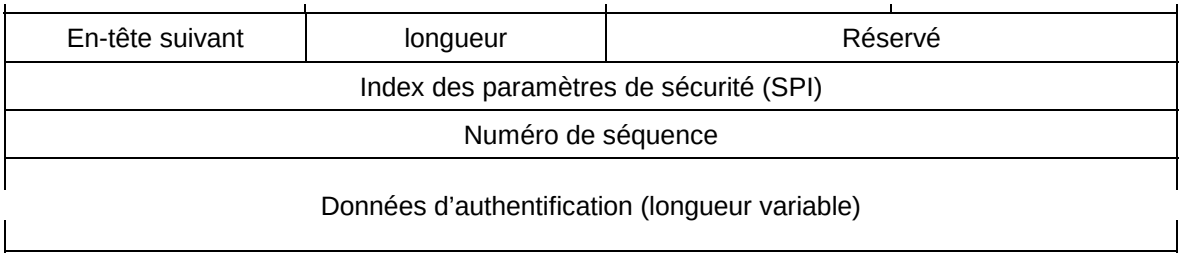
AH assure **l'intégrité des données en mode non connecté, l'authentification de l'origine des données et, de façon optionnelle, la protection contre le rejeu**.

L'absence de confidentialité dans AH permet de s'assurer que ce standard pourra être largement répandu sur l'Internet, y compris dans les endroits où l'exportation, l'importation ou l'utilisation du chiffrement dans des buts de confidentialité est restreint par la loi. Cela constitue l'une des raisons de l'utilisation de deux mécanismes distincts.

Dans AH, **intégrité et authentification sont fournies ensembles, à l'aide d'un bloc de données supplémentaire adjoint au message** à protéger. Ce bloc de données est appelé "valeur de vérification d'intégrité" (*Integrity Check Value, ICV)*,terme générique pour désigner soit un code d'authentification de message (*Message Authentication Code, MAC*), soit une signature numérique. Pour des raisons de performances, les algorithmes proposés actuellement sont tous des algorithmes de scellement (code d'authentification de message et non signature).

La protection contre le **rejeu** se fait grâce à un **numéro de séquence** ; elle n'est disponible que si IKE est utilisé, car en mode manuel aucune "ouverture de connexion" ne permet d'initialiser le compteur.

Voici l'organisation de l'en-tête d'authentification :



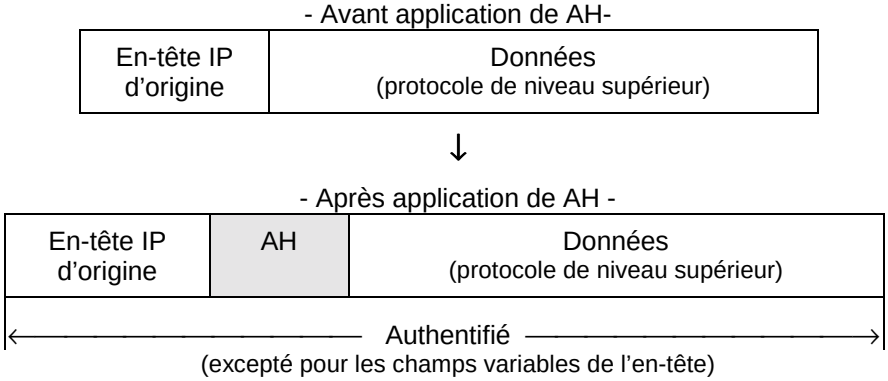

*- Figure 4. Format de AH -*

**L'expéditeur calcule les données d'authentification à partir de l'ensemble des champs invariants du datagramme IP final**, AH compris, ce qui permet d'étendre l'authentification au SPI et au numéro de séquence notamment. Les champs variables (TTL, routage...) et le champ destiné à recevoir les données d'authentification sont considérés comme égaux à zéro pour le calcul. Les

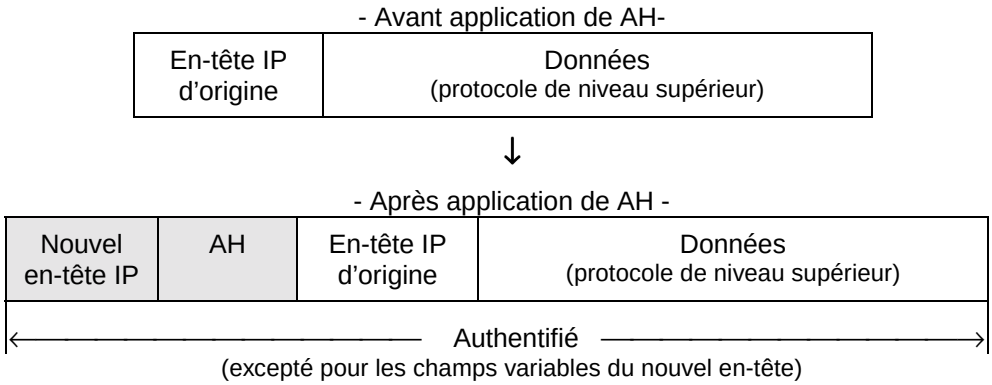
 

données d'authentification sont alors adjointes au paquet IP par le biais de l'en-tête d'authentification (AH). Le récepteur vérifie l'exactitude de ces données à la réception.

Les figures ci-dessous indiquent la position de AH et le service apporté en fonction du mode choisi (transport ou tunnel).




*- Figure 5. Position de AH en mode transport (IPv4) -*




*- Figure 6. Position de AH en mode tunnel (IPv4) -*

Les algorithmes par défaut que doit fournir toute réalisation de IPsec pour AH sont, au moment où ce document est rédigé, HMAC-MD5 et HMAC-SHA-1. Les autres algorithmes prévus sont KPDK-MD5, DES-MAC et HMAC-RIPE-MD.

### *Résumé et bibliographie*

AH assure l'authenticité des données transportées et de l'en-tête IP par l'ajout d'un code d'authentification de message (HMAC-MD5, HMAC-SHA-1) au paquet protégé. En configuration dynamique *via* IKE, AH peut également fournir une protection contre le rejeu.

AH est décrit dans le document intitulé "*IP Authentication Header*", dont la dernière version est la [RFC 2402].

Les algorithmes d'authentification utilisables avec AH sont listés dans le DOI IPsec [RFC 2407] et font l'objet de divers documents (voir bibliographie page 47).

## *2.3. Encapsulating Security Payload (ESP)*

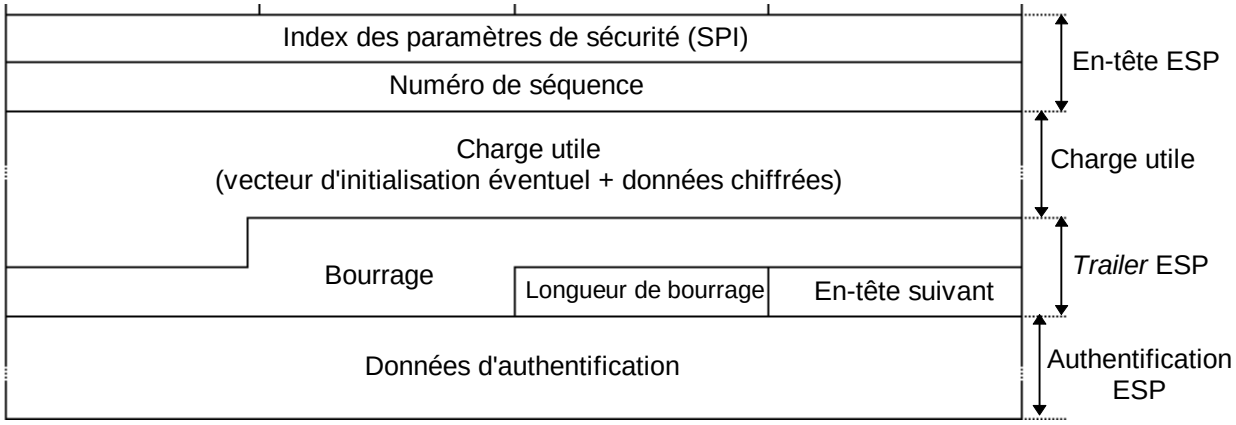
ESP peut assurer, au choix, un ou plusieurs des services suivants :

- **confidentialité** (confidentialité des données et protection partielle contre l'analyse du trafic si l'on utilise le mode tunnel),

- **intégrité** des données en mode non connecté et **authentification de l'origine** des données, protection contre le **rejeu**.

La confidentialité peut être sélectionnée indépendamment des autres services, mais son utilisation sans intégrité/authentification (directement dans ESP ou avec AH) rend le trafic vulnérable à certains types d'attaques actives qui pourraient affaiblir le service de confidentialité. Comme dans AH, authentification et intégrité sont deux services qui vont de pair et que l'on désigne souvent sous le terme "authentification" ; ils sont fournis par l'utilisation d'une **ICV** (en pratique, un MAC). La protection contre le rejeu ne peut être sélectionnée que si l'authentification l'a été et que IKE est utilisé. Elle est fournie par un **numéro de séquence** que le destinataire des paquets vérifie.

Contrairement à AH, où l'on se contentait d'ajouter un en-tête supplémentaire au paquet IP, ESP fonctionne suivant le principe de l'encapsulation : **les données originales sont chiffrées puis encapsulées** entre un en-tête et un *trailer*. Voici l'organisation de ESP :




*- Figure 7. Format de ESP -*

Le champ bourrage peut être nécessaire pour les algorithmes de chiffrement par blocs ou pour aligner le texte chiffré sur une limite de 4 octets.

Les données d'authentification ne sont présentes que si ce service a été sélectionné.

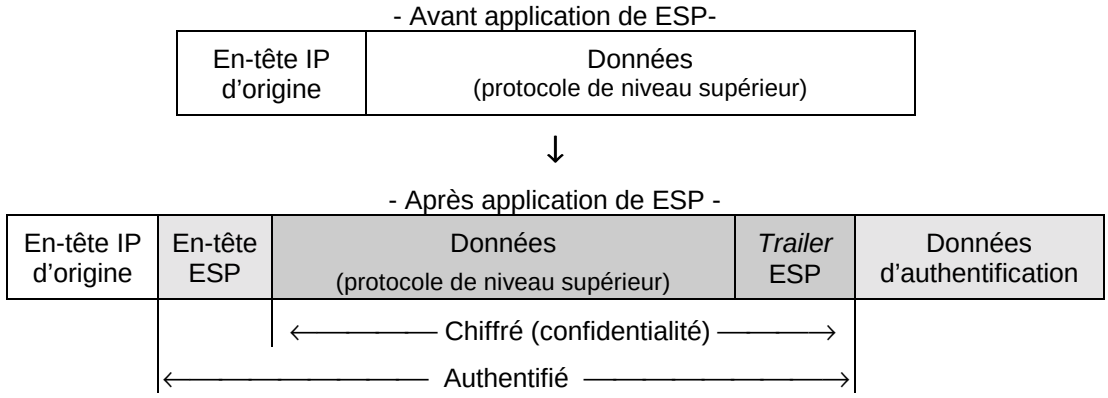
Voyons maintenant comment est appliquée la confidentialité dans ESP. L'expéditeur :

- Encapsule, dans le champ "charge utile" de ESP, les données transportées par le datagramme original et éventuellement l'en-tête IP (mode tunnel).
- Ajoute si nécessaire un bourrage.
- Chiffre le résultat (données, bourrage, champs longueur et en-tête suivant).
- Ajoute éventuellement des données de synchronisation cryptographiques (vecteur d'initialisation) au début du champ "charge utile".

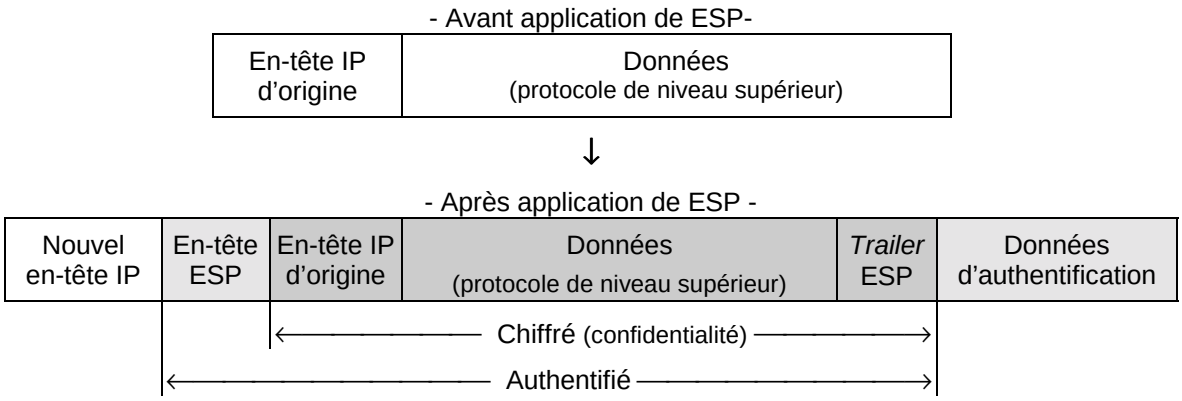
Si elle a été sélectionnée, **l'authentification est toujours appliquée après que les données ne soient chiffrées**. Cela **permet, à la réception, de vérifier la validité du datagramme avant de se lancer dans la coûteuse tâche de déchiffrement**. Contrairement à AH, l'authentification dans ESP est appliquée uniquement sur le "paquet" (en-tête + charge utile + *trailer*) ESP et n'inclut ni l'en-tête IP ni le champ d'authentification.

Les figures ci-dessous indiquent la position de ESP et les services apportés en fonction du mode choisi (transport ou tunnel).




*- Figure 8. Position de ESP en mode transport (IPv4) -*




*- Figure 9. Position de ESP en mode tunnel (IPv4) -*

Les algorithmes prévus pour être utilisés avec ESP sont, au moment où ce document est rédigé :

- Confidentialité : DES triple (obligatoire), DES, RC5, CAST, IDEA, IDEA triple, Blowfish, RC4 et NULL pour le cas ou le chiffrement n'est pas souhaité.
- Authentification : HMAC-MD5 (obligatoire), HMAC-SHA-1 (obligatoire), DES-MAC, HMAC-RIPE-MD, KPDK-MD5 et NULL pour le cas où l'authenticité n'est pas sélectionnée.

### *Résumé et bibliographie*

ESP peut assurer, au choix, la confidentialité et/ou l'authenticité des données.

ESP est décrit dans le document intitulé "*IP Encapsulating Security Payload (ESP)*", dont la dernière version est la [RFC 2406].

Les algorithmes d'authentification utilisables avec ESP sont décrits dans les mêmes documents que pour AH.

Les algorithmes de chiffrement utilisables avec ESP sont listés dans le DOI IPsec [RFC 2407] et font l'objet de divers documents (voir bibliographie page 47).

# 3. La gestion des clefs

Les protocoles sécurisés présentés dans le chapitre précédent ont recours à des algorithmes cryptographiques et ont donc besoin de clefs. Un des problèmes fondamentaux d'utilisation de la cryptographie est la gestion de ces clefs. Le terme "gestion" recouvre la génération, la distribution, le stockage et la suppression des clefs.

## *3.1. Concepts généraux relatifs à la gestion des clefs*

Ce paragraphe a pour but de présenter un certain nombre de concepts utiles pour la compréhension des parties de cette présentation relatives à la gestion des clefs.

### 3.1.1. Types de clefs

On peut définir plusieurs types de clefs suivant leurs rôles :

- **Clefs de chiffrement de clefs**
  Ces clefs **servent exclusivement à chiffrer d'autres clefs** et ont généralement une durée de vie longue. On notera que leur utilisation, restreinte au chiffrement de valeurs aléatoires que sont des clefs, rend les attaques par cryptanalyse plus difficiles à leur niveau. La cryptographie à clef publique est souvent utilisée pour le transport de clefs en chiffrant la clef à transporter à l'aide d'une clef publique.
- **Clefs maîtresses**
  Les clefs maîtresses sont des clefs qui ne servent pas à chiffrer mais uniquement à **générer d'autres clefs par dérivation**. Une clef maîtresse peut ainsi être utilisée, par exemple, pour générer deux clefs : une pour le chiffrement et une pour l'authentification.
- **Clefs de session ou de chiffrement de données**
  D'une **durée d'utilisation généralement faible** (elle peut aller jusqu'à changer à chaque message), une telle clef, par opposition aux précédentes, sert à chiffrer des données. Du fait de leur lenteur, les algorithmes asymétriques sont très peu utilisés en chiffrement de données, et **les clefs de session sont donc généralement des clefs secrètes**.

Il est à noter que des abus de langage ont souvent lieu avec ces termes. Ainsi, on parle parfois de clef de session pour désigner en fait une clef maîtresse de même durée de vie et servant à générer plusieurs clefs : une pour l'authentification, une pour le chiffrement...

### 3.1.2. Infrastructures à clef publique

De nombreux protocoles et applications utilisent la cryptographie à clef publique à grande échelle et doivent donc pouvoir gérer des listes importantes de clefs publiques pour des systèmes distribués. Pour cela, ils ont recours à **des infrastructures à clés publiques, systèmes de gestion des clés publiques prévus pour une utilisation à grande échelle.**

La plupart de ces systèmes se basent sur des autorités de certification (*Certificate Authorities*, CA) qui sont habilitées à délivrer des certificats. Plus précisément, elles vérifient et authentifient des clés publiques. Ces autorités sont généralement organisées en une hiérarchie qui permet une plus grande souplesse pour la gestion des clés publiques par le biais de certificats signés par ces autorités et de listes de révocations de certificats (*Certificate Revocation Lists*, CRL).

Il existe de nombreuses PKI, la plupart en cours d'évolution. Des exemples d'infrastructures à clef publiques sont PKIX (*Public-Key Infrastructure X.509*) et SPKI (*Simple Public Key Infrastructure*) et DNSSEC (*Domain Name System Security*), qui font toutes trois l'objet d'une normalisation à l'IETF.

### 3.1.3. Échange de clefs et authentification

Pour établir une communication sécurisée, on procède généralement, en premier lieu, à une authentification à des fins de contrôle d'accès. Puis, un échange de clef permet l'utilisation d'un mécanisme de sécurisation des échanges : l'authentification est ainsi étendue à la suite de la communication. Il va de soi que l'échange de clef doit nécessairement être authentifié. La combinaison de l'authentification et de l'échange de clef prend la forme d'un échange de messages appelé **protocole d'authentification mutuelle avec échange de clef**.

### 3.1.4. Propriétés des protocoles d'échange de clef

DIFFIE, VAN OORSCHOT et WIENER définissent, dans [DOW92], la notion de protocole d'authentification mutuelle avec échange de clef sûr. Un protocole est dit sûr si les deux conditions suivantes sont valables dans chaque instance du protocole où l'un des deux tiers, par exemple Alice, exécute le protocole honnêtement et accepte l'identité de l'autre tiers :

- Au moment où Alice accepte l'identité de Bob, les enregistrements des messages échangés par les deux tiers se correspondent (i.e. les messages n'ont pas été altérés en route).
- Il est matériellement impossible pour toute personne autre que les tiers en présence de retrouver la clef échangée.

La propriété évoquée ci-dessus représente le minimum requis pour tout protocole. Cependant, d'autres propriétés des protocoles d'échange de clef peuvent être souhaitables :

- La propriété dite de ***Perfect Forward Secrecy*** *(PFS)* est garantie si **la découverte par un adversaire du ou des secrets à long terme ne compromet pas les clefs de session** générées précédemment : les clefs de session passées ne pourront pas être retrouvées à partir des secrets à long terme. On considère généralement que cette propriété assure également que **la découverte d'une clef de session ne compromet ni les secrets à long terme ni les autres clefs de session**. La propriété de *Perfect Forward Secrecy* peut être fournie, par exemple, par la génération des clefs de session au moyen du protocole Diffie-Hellman (voir ci-dessous), où les exponentiels Diffie-Hellman sont des valeurs à court terme.

- La propriété dite de ***Back Traffic Protection*** est fournie si la génération de chaque clef de session se fait de manière indépendante : les nouvelles clefs ne dépendent pas des clefs précédentes et **la découverte d'une clef de session ne permet ni de retrouver les clefs de session passées ni d'en déduire les clefs à venir**.

- On dit qu'il y a **authentification directe** (*Direct Authentication*) si, **à la fin du protocole, les valeurs servant à générer le secret partagé sont authentifiées ou si chaque tiers a prouvé qu'il connaissait la clef de session**. Par opposition, l'authentification est dite indirecte (Indirect authentication) si elle n'est pas garantie à la fin du protocole, mais dépend de la capacité de chaque tiers à utiliser, dans la suite des échanges, la ou les clefs mises en place précédemment.

- On parle de **protection de l'identité** (*Identity Protection*) lorsque le protocole garantit qu'**un attaquant espionnant les échanges ne pourra pas connaître les identités des tiers communicants**.

- Enfin, l'utilisation du temps (*Timestamps*) afin d'éviter la rejouabilité est très controversée du fait, principalement, de sa trop grande dépendance d'horloges synchronisées.

### 3.1.5. Diffie-Hellman

Inventé en 1976 par DIFFIE et HELLMAN, ce protocole permet à deux tiers de **générer un secret partagé sans avoir aucune information préalable l'un sur l'autre**. Il est basé sur la cryptologie à clef publique (dont il est d'ailleurs à l'origine), car il fait intervenir des valeurs publiques et des valeurs privées. Sa sécurité dépend de la difficulté de calculer des logarithmes discrets sur un corps fini. Le secret généré à l'aide de ce protocole peut ensuite être utilisé pour dériver une ou plusieurs clefs (clef secrète, clef de chiffrement de clefs...).

Voici le déroulement du protocole :
1. Alice et Bob se mettent d'accord sur un grand entier *n* tel que *(n-1)/2* soit premier et sur un entier *g* primitif par rapport à *n*. Ces deux entiers sont publics.
2. Alice choisit de manière aléatoire un grand nombre entier *a*, qu'elle garde secret, et calcule sa valeur publique, $A = g^a \bmod n$. Bob fait de même et génère *b* et $B = g^b \bmod n$.
3. Alice envoie *A* à Bob ; Bob envoie *B* à Alice.
4. Alice calcule $K_{AB} = B^a \bmod n$ ; Bob calcule $K_{BA} = A^b \bmod n$. $K_{AB} = K_{BA} = g^{ab} \bmod n$ est le secret partagé par Alice et Bob.

Une personne qui écoute la communication connaît *g*, *n*, $A=g^a \bmod n$ et $B=g^b \bmod n$, ce qui ne lui permet pas de calculer $g^{ab} \bmod n$ : il lui faudrait pour cela calculer le logarithme de *A* ou *B* pour retrouver *a* ou *b*.

En revanche, **DIFFIE–HELLMAN est vulnérable à l'attaque active dite attaque de l'intercepteur.**

Le principe de l'attaque de l'intercepteur (*man-in-the-middle attack*) est que, pendant que deux tiers procèdent à un échange de clef, en utilisant un protocole du type DIFFIE–HELLMAN par exemple, **un adversaire se positionne entre les deux tiers et intercepte les échanges**. De cette façon, il procède à un échange de clef avec chaque tiers. À la fin du protocole, chaque tiers utilisera donc une clef différente, chacune de ces clefs étant connue de l'intercepteur. Pour chaque message transmis par la suite, l'intercepteur procédera à son déchiffrement avec la clef correspondante puis le rechiffrera avec l'autre clef avant de l'envoyer à son destinataire : les deux tiers croiront communiquer de façon sûre alors que l'intercepteur pourra en fait lire tous les messages, voire même forger de faux messages.

Voici comment se déroule cette attaque dans le cas de DIFFIE–HELLMAN :
1. Alice envoie sa valeur publique $A = g^a \bmod n$ à Bob ; Ingrid l'intercepteur remplace cette valeur publique par la sienne. Bob reçoit donc $I = g^i \bmod n$.
2. Bob envoie sa valeur publique à Alice ; Ingrid remplace là aussi cette valeur par la sienne.
3. Alice génère le "secret" $K_{AI} = I^a \bmod n$. Ingrid génère le même secret en calculant $A^i \bmod n$.
4. Bob génère le "secret" $K_{BI} = I^b \bmod n$. Ingrid génère le même secret en calculant $B^i \bmod n$.

Alice et Bob croient tous les deux être en possession d'un secret partagé, mais en fait chacun d'eux partage un secret différent avec Ingrid.

Une façon de contourner le problème de l'attaque de l'intercepteur est d'**authentifier les valeurs publiques utilisées pour la génération du secret partagé**. Cela peut se faire à deux niveaux :
- En utilisant des valeurs publiques authentifiées, à l'aide de certificats par exemple. Cette méthode est notamment à la base du protocole SKIP.
- En authentifiant les valeurs publiques après les avoir échangées, en les signant par exemple. Cette méthode est utilisée entre autres par le protocole Photuris.

L'inconvénient, dans les deux cas, est que l'on perd un gros avantage de DIFFIE–HELLMAN, qui est la possibilité de générer un secret partagé sans aucune information préalable sur l'interlocuteur. Mais, si les valeurs publiques sont de courtes durée, Diffie-Hellman authentifié fournit la propriété de *perfect formard secrecy*.

### *Bibliographie*

Pour plus de détails sur la sécurité des échanges, on pourra consulter [OPP98].

Pour plus de détails sur les notions cryptographiques mentionnées ici, on pourra se référer à la bibliographie sur la cryptologie page 45.

## *3.2. Les protocoles d'authentification mutuelle avec échange de clef développés pour IP*

Il existe de nombreux protocoles d'authentification mutuelle avec échange de clef, qui se différencient suivant les prérequis qu'ils imposent (secret partagé préalable, infrastructure à clef publique...) et les propriétés qu'ils vérifient (*direct authentication*, *perfect forward secrecy*...).

Dans le cadre des protocoles d'échange de clef développés pour la sécurisation des échanges sous IP, une distinction supplémentaire s'impose entre les protocoles orientés connexion et ceux sans connexion. Dans le premier cas, **on a recours à un protocole d'établissement de clef de session authentifiée, hors bande, avant la communication**. La clef résultante est ensuite utilisée pour sécuriser le trafic IP. L'inconvénient de cette approche est qu'elle **nécessite l'établissement et la gestion d'une pseudo couche session sous IP**, alors qu'IP est un protocole sans connexion. Dans le second cas, on a recours à une gestion des clefs sans état (*stateless*), qui ne nécessite donc **pas de connexion**. Ceci est réalisable à travers un système en bande, où **la clef ayant servi à chiffrer le paquet est transmise avec celui-ci**, chiffrée avec la clef publique du destinataire par exemple. L'inconvénient de ce système est qu'il ajoute des données à chaque paquet transmis.

D'autre part, DIFFIE–HELLMAN est très utilisé dans tous les protocoles présentés ici, les différences venant de la durée de vie des valeurs publiques utilisées et de la façon dont elles sont authentifiées et échangées.

### 3.2.1. SKIP

SKIP (*Simple Key management for Internet Protocols*) est un exemple de **protocole qui ne se base pas sur l'établissement d'une "connexion"**. En effet, aucun échange préalable de messages n'est nécessaire avant de pouvoir envoyer un paquet chiffré et **chaque paquet transporte l'information qui permettra de le déchiffrer**. Au niveau des couches réseau, cela se traduit par le fait que SKIP se situe au niveau IP, et non au-dessus de TCP ou UDP comme la plupart des protocoles de gestion de clefs.

D'autre part, SKIP se base sur une génération de secret partagé DIFFIE–HELLMAN avec valeurs publiques authentifiées, donc le seul prérequis est que **chaque participant doit être en possession d'une valeur publique DIFFIE–HELLMAN authentifiée**.

#### a/ Historique

SKIP a été créé en 1994 par Ashar AZIZ et Whitfield DIFFIE de SUN MICROSYSTEMS. Un brevet fut déposé par SUN MICROSYSTEMS en juin 1994 puis placé dans le domaine public peu de temps après. SKIP fut proposé comme protocole de gestion des clefs standard pour IPsec, et un certain nombre *d'Internet drafts* furent publiés dans ce sens jusqu'en août 1996. À cette époque, les deux standards possibles pour la gestion des clefs avec IPsec étaient SKIP et ISAKMP/Oakley. Un choix s'imposait, et la question fut tranchée en faveur de ISAKMP/Oakley en septembre 1996. Si ISAKMP/Oakley a été choisi pour être le protocole imposé dans toute implémentation, l'utilisation de SKIP n'est pas exclue. Cependant, la publication d'*Internet drafts* à son sujet a cessé depuis cette date. SUN MICROSYSTEMS continue à développer ce protocole et à l'intégrer dans un certain nombre de produits, notamment *SunScreen SKIP*.SKIP est également utilisable pour la gestion des clefs IPsec dans le produit *Firewall-1* de CHECK POINT.

#### b/ Principe

SKIP est basé sur le principe de génération de secret partagé DIFFIE–HELLMAN, avec authentification pour éviter une possible attaque de l'intercepteur. **Les deux tiers possédant chacun une valeur publique DIFFIE–HELLMAN authentifiée, ils peuvent, à partir de la connaissance de la valeur publique de l'interlocuteur et de leur propre valeur privée, générer un secret partagé**.

Pour implémenter SKIP, chaque tiers doit donc posséder une valeur publique DIFFIE–HELLMAN authentifiée. Cette authentification peut être obtenue de différentes façons : certificat X.509, DNS sécurisé, clef PGP signée... D'autre part, pour communiquer avec un interlocuteur choisi, un tiers doit obtenir sa valeur publique. Une façon de réaliser cela est de distribuer les valeurs publiques à l'aide d'un "service d'annuaire" (*directory service*) ou à l'aide du "protocole de découverte de certificats" (*Certificate Discovery Protocol, CDP*).

Soient I et J les deux tiers. Les valeurs privées sont respectivement *i* et *j* et les valeurs publiques $g^i \bmod p$ et $g^j \bmod p$. Chaque tiers obtient le secret partagé en élevant la valeur publique de son interlocuteur à la puissance sa valeur privée : $(g^j \bmod p)^i = (g^i \bmod p)^j$. $g^{ij} \bmod p$ est appelé **secret partagé à long terme et sert à dériver une clef secrète $K_{ij}$.** En effet, $g^{ij} \bmod p$ sera typiquement de longueur 1024 bits ou plus, alors que $K_{ij}$ est une clef secrète de longueur 40 à 256 bits typiquement. Dans SKIP, $K_{ij}$ est constituée des bits de poids faible de $g^{ij} \bmod p$.

Voilà pour la partie échange de clef proprement dite. SKIP va plus loin en précisant comment cette clef est ensuite utilisée pour protéger les échanges entre I et J. $K_{ij}$ est en fait une clef de chiffrement de clef, c'est-à-dire qu'elle est utilisée exclusivement pour chiffrer des clefs de durée de vie beaucoup plus faible. En effet, **$K_{ij}$ est utilisée pour chiffrer une clef $K_p$, appelée clef de paquet, qui est elle-même utilisée pour générer deux clefs, servant respectivement au chiffrement et à l'authentification d'un paquet IP ou d'un ensemble réduit de paquets**.

Le mode de fonctionnement de SKIP, s'il ne requiert pas d'échange préalable à l'envoi de paquet chiffré, implique en revanche une augmentation de la taille de chaque paquet. En effet, un en-tête supplémentaire, dit en-tête SKIP, est adjoint au datagramme IP ; il sert notamment à transmettre la clef $K_p$ (chiffrée avec $K_{ij}$) et à indiquer les algorithmes utilisés.

### c/ Extension pour la propriété de perfect forward secrecy

Le fonctionnement exposé ci-dessus présente le défaut de ne pas fournir la propriété de *perfect forward secrecy*. En effet, **si $K_{ij}$ est découverte, l'ensemble des clefs de session $K_p$ utilisées par le passé sont compromises**. Une extension de SKIP garantissant cette sécurité supplémentaire a donc été développée. Elle se présente sous la forme d'une génération de clefs DIFFIE–HELLMAN dite éphémère, car reposant sur des valeurs publiques à court terme. Le secret partagé ainsi généré remplace le secret partagé à long terme de la version classique de SKIP.

L'utilisation d'une génération de clefs DIFFIE–HELLMAN éphémère implique l'échange de certificats contenant les valeurs publiques correspondantes. Cet échange se fait à l'aide du protocole CDP (*Certificate Discovery Protocol*) et utilise la clef $K_{ij}$. Contrairement à la version de base de SKIP, l'extension SKIP PFS requiert donc un échange de données entre les tiers avant de leur permettre de communiquer.

SKIP PFS fournit également la protection des identités, ce que ne garantissait pas SKIP.

#### *Résumé et bibliographie*

SKIP est un protocole de gestion des clefs "en ligne" spécifiquement prévu pour sécuriser des protocoles sans connexion comme IP. Il utilise une génération de clef DIFFIE–HELLMAN, à base de valeurs publiques authentifiées. Le secret partagé ainsi généré sert de clef de chiffrement de clef pour des clefs de session. Il n'y a donc pas de *perfect forward secrecy (PFS)*. Une extension de SKIP a été définie qui fournit la propriété de PFS et la protection de l'anonymat, au prix d'échange de certificats supplémentaires entre les entités.

La bibliographie sur SKIP se trouve page 48.

## 3.2.2. Photuris

### a/ Contexte

Développé depuis 1995 par Phil KARN de chez QUALCOMM et William Simpson de chez DayDreamer, Photuris utilise le même principe que le protocole STS (*Station-To-Station*) créé par DIFFIE, VAN OORSCHOT et WIENER [DOW92]. Il a fait l'objet d'*Internet drafts* puis de RFC indépendants de tout groupe de travail, et est utilisable avec IPsec. Quelques implémentations l'utilisent d'ailleurs actuellement, même si IKE est bien plus répandu.

Contrairement à SKIP, **Photuris est un protocole "orienté connexion"** au sens où il comporte un certain nombre d'échanges (pour la négociation d'options et la génération de clef) préalables à tout échange de messages chiffrés. Photuris s'est vu attribué le port UDP 468 par l'IANA.

#### b/ Principe

Photuris est basé sur la génération d'un secret partagé selon le principe de DIFFIE–HELLMAN. **Ce secret partagé a une durée de vie faible : il sert à générer les clefs de session nécessaires pour protéger la suite des échanges.** Afin de contrer l'attaque de l'intercepteur à laquelle est vulnérable DIFFIE–HELLMAN, l'échange des valeurs servant à générer le secret partagé est suivi d'une authentification de ces valeurs à l'aide des secrets à long terme. Ces secrets servant uniquement à l'authentification, Photuris fournit la propriété de *perfect forward secrecy*.

Un problème de DIFFIE–HELLMAN est que ce protocole requiert des opérations coûteuses en ressources système, ce qui le rend vulnérable à des attaques en **déni de service** appelées "attaques par inondation" (*flooding attacks*). **Afin de rendre de telles attaques plus difficiles, Photuris a recours à un échange de *cookies*** avant de procéder à l'échange de valeurs DIFFIE–HELLMAN. La valeur du *cookie* dépend des tiers en présence, en particulier par l'intermédiaire de l'adresse IP et du port UDP, ceci afin d'empêcher un attaquant d'obtenir un *cookie* valable puis de l'utiliser pour inonder la victime de requêtes provenant d'adresses IP et/ou de ports arbitraires. D'autre part, il ne doit pas être possible, pour un attaquant, de générer des *cookies* qui seront acceptés par une entité comme générés par elle-même. Ceci implique que l'entité émettrice utilise une information locale secrète dans la génération de ses *cookies* et dans leur vérification ultérieure.

Le protocole Photuris est composé des trois étapes suivantes :
1. Un **échange de *cookies*** permet de contrer certaines attaques simples en déni de service. Chaque tiers en présence génère un *cookie*, et les *cookies* sont répétés dans chaque message transmis.
2. Un **échange de valeurs publiques** pour la génération d'un secret partagé.
3. Un **échange d'identités** afin que les tiers s'identifient l'un l'autre et vérifient l'authenticité des valeurs échangées durant la phase précédente. Cet échange est protégé en confidentialité grâce à des clefs privées dérivées du secret partagé et des *cookies* entre autres.

D'autres messages peuvent être échangés ultérieurement pour changer les clefs de session ou les paramètres de sécurité. Ces messages seront protégés en confidentialité de la même façon que les messages de l'échange 3.

En parallèle aux échanges exposés ci-dessus, les tiers communicants se mettent d'accord sur la méthode de génération du secret partagé, puis sur un certain nombre de paramètres de sécurité utiles à l'association de sécurité mise en place.

#### *Résumé et bibliographie*

Photuris est un protocole orienté connexion qui utilise une génération de secret partagé DIFFIE–HELLMAN, suivie d'une authentification des valeurs publiques utilisées, pour établir une clef de session. Photuris introduit le concept de *cookie*, mécanisme qui permet de contrer certaines attaques en déni de service. Le protocole comporte trois phases : échange de *cookies*, échange de valeurs publiques et échange d'identités.

La bibliographie sur Photuris se trouve page 48.

### 3.2.3. SKEME

#### a/ Contexte

Développé spécifiquement pour IPsec, SKEME est une extension de Photuris proposée en 1996 par Hugo KRAWCZYK de l'IBM T. J. WATSON RESEARCH CENTER. Contrairement à Photuris, qui impose un déroulement précis du protocole, SKEME fournit **divers modes d'échange de clef possibles**.

### b/ Principe

De la même façon que les protocoles STS et Photuris, le mode de base de SKEME repose sur l'utilisation de **clefs publiques** et sur une génération de secret partagé **DIFFIE–HELLMAN**. SKEME n'est cependant pas restreint à l'utilisation de clefs publiques, mais **permet également l'utilisation d'une clef précédemment partagée**. Cette clef peut avoir été obtenue par distribution manuelle ou par l'intermédiaire d'un centre de distribution de clef (*Key Distribution Center*, KDC), comme pour Kerberos. Le KDC permet aux entités communiquantes de partager un secret par l'intermédiaire d'un tiers de confiance. L'utilisation de ce secret pour l'authentification du secret DIFFIE–HELLMAN et non directement en tant que clef de session diminue la confiance requise en le KDC. Enfin, SKEME permet également d'effectuer des échanges plus rapides en omettant d'utiliser DIFFIE–HELLMAN lorsque la propriété de *perfect forward secrecy* n'est pas requise.

En résumé, SKEME comporte quatre modes distincts :
- Le mode de base, qui fournit un échange de clef basé sur des clefs publiques et présentant la propriété de PFS grâce à DIFFIE–HELLMAN.
- Un échange de clef basé sur l'utilisation de clefs publiques, mais sans DIFFIE–HELLMAN.
- Un échange de clef basé sur l'utilisation d'une clef partagée précédemment et sur DIFFIE–HELLMAN.
- Un mécanisme de changement de clef rapide basé uniquement sur des algorithmes symétriques.

D'autre part, SKEME se décompose en trois phases : SHARE, EXCH et AUTH.
- Durant la **phase de partage (SHARE)**, les tiers échangent des demi-clefs, chiffrées avec la clef publique l'un de l'autre. Ces deux demi-clefs permettent de générer une clef secrète K. Si l'on désire protéger l'anonymat des tiers en présence, leur identité est également chiffrée. Dans le cas où l'on possède déjà un secret partagé, cette phase est sautée.
- La **phase d'échange (EXCH)** est utilisée, suivant le mode choisi, pour échanger des valeurs publiques DIFFIE–HELLMAN ou des nombres aléatoires. Le secret partagé DIFFIE–HELLMAN ne sera calculé qu'après la fin des échanges.
- Les valeurs publiques ou nombres aléatoires précédents sont authentifiées pendant la **phase d'authentification (AUTH)**, à l'aide de la clef secrète établie durant la phase SHARE.

Il va de soi que les messages correspondant à ces trois phases ne se suivent pas nécessairement de la façon décrite ci-dessus, mais sont en pratique combiné pour minimiser le nombre de messages à échanger.

Une autre phase, dite **phase COOKIES**, peut être ajoutée avant la phase SHARE afin de protéger contre les attaques en déni de service en ayant recours au mécanisme des *cookies* introduit par Photuris.

#### *Résumé et bibliographie*

SKEME est un protocole orienté connexion qui propose quatre modes d'échange de clefs distincts et se compose de trois phases : partage, échange et authentification. SKEME peut également utiliser le mécanisme de *cookies* de Photuris.

La bibliographie sur SKEME se trouve page 48.

## 3.2.4. Oakley

### a/ Contexte

Initialement proposé par Hilarie ORMAN du département informatique de l'université d'Arizona, Oakley a fait l'objet d'une RFC dans le cadre du groupe IPsec et est, avec ISAKMP et SKEME, à la base de l'échange de clef pour IPsec.

#### b/ Principe

Oakley est un protocole d'échange de clef qui ressemble beaucoup à SKEME, en ce sens qu'il possède également plusieurs modes, a recours aux *cookies* et ne nécessite pas le calcul du secret partagé DIFFIE–HELLMAN avant la fin du protocole. Il se distingue des protocoles présentés jusqu'à présent par le fait qu'il **permet explicitement aux tiers en présence de se mettre d'accord sur les mécanismes d'échange de clef, de chiffrement et d'authentification utilisés**.

De fait, le but d'Oakley est de permettre le partage, de façon sûre entre les tiers, d'un ensemble d'informations relatives au chiffrement : nom de la clef, clef secrète, identités des tiers, algorithmes de chiffrement, d'authentification et fonction de hachage.

Plusieurs options sont disponibles dans Oakley. En plus du classique DIFFIE–HELLMAN, Oakley peut être utilisé pour dériver une nouvelle clef d'une clef ancienne ou pour distribuer une clef en la chiffrant. Ces options se traduisent par l'existence de plusieurs modes.

Le principe général d'Oakley est que l'initiateur de l'échange commence par spécifier autant d'information qu'il le désire dans un premier message. Son interlocuteur lui répond en fournissant également autant d'information qu'il le désire. La conversation se poursuit jusqu'à ce que l'état recherché soit atteint. Le choix de la quantité d'information à inclure dans chaque message dépend des options sélectionnées (utilisation de *cookies* sans état, protection de l'identité, PFS, non-répudiation...).

Les trois composants du protocole sont :
1. échange de *cookies* (éventuellement sans état),
2. échange de valeurs publiques DIFFIE–HELLMAN (optionnel),
3. authentification (options : anonymat, PFS sur les identités, non-répudiation).

#### *Résumé et bibliographie*

Oakley ressemble beaucoup à SKEME, et permet en plus la négociation d'un ensemble de paramètres.

Oakley est décrit dans la [RFC 2412].

## *3.3. La gestion des clefs pour IPsec : ISAKMP et IKE*

IKE (*Internet Key Exchange*) est un système développé spécifiquement pour IPsec qui vise à fournir des mécanismes d'authentification et d'échange de clef adaptés à l'ensemble des situations qui peuvent se présenter sur l'Internet. Il est composé de plusieurs éléments : le cadre générique ISAKMP et une partie des protocoles Oakley et SKEME. Lorsqu'il est utilisé pour IPsec, IKE est de plus complété par un "domaine d'interprétation" pour IPsec.

### 3.3.1. ISAKMP

Nous avons vu au début de cette présentation que l'apport de services de sécurité passait par l'utilisation d'associations de sécurité qui permettent de définir les paramètres (clefs, protocoles…) nécessaires à la sécurisation d'un flux donné. ISAKMP (*Internet Security Association and Key Management Protocol*) a pour rôle **la négociation, l'établissement, la modification et la suppression des associations de sécurité et de leurs attributs.**

#### a/ Indépendance vis à vis des mécanismes : les domaines d'interprétation et les phases

ISAKMP est un **cadre générique indépendant des mécanismes en faveur desquels la négociation a lieu**. En effet, ISAKMP est a priori utilisable pour négocier, sous forme d'associations de sécurité, les paramètres relatifs à n'importe quels mécanismes de sécurité : IPsec, TLS… Il est en effet prévu pour supporter la négociation de SA pour n'importe quel protocole de niveau supérieur ou égal à IP.

Ceci est possible grâce à la façon dont les négociations ont lieu. Tout d'abord, ISAKMP est prévu pour fonctionner indépendamment des mécanismes pour lesquels il travaille : les données relatives à la gestion des clefs seront transportées à part. ISAKMP peut être implémenté directement au-dessus d'IP, ou au-dessus de tout protocole de la couche transport. Il s'est notamment vu attribuer le port 500 sur UDP par l'IANA.

Ensuite, ISAKMP définit un cadre pour négocier les associations de sécurité, mais n'impose rien quant aux paramètres qui les composent. **Un document appelé "domaine d'interprétation" (*Domain of Interpretation*, DOI) doit définir les paramètres négociés et les conventions relatives à l'utilisation de ISAKMP dans un cadre précis**. Un identificateur de DOI est utilisé pour interpréter le contenu des messages ISAKMP. La [RFC 2407] définit le DOI pour l'utilisation de ISAKMP avec IPsec. Nous reviendrons sur ce document dans le chapitre suivant.

Enfin, **ISAKMP comporte deux phases**, qui permettent une séparation nette de la négociation des SA pour un protocole donné et de la protection du trafic propre à ISAKMP :

- Durant la **première phase**, un ensemble d'attributs relatifs à la sécurité est négocié, les identités des tiers sont authentifiées et des clefs sont générées. Ces éléments constituent une première "association de sécurité", dite **SA ISAKMP**. Contrairement aux SA IPsec, la SA ISAKMP est bidirectionnelle. Elle servira à sécuriser l'ensemble des échanges ISAKMP futurs.
- La **seconde phase** permet de négocier les paramètres de sécurité relatifs à une **SA à établir pour le compte d'un mécanisme de sécurité donné** (par exemple AH ou ESP). Les échanges de cette phase sont sécurisés (confidentialité, authenticité...) grâce à la SA ISAKMP. Celle-ci peut bien sûr être utilisée pour négocier plusieurs SA destinées à d'autres mécanismes de sécurité.

Les paramètres de la SA ISAKMP peuvent être propres à ISAKMP seulement, ou comporter également des éléments propres à un protocole de sécurité donné et définis dans le DOI correspondant. Dans le premier cas, on parlera de *Generic ISAKMP SA*, et celle-ci pourra servir pour la négociation de SA pour n'importe quel protocole de sécurité. Dans le second cas, la SA ISAKMP ne pourra être utilisée que pour négocier une SA dépendant du même DOI.

### b/ Indépendance vis à vis du protocole de gestion des clefs : la construction des messages par blocs

ISAKMP est également indépendant de la méthode de génération des clefs et des algorithmes de chiffrement et d'authentification utilisés. Il est donc indépendant de tout protocole d'échange de clef, ce qui permet de séparer clairement les détails de la gestion des associations de sécurité des détails de l'échange de clef. Différents protocoles d'échange de clef, présentant des propriétés différentes sont ainsi utilisables avec ISAKMP.

Ceci est possible à cause du fait que ISAKMP n'impose pas un déroulement fixe, basé sur la définition d'un ensemble de messages limité. ISAKMP est plutôt une sorte de "kit de construction", puisque **les messages d'ISAKMP sont constitués d'un en-tête suivi d'un nombre variable de blocs. Ces blocs ( *payloads* en anglais) sont les briques de base d'ISAKMP**.

Chaque message ISAKMP commence par un en-tête (*ISAKMP Header*) de longueur fixe. Celui-ci comprend notamment deux ***cookies***, l'*initiator cookie* et le *responder cookie*, qui, en plus du rôle classique de protection contre le déni de service (*anti-clogging*), **permettent d'identifier l'association de sécurité ISAKMP** en cours (contrairement aux autres SA, elle n'est pas identifiée pas un SPI).

Un champ *Exchange Type* permet de connaître le type d'échange en cours (voir plus loin) et donc de connaître l'organisation et le nombre des messages.

L'en-tête ISAKMP comprend également un champ, *Next Payload*, qui indique le type du premier bloc du message. Chaque bloc commence à son tour par un en-tête propre, qui indique la longueur du bloc courant et le type du bloc suivant. Le dernier bloc du message indique 0 comme type de bloc suivant. La construction des messages ISAKMP repose donc sur le **chaînage de blocs**.

Le document de base sur ISAKMP définit 13 types de blocs différents :

| Nom | Sigle |
|---|---|
| *Security Association* | SA |
| *Proposal* | P |
| *Transform* | T |
| *Key Exchange* | KE |
| *Identification* | ID |
| *Certificate* | CERT |
| *Certificate Request* | CR |
| *Hash* | HASH |
| *Signature* | SIG |
| *Nonce* | NONCE |
| *Notification* | N |
| *Delete* | D |
| *Vendor ID* | VID |

- **Le bloc *Security Association (SA)* est utilisé pour négocier les attributs de sécurité**.
  En lui-même, il contient des champs qui indiquent le contexte de la négociation (DOI et situation). La **situation** est un paramètre qui dépend du DOI et qui permet d'indiquer quel type de politique de sécurité on désire utiliser. Une valeur de 0 pour le **DOI** pendant la phase 1 indique que l'on négocie une SA ISAKMP générique. Une valeur de 1 indique le DOI IPsec.
  Un bloc SA est toujours suivi d'un ou plusieurs blocs *Proposal*, qui permettent de faire des propositions (présentées par ordre de préférence) à l'interlocuteur.

- **Chaque bloc *Proposal* constitue une proposition d'un ensemble d'attributs relatifs à une association de sécurité.**
  En lui-même, le bloc *Proposal* indique le **mécanisme de sécurité** que l'on désire utiliser (AH, ESP…) ainsi que le **SPI** à associer à la SA si cette proposition est retenue. Comme il est possible de laisser le choix à l'interlocuteur en lui faisant plusieurs propositions, chaque bloc *Proposal* est numéroté. Lorsque plusieurs propositions constituent un tout (par exemple si l'on veut négocier une protection par AH + ESP), elles portent le même numéro et résulteront en la création d'un paquet de SA.
  Un bloc P est toujours suivi d'un ou plusieurs blocs *Transform*, qui permettent de préciser les attributs choisis pour la SA en question.

- **Chaque bloc *Transform* indique une transformation (algorithme de chiffrement, fonction de hachage…) et ses attributs.** Ces éléments dépendent bien sûr du DOI et du mécanisme de sécurité sélectionné dans les blocs précédents.
  Comme pour les blocs *Proposal*, les blocs *Transform* sont numérotés : si deux blocs portent le même numéro, ils forment un tout et doivent être sélectionnés ensemble ; des blocs de numéros différents indiquent une possibilité de choix.

Ces trois premiers types de blocs ne sont pas indépendants et on peut considérer qu'ils sont emboîtés. On désigne donc souvent par le bloc SA seul l'ensemble des blocs SA, P et T.

| SA | P1 | T1.1 | T1.2 | T1.3 | P2 | T2.1 | T2.2 |
|---|---|---|---|---|---|---|---|
| →DOI<br>→Situation | → Mécanisme<br>→ SPI | → Transfo.<br>→ Attributs | → Transfo.<br>→ Attributs | → Transfo.<br>→ Attributs | → Mécanisme<br>→ SPI | → Transfo.<br>→ Attributs | → Transfo.<br>→ Attributs |

*- Figure 10. Organisation des blocs SA, P et T –*

L'ensemble représenté dans le schéma ci-dessus pourrait être un ensemble de propositions envoyé par un tiers à un autre. Le destinataire de ce message doit répondre par une suite identique dans laquelle il ne conserve que la proposition (ou le groupe de propositions) retenue. L'association de sécurité (ou le

paquet d'associations de sécurité) résultant de cette négociation se verra attribué le SPI de la proposition retenue.

- **Le bloc *Key Exchange* sert à transporter les données nécessaires à la génération de la clef de session**. Son interprétation dépend du DOI et du protocole d'échange de clefs choisi.

- **Le bloc *Identification* sert à transporter les données nécessaires à l'identification des tiers.** Son interprétation dépend du DOI.
  Un champ intitulé *ID Type* indique le type de donnée d'identification contenu dans le bloc. Pour ISAKMP ce sont une adresse IP (IPv4 ou IPv6) ou une plage d'adresses IP (adresse / masque de sous-réseau). Chaque DOI peut définir différents types d'identification.

- **Le bloc *Certificate* fournit un moyen de transporter des certificats ou toute autre information en relation avec les certificats.**
  Un champ intitulé *Certificate Encoding* indique le type de certificat ou de donnée relative aux certificats contenue dans le champ *Certificate Data*. Les types définis actuellement sont :
  - *PKCS #7 wrapped X.509 certificate*
  - *PGP certificate*
  - *DNS signed key*
  - *X.509 certificate – signature*
  - *X.509 certificate – key exchange*
  - *Kerberos tokens*
  - *Certificate revocation list (CRL)*
  - *Authority revocation list (ARL)*
  - *SPKI certificate*
  - *X.509 certificate – attribute*

- **Le bloc *Certificate Request* permet à un tiers de réclamer un certificat à son interlocuteur.**
  Comme dans le bloc précédent, un champ indique le type de certificat requis. Un second champ, intitulé *Certificate Authority*, contient les références d'une autorité de certification acceptable par le demandeur. Ce champ est facultatif.

- **Le bloc *Hash* contient le résultat de l'application d'une fonction de hachage sélectionnée au préalable à tout ou partie du message ou à une variable d'état donnée.**
  Ce bloc peut être utilisé à des fins de vérification de l'authenticité d'un message ISAKMP d'authentification des tiers en présence.

- De même, **le bloc *Signature* contient le résultat de l'application d'une fonction de signature numérique sélectionnée au préalable à tout ou partie du message ou à une variable d'état donnée.**
  L'utilisation est la même que pour le bloc *Hash*.

- **Le bloc *Nonce* sert à transporter des aléas.**

- **Le bloc *Notification* sert à véhiculer des messages d'erreur ou d'information sur l'état actuel des négociations.** Son interprétation dépendant du DOI, celui-ci est indiqué au début du bloc.
  Le début du bloc contient également les références de l'association de sécurité concernée (mécanisme et SPI). Le message est représenté par deux champs : *Notify Message Type* et *Notification Data* (facultatif).

- **Le bloc *Delete* permet à l'émetteur de signaler à son interlocuteur qu'il vient de supprimer une association de sécurité et que celle-ci n'est donc plus valable.**
  Un seul bloc permet éventuellement d'indiquer plusieurs SA, à conditions qu'elles soient toutes relatives au même mécanisme.
  Dans ISAKMP, la modification des SA se fait en créant une nouvelle SA puis en supprimant l'ancienne.

- **Le bloc *Vendor ID* peut être utilisé par un programmeur pour permettre à deux instances de son implémentation de se reconnaître** et de pouvoir ensuite utiliser des éléments propres à cette implémentation.

### c/ Interopérabilité et ligne directrice : les types d'échanges

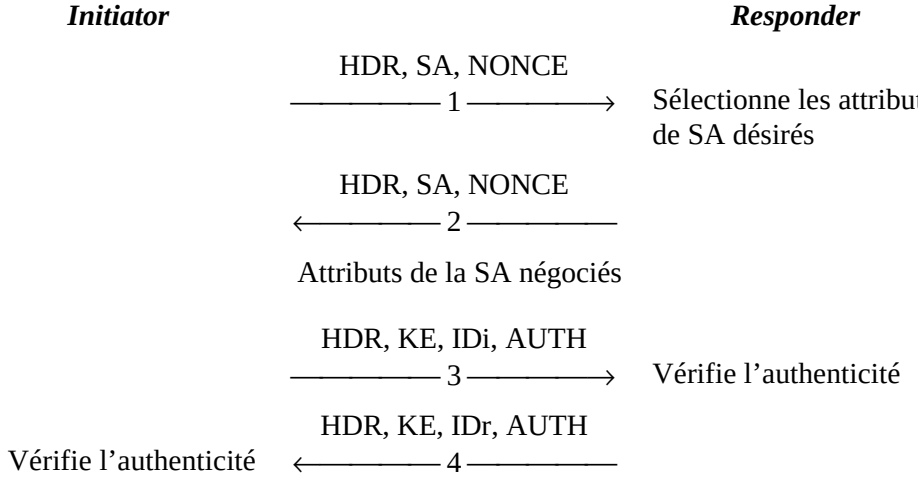
À partir de ces blocs, ISAKMP définit des types d'échanges (*exchange types*). Un type d'échange est une spécification de l'ensemble des messages constituant un type d'échange donné (nombre, contenu,...). Chaque type d'échange est conçu pour fournir un certain nombre de services de sécurité spécifiques, comme l'anonymat, la propriété de *perfect forward secrecy*, l'authentification mutuelle,... Le document de base sur ISAKMP, [RFC 2408], définit, à des fins d'interopérabilité et surtout pour indiquer comment doivent être utilisés les blocs dans le cadre d'une négociation donnée, des types d'échanges par défaut. D'autres types peuvent bien sûr être définis, en fonction du protocole d'échange de clef utilisé et du DOI notamment. D'autre part, plusieurs propositions de types d'échanges supplémentaires pour ISAKMP font l'objet d'*Internet Drafts* séparés.

La spécification actuelle de ISAKMP comporte cinq types d'échanges par défaut : *Base Exchange*, *Identity Protection Exchange*, *Authentication-Only Exchange*, *Aggressive Exchange*, *Informational Exchange*. Seul ce dernier est obligatoire. Tous ces échanges peuvent être utilisés soit durant la phase 1, soit durant la phase 2. Dans les descriptions qui vont suivre, on utilisera les notations suivantes :

| | |
|---|---|
| HDR | *ISAKMP Header* |
| SA | *Security Association Payload* |
| KE | *Key Exchange Payload* |
| ID | *Identity Payload* |
| AUTH | *Authentication Payload* (HASH ou SIG) |
| NONCE | *Nonce Payload* |
| * | Signifie que le message est chiffré |

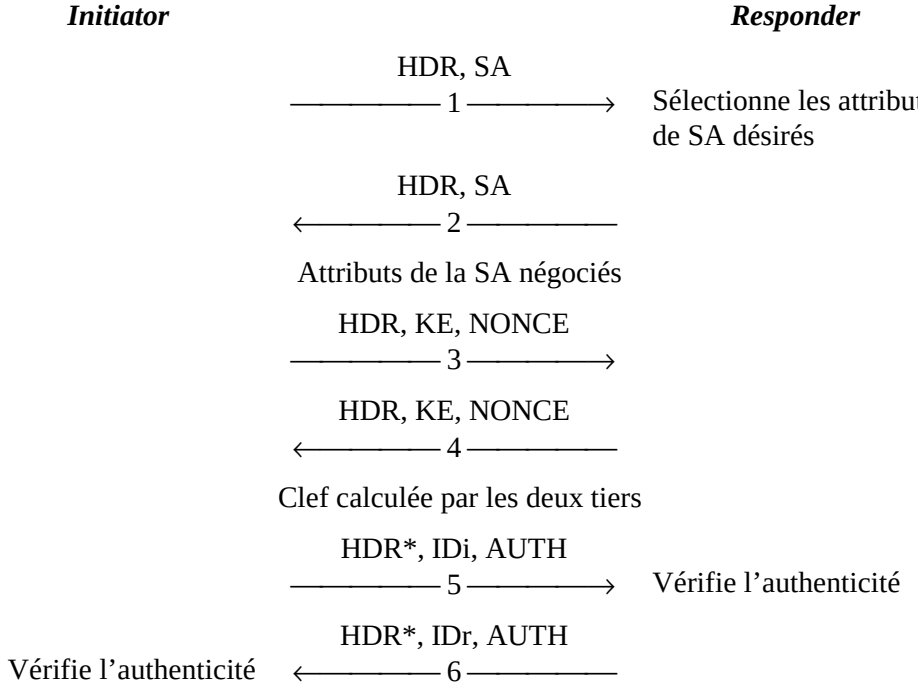
*- Figure 11. Notations pour les échanges ISAKMP -*

- **L'échange de base** (*Base Exchange*) est conçu pour permettre le transfert simultané des données d'identification et des données servant à la génération de la clef, ce qui permet de réduire le nombre total de messages nécessaires, au détriment de la protection de l'anonymat des tiers en présence. En effet ; les identités sont échangées avant qu'un secret partagé ne soit établi et ne permette de les chiffrer.

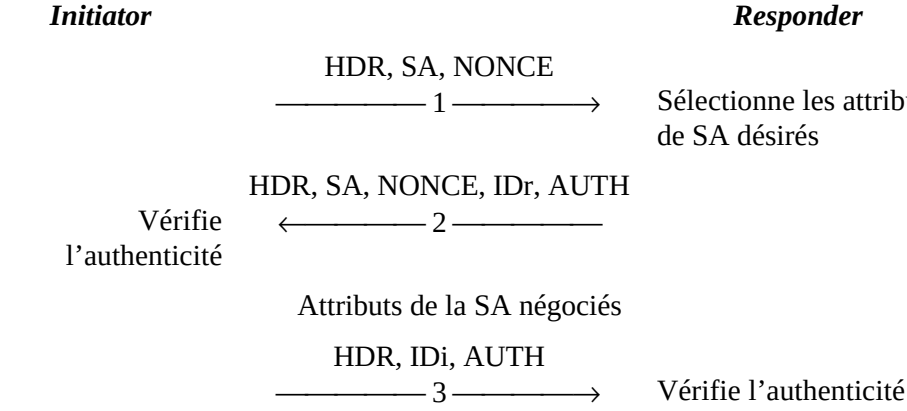



Les données échangées au cours des messages 3 et 4 sont protégées, par le biais du bloc AUTH, par la fonction d'authentification sélectionnée au cours de la négociation des messages 1 et 2.

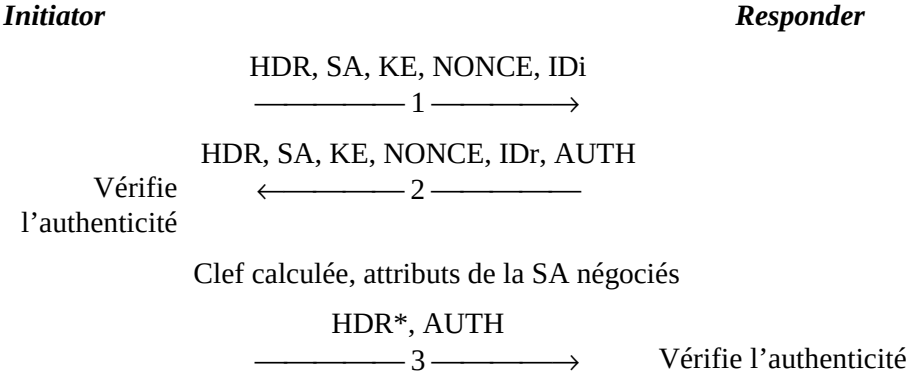
- **L'échange de protection d'identité** (*Identity Protection Exchange*), quant-à-lui, repousse l'envoi des données d'identification à après l'échange des données permettant la génération du secret partagé, assurant ainsi l'anonymat des tiers.



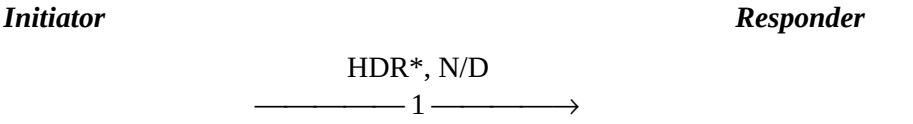

- **L'échange d'authentification seule** (*Authentication Only Exchange*) est conçu pour aboutir uniquement à l'authentification des tiers, évitant ainsi le temps de calcul engendré par la génération de clef lorsque celle-ci n'est pas nécessaire. Cet échange est particulièrement utile durant la seconde phase, où il sera protégé par les services de sécurité négociés au cours de la première phase.



- **L'échange agressif** (*Aggressive Exchange*) combine les données de négociation de la SA, d'authentification et d'échange de clef en un seul message afin de réduire au maximum le nombre de messages nécessaires. Tout comme pour l'échange de base, l'anonymat des tiers n'est donc pas préservé. De plus, le fait de ne pas séparer la négociation de la SA de l'envoi du bloc KE empêche la négociation du groupe Diffie-Hellman.




- **L'échange d'information** (*Informational Exchange*) est constitué d'un seul message et sert à transmettre une information relative à la gestion des SA : message d'erreur, information d'état, annonce de suppression de SA… Il utilise soit un bloc *Notification*, soit un bloc *Delete*. Ce message est protégé par la SA ISAKMP si celle-ci a déjà été établie, sinon il n'est pas protégé.




### *Résumé et bibliographie*

ISAKMP pose les bases permettant de construire divers protocoles de gestion des clefs (et plus généralement des associations de sécurité). Il comporte trois aspects principaux :

- Il définit une façon de procéder, en deux étapes appelées phase 1 et phase 2 : dans la première, un certain nombre de paramètres de sécurité propres à ISAKMP sont mis en place, afin d'établir entre les deux tiers un canal protégé ; dans un second temps, ce canal est utilisé pour négocier les associations de sécurité pour les mécanismes de sécurité que l'on souhaite utiliser (AH et ESP par exemple)
- Il définit des formats de messages, par l'intermédiaire de blocs ayant chacun un rôle précis et permettant de former des messages clairs.
- Il présente un certain nombre d'échanges types, composés de tels messages, qui permettant des négociations présentant des propriétés différentes : protection ou non de l'identité, *perfect forward secrecy*,…

ISAKMP est décrit dans la [RFC 2408].

## 3.3.2. IPsec DOI

ISAKMP définit un cadre pour négocier les associations de sécurité, mais n'impose rien quant aux paramètres qui les composent. **Un document appelé "domaine d'interprétation" (*Domain of Interpretation*, DOI) définit les paramètres négociés et les conventions relatives à l'utilisation de ISAKMP dans un cadre précis**. Un identificateur de DOI est utilisé pour interpréter le contenu des messages ISAKMP. La [RFC 2407] définit le DOI pour l'utilisation de ISAKMP avec IPsec.

### a/ Bloc SA : situation

Utilisé dans le bloc SA de ISAKMP, le champ situation permet de préciser la situation à laquelle doit être rattachée la négociation. Le DOI IPsec définit trois situations différentes :

- *Identity only*, qui impose une identification des tiers par le biais d'un bloc ID.
- *Secrecy*, qui permet l'utilisation d'indicateurs de niveaux de confidentialité.
- *Integrity*, qui permet l'utilisation d'indicateurs ne niveaux pour l'intégrité.

Le DOI IPsec définit également des champs supplémentaires dans le bloc SA pour transporter les données relatives à la situation : niveau de confidentialité, niveau d'intégrité…

### b/ Bloc P : protocole de sécurité

Les protocoles (ou mécanismes) qui entrent dans le cadre du DOI IPsec sont :
- ISAKMP (pour une négociation de phase 1 avec le DOI IPsec au lieu de *Generic ISAKMP*)
- AH
- ESP
- IPCOMP
  IPCOMP est un protocole de compression des données au niveau IP. Le chiffrement des données rendant inefficace toute compression ultérieure, il peut être intéressant de compresser les données avant de les chiffrer.

### c/ Bloc T : transformation et attributs

À chacun des quatre protocoles mentionnés ci-dessus sont associées des transformations permettant de faire des choix par le biais de blocs *Transform*.

Pour ISAKMP, cette méthode permet de choisir le protocole d'échange de clef à utiliser. Le seul choix possible dans le cadre de IPsec est IKE.

Les transformations relatives à AH sont MD5, SHA et DES. Cette information est à compléter, par le biais du champ *SA Attributes* du bloc *Transform*, par la référence de l'algorithme à utiliser. D'où en fait quatre possibilités : HMAC-MD5, KPDK-MD5, HMAC-SHA, DES-MAC.

Les transformations relatives à ESP sont DES_IV32, DES_IV64 (DES en mode CBC avec un vecteur d'initialisation de longueur 32 ou 64 respectivement), DES (DES en mode CBC, demande une précision par le biais d'attributs), 3DES (demande une précision par le biais d'attributs), RC5, IDEA, CAST, BLOWFISH, 3IDEA, RC4 et NULL (permet d'utiliser ESP sans chiffrement des données).

Enfin, les transformations utilisables avec IPCOMP sont OUI (transformation propriétaire, à préciser par le biais d'un attribut), DEFLATE, LZS et V42BIS.

En plus des attributs mentionnés ci-dessus, il est possible d'avoir recours à divers attributs pour préciser le groupe sur lequel effectuer les calculs relatifs à DIFFIE–HELLMAN, la durée de vie de la SA, la longueur de la clef pour les algorithmes à clef de longueur variable... Ces éléments sont décrits en détail dans [RFC 2407].

### d/ Bloc ID

Le DOI IPsec ajoute au bloc ID les champs *Protocol ID* (UDP, TCP...) et *Port*, et définit les modes d'identification suivants :
- Adresse IPv4, adresse IPv6
- Sous réseau IPv4 ou IPv6 (adresse / masque de sous-réseau)
- Plage d'adresses IPv4 ou IPv6 (adresse de début, adresse de fin)
- *FQDN* (foo.bar.com)
- *User FQDN* (user@foo.bar.com)
- *Binary DER encoding of an ASN.1 X.500 Distinguished Name [X.501]*
- *binary DER encoding of an ASN.1 X.500 General Name [X.509]*
- *KEY ID :* information propre à un fournisseur et permettant d'identifier le secret partagé préalable à utiliser

### e/ Bloc N

3 nouveaux messages de statut sont introduits :

- *RESPONDER-LIFETIME*
- *REPLAY-STATUS*
- *INITIAL-CONTACT*

#### *Résumé et bibliographie*

Le DOI IPsec est un document qui définit les paramètres négociés et les conventions relatives à l'utilisation de ISAKMP dans le cadre d'IPsec.

Le domaine d'interprétation IPsec fait l'objet de la [RFC 2407].

## 3.3.3. IKE

Si le DOI IPsec précise la contenu des blocs ISAKMP dans le cadre de IPsec, IKE en revanche utilise ISAKMP, pour construire un protocole pratique. Le protocole de gestion des clefs associé à ISAKMP dans ce but est inspiré à la fois d'Oakley et de SKEME. Plus exactement, IKE utilise certains des modes définis par Oakley et emprunte à SKEME son utilisation du chiffrement à clef publique pour l'authentification et sa méthode de changement de clef rapide par échange d'aléas. D'autre part, IKE ne dépend pas d'un DOI particulier mais peut utiliser tout DOI.

IKE comprend quatre modes : le mode principal (*Main Mode*), le mode agressif *(Aggressive Mode*), le mode rapide (*Quick Mode*) et le mode nouveau groupe (*New Group Mode*). *Main Mode* et *Aggressive Mode* sont utilisés durant la phase 1, *Quick Mode* est un échange de phase 2. *New Group Mode* est un peu à part : ce n'est ni un échange de phase 1, ni un échange de phase 2, mais il ne peut avoir lieu qu'une fois une SA ISAKMP établie ; il sert à se mettre d'accord sur un nouveau groupe pour de futurs échanges DIFFIE–HELLMAN.

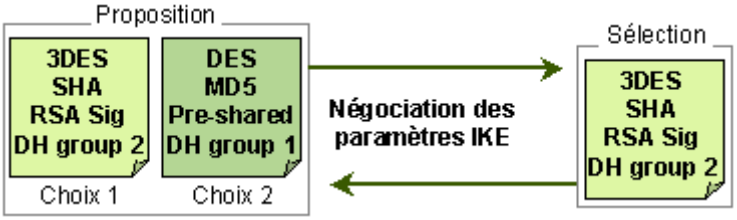
### a/ Phase 1 : Main Mode et Aggressive Mode

Les attributs suivants sont utilisés par IKE et négociés durant la phase 1 : un algorithme de chiffrement, une fonction de hachage, une méthode d'authentification et un groupe pour DIFFIE–HELLMAN.

**Trois clefs sont générées à l'issue de la phase 1 : une pour le chiffrement, une pour l'authentification et une pour la dérivation d'autres clefs.** Ces clefs dépendent des *cookies*, des aléas échangés et des valeurs publiques DIFFIE–HELLMAN ou du secret partagé préalable. Leur calcul fait intervenir la fonction de hachage choisie pour la SA ISAKMP et dépend du mode d'authentification choisi. Pour connaître les formules exactes, référez-vous à [RFC 2409].

Composé de six messages, ***Main Mode*** est une instance de l'échange ISAKMP *Identity Protection Exchange* :

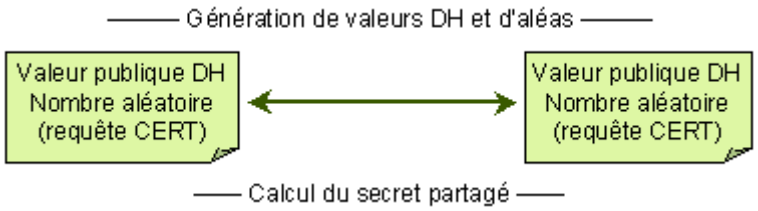
- Les deux premiers messages servent à **négocier les paramètres IKE** : algorithme de chiffrement, fonction de hachage, méthode d'authentification des tiers et groupe pour DIFFIE–HELLMAN.




*- Figure 12. Main Mode : exemple de premier échange -*

Les quatre méthodes d'authentification possibles sont la signature numérique, deux formes d'authentification par chiffrement à clef publique et l'utilisation d'un secret partagé préalable.

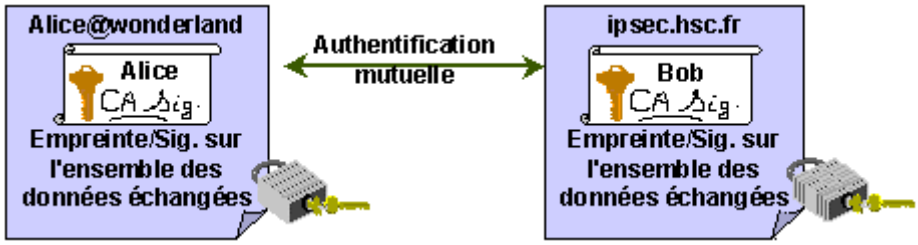
- Les deux seconds messages permettent l'**établissement d'un secret partagé** *via* l'utilisation de à l'échange de valeurs publiques DIFFIE–HELLMAN.




*- Figure 13. Main Mode : second échange -*

Le secret partagé sert à **dériver des clefs de session**, deux d'entre elles étant utilisées pour protéger la suite des échanges avec les algorithmes de chiffrement et de hachage négociés précédemment.

- Les deux derniers messages servent à l'authentification de des valeurs publiques.




*- Figure 14. Main Mode : troisième et dernier échange -*

***Aggressive Mode*** est une instance de l'échange ISAKMP *Aggressive Exchange* : il combine les échanges décrits ci-dessus pour Main Mode de façon à ramener le nombre total de messages à trois.

Dans ces deux cas, la méthode choisie pour l'authentification influence le contenu des messages et la méthode de génération de la clef de session. La [RFC 2409] détaille les messages échangés dans chaque cas et les formules de calcul des signatures et empreintes.

### b/ Phase 2 : Quick Mode

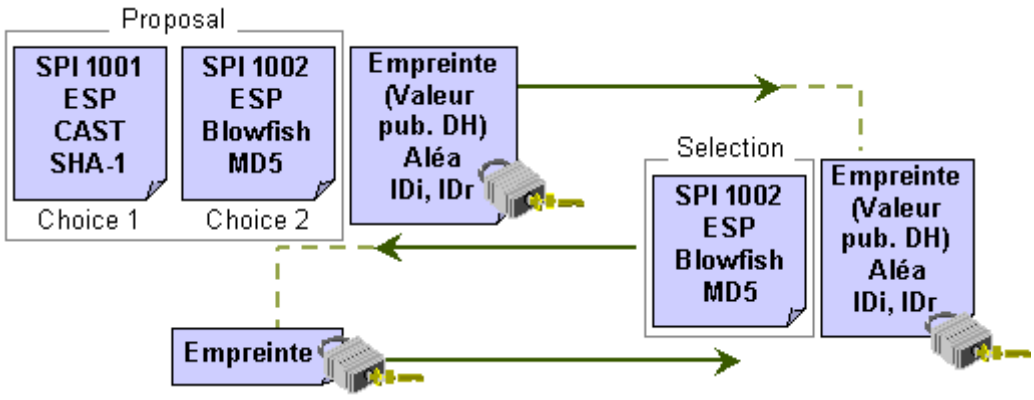
**Les messages échangés durant la phase 2 sont protégés en authenticité et en confidentialité grâce aux éléments négociés durant la phase 1**. L'authenticité des messages est assurée par l'ajout d'un bloc HASH après l'en-tête ISAKMP, et la confidentialité est assurée par le chiffrement de l'ensemble des blocs du message.

*Quick Mode* est utilisé pour la négociation de SA pour des protocoles de sécurité donnés comme IPsec. Chaque négociation aboutit en fait à deux SA, une dans chaque sens de la communication.

Plus précisément, les échanges composant ce mode ont le rôle suivant :

- Négocier un ensemble de paramètres IPsec (paquets de SA)
- Échanger des nombres aléatoires, utilisés pour générer une nouvelle clef qui dérive de celle de la SA ISAKMP. **De façon optionnelle, il est possible d'avoir recours à un nouvel échange DIFFIE-HELLMAN, afin d'accéder à la propriété de *Perfect Forward Secrecy***, qui n'est pas fournie si on se contente de générer une nouvelle clef à partir de l'ancienne et des aléas.
- Optionnellement, identifier le trafic que ce paquet de SA protégera, au moyen de sélecteurs (blocs optionnels IDi et IDr ; en leur absence, les adresses IP des interlocuteurs sont utilisées)

Les échanges composant ce mode sont les suivants :




*- Figure 15. Quick Mode -*

Ou, en terme de composition des messages par blocs :

***Initiator*** ***Responder***

HDR*, HASH, SA, NONCE [, KE] [, IDi, IDr]

——————— 1 ———————→

HDR*, HASH, SA, NONCE [, KE] [, IDi, IDr]

←——————— 2 ———————

Clef calculée, attributs de la SA négociés

HDR*, HASH

——————— 3 ———————→

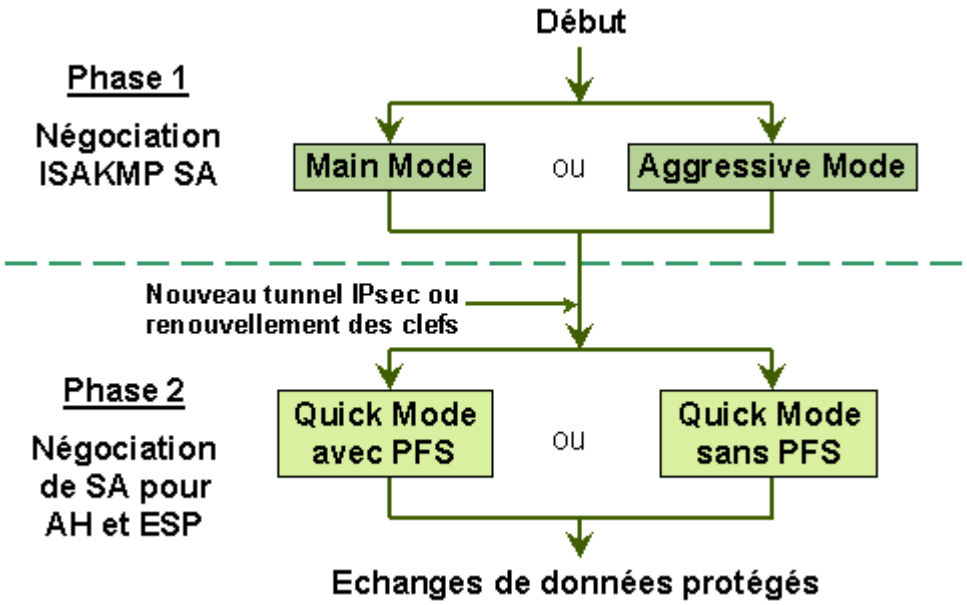
## c/ Les groupes : New Group Mode

Le groupe à utiliser pour DIFFIE-HELLMAN peut être négocié, par le biais du bloc SA, soit au cours du *Main Mode*, soit ultérieurement par le biais du *New Group Mode*. Dans les deux cas, il existe deux façons de désigner le groupe à utiliser :

- Donner la référence d'un groupe prédéfini : il en existe actuellement quatre, les quatre groupes Oakley (deux groupes MODP et deux groupes EC2N).
- Donner les caractéristiques du groupe souhaité : type de groupe (MODP, ECP, EC2N), nombre premier ou polynôme irréductible, générateurs…

### d/ Phases et modes

Au final, le déroulement d'une négociation IKE suit le diagramme suivant :




### *Résumé et bibliographie*

Combinaison d'ISAKMP, Oakley et SKEME, IKE (*Internet Key Exchange*), est le protocole de gestion de paramètres de sécurité utilisé par IPsec.

IKE comporte différents modes et options :

- Durant la phase 1, la négociation d'une SA ISAKMP peut se faire soit avec *Main Mode* (6 messages), soit avec *Aggressive Mode* (3 messages). L'authentification des tiers peut se faire par secret partagé préalable, chiffrement à clef publique ou signature.
- Durant la phase 2, la négociation de SA pour IPsec utilise *Quick Mode*, avec ou sans option de *Perfect Forward Secrecy*.

IKE est décrit dans la [RFC 2409].

# Annexe A – Sigles et acronyme

| | |
|---|---|
| 3DESE | Triple-DES Encryption protocol |
| AFNOR | Association Française de Normalisation |
| AH | Authentication Header |
| CAM | Code d'Authentification de Message |
| CBC | Cipher Block Chaining |
| DES | Data Encryption Standard |
| DH | DIFFIE–HELLMAN |
| DOI | Domain Of Interpretation |
| ESP | Encapsulating Security Payload |
| FAQ | Frequently Asked Questions |
| GSS-API | Generic Security Service API |
| HMAC | a MAC mechanism based on cryptographic Hash functions |
| IANA | Internet Assigned Numbers Authority |
| ICMP | Internet Control Message Protocol |
| ICV | Integrity Check Value |
| IDEA | International Data Encryption Algorithm |
| IETF | Internet Engineering Task Force |
| IKE | Internet Key Exchange |
| IP | Internet Protocol |
| IPng | IP new generation |
| IPPCP | IP Payload Compression Protocol |
| IPsec | IP security protocol |
| IPv4 | IP version 4 |
| IPv6 | IP version 6 |
| ISAKMP | Internet Security Association and Key Management Protocol |
| ISO | International Standardization Organization |
| IV | Initialization Vector |
| MAC | Message Authentication Code |
| MD*n* | Message Digest *n* |
| OSI | Open Systems Interconnection |
| PFS | Perfect Forward Secrecy |
| RACE | Research and development in Advanced Communication technologies in Europe |
| RC*n* | RIVEST'S Code *n* |
| RFC | Request For Comments |
| RIPE | RACE Integrity Primitives Evaluation |
| RIPE-MD | RIPE Message Digest |
| RSA | RIVEST, SHAMIR, ADLEMAN |

| | |
|---|---|
| SA | Security Association |
| SAD | Security Association Database |
| SHA | Secure Hash Algorithm |
| SKEME | a Versatile Secure Key Exchange Mechanism for Internet |
| SKIP | Simple Key management for Internet Protocols |
| SPD | Security Policy Database |
| SPI | Security Parameter Index |
| SSH | Secure SHell |
| SSL | Secure Socket Layer |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| TTL | Time To Live |
| UDP | User Datagram Protocol |
| VPN | Virtual Private Network |

# Annexe B – Glossaire

*Algorithme cryptographique ou de chiffrement*

Procédé ou fonction mathématique utilisée pour le *chiffrement* et le *déchiffrement*. Dans la cryptographie moderne, l'algorithme est souvent public et le secret du chiffre dépend d'un paramètre appelé *clef*.

*Analyse du trafic*

Observation des caractéristiques extérieures du trafic transitant sur un réseau afin de tenter d'en tirer des informations : fréquence des transmissions, identités des tiers communicants, quantités de données transférées. Associées à des informations de nature différente (date de rendez-vous, actualité...) ces éléments peuvent permettre aux adversaires de faire des déductions intéressantes.

*Association de sécurité (Security Association, SA)*

Connexion simplexe (unidirectionnelle) créée pour sécuriser des échanges. Tout le trafic qui traverse une SA se voit appliquer les mêmes services de sécurité. Une SA correspond donc à un ensemble de paramètres, qui caractérisent les services fournis au travers de mécanismes de sécurité comme AH et ESP.

⇨ Voir les chapitres "La notion d'association de sécurité" page 5 et "La gestion des clefs pour IPsec : ISAKMP et IKE" page 23.

*Authenticité*

Terme utilisé dans ce document pour désigner le service de sécurité qui consiste à assurer à la fois l'*intégrité* et l'*authentification de l'origine des données*.

*Authentification*

On distingue deux types d'authentification :

- Authentification d'un tiers.
  C'est l'action qui consiste à prouver son identité.
  Ce service est généralement rendu par l'utilisation d'un "échange d'authentification" qui implique un certain dialogue entre les tiers communicants.
- Authentification de l'origine des données
  Elle sert à prouver que les données reçues ont bien été émises par l'émetteur déclaré.
  Dans ce cas, l'authentification désigne souvent la combinaison de deux services : authentification et intégrité en mode non connecté. Ces deux services n'ont en effet pas de sens séparément et sont souvent fournis conjointement.

*Bump-in-the-stack*

Une implémentation est dite de type *bump-in-the-stack* si elle s'intercale entre deux couches de la pile de protocole (par exemple, entre PPP et le modem). La logique ainsi insérée est perçue par la couche de rang *n* comme étant celle de rang *n-1* et réciproquement.

*CAM* - Voir *Code d'authentification de message*

*Certificat*

Document électronique qui renferme la clef publique d'une entité, ainsi qu'un certain nombre d'informations la concernant, comme son identité. Ce document est signé par une autorité de certification ayant vérifié les informations qu'il contient.

*Chiffrement, chiffrer*

Application d'un *algorithme cryptographique* à un ensemble de données appelées *texte en clair* afin d'obtenir un *texte chiffré.*

Le chiffrement est un mécanisme de sécurité permettant d'assurer la confidentialité des données.

*Clef (secrète, publique, privée)*

Paramètre d'un algorithme de *chiffrement* ou de *déchiffrement*, sur lequel repose le secret.

On distingue deux types de clefs :

- les clefs secrètes, utilisées par les algorithmes symétriques, pour lesquels la clef de *chiffrement* et de *déchiffrement* sont égales.
- les couples (clef publique, clef privée), utilisés par les algorithmes asymétriques, pour lesquels clef de *chiffrement* et de *déchiffrement* sont distinctes.

*Clef de chiffrement de clefs*

Clef utilisée exclusivement pour chiffrer d'autres clefs, afin de les faire parvenir à un interlocuteur. Une clef de chiffrement de clef a généralement une durée de vie assez longue, par opposition aux clefs qu'elle sert à chiffrer.

⇨ Voir le chapitre "Types de clefs" page 16.

*Clef de session*

Clef ayant une durée de vie très limitée, généralement à une session.

Les clefs de session sont généralement des clefs secrètes, utilisées pour chiffrer les données transmises, et que les tiers communicants génèrent en début de communication.

⇨ Voir le chapitre "Types de clefs" page 16.

*Clef maîtresse*

Clef servant à générer d'autres clefs.

⇨ Voir le chapitre "Types de clefs" page 16.

*Code d'authentification de message (CAM)*

Résultat d'une *fonction de hachage à sens unique* à clef secrète. L'empreinte dépend à la fois des données et de la clef ; elle n'est donc calculable que par les personnes connaissant la clef. Adjointe aux données sur lesquelles elle a été calculée, elle permet de vérifier leur authenticité (authentification + intégrité).

*Confidentialité*

Service de sécurité qui consiste à s'assurer que seules les personnes autorisées peuvent prendre connaissance d'un ensemble de données.

Le mécanisme qui permet d'obtenir ce service est généralement le chiffrement des données concernées à l'aide d'un algorithme cryptographique.

On parle aussi de confidentialité du trafic lorsqu'on désire empêcher l'analyse du trafic en cachant les adresses source et destination, la taille des paquets, la fréquence des échanges...

*Connexion*

Relation logique établie entre deux entités.

Chaque couche réseau fournit aux couches supérieures un certain nombre de services dont certains sont dits sans connexion et d'autres orientés connexion. Dans un service **sans connexion**, chaque message est considéré comme totalement indépendant des autres et peut être envoyé à tout moment, sans procédure préalable et sans que le destinataire final soit

nécessairement présent à ce moment. C'est le cas par exemple de IP, qui n'offre qu'un service de type remise de datagrammes. Dans un service **orienté connexion**, l'initiateur de la communication doit d'abord établir un lien logique avec l'entité avec laquelle il désire communiquer. Cette procédure est appelée ouverture de la connexion et implique généralement de se mettre d'accord sur un certain nombre d'options.

*Contrôle d'accès*

Service de sécurité permettant de déterminer, après avoir authentifié un utilisateur, quels sont ses privilèges et de les appliquer. Ce service a pour but d'empêcher l'utilisation d'une ressource (réseau, machine, données...) sans autorisation appropriée.

*Cryptage, crypter*

Termes dérivés de l'anglais *to encrypt* et souvent employés incorrectement à la place de *chiffrement* et *chiffrer*. En toute rigueur, ces termes n'existent pas dans la langue française.

Si le "cryptage" existait, il pourrait être défini comme l'inverse du *décryptage*, c'est-à-dire comme l'action consistant à obtenir un *texte chiffré* à partir d'un *texte en clair* sans connaître la *clef*. Un exemple concret pourrait être de signer un texte choisi en reproduisant un chiffrement avec la *clef privée* de la victime. Mais on préfère parler dans ce cas de contrefaçon.

*Cryptanalyse ou analyse cryptographique*

Science qui étudie la sécurité des procédés cryptographiques pour tenter de trouver des faiblesses et pouvoir en particulier effectuer un *décryptement* avec succès.

"Analyse d'un système cryptographique, et/ou de ses entrées et sorties, pour en déduire des variables confidentielles et/ou des données sensibles (y compris un *texte en clair*)." [ISO 7498-2]

*Cryptogramme*

Aussi appelé *texte chiffré*. Données obtenues par application d'un algorithme de chiffrement. Le contenu sémantique de ces données n'est pas compréhensible.

*Cryptographie*

Étude du *chiffrement* et du *déchiffrement*, ainsi que des procédés permettant d'assurer l'intégrité, l'authentification...

"Discipline incluant les principes, moyens et méthodes de transformation des données, dans le but de cacher leur contenu, d'empêcher que leur modification passe inaperçue et/ou d'empêcher leur utilisation non autorisée." [ISO 7498-2]

*Cryptologie*

Étude scientifique de la *cryptographie* et de la *cryptanalyse*.

*Datagramme*

Les datagrammes sont des *paquets* totalement indépendants les uns des autres et circulant en mode non connecté. Par exemple, les paquets d'IP et ceux d'UDP sont des datagrammes.

Chaque paquet de type datagramme transite à travers le réseau avec l'ensemble des informations nécessaires à son acheminement, et notamment les adresses de l'expéditeur et du destinataire. Le routage étant effectué séparément pour chaque datagramme, deux datagrammes successifs peuvent donc suivre des chemins différents et être reçus par le destinataire dans un ordre différent de celui d'émission. De plus, en cas de problème dans le réseau, des datagrammes peuvent être perdus. Le destinataire doit donc réordonner les datagrammes pour reconstituer les messages et contrôler qu'aucun datagramme n'est perdu.

*Déchiffrement*

Action inverse du *chiffrement*, lorsque celui-ci est réversible : à l'aide d'un *algorithme cryptographique* et d'une *clef*, on reconstruit le *texte en clair* à partir du *texte chiffré*.

*Décryptement, décryptage*

Action qui consiste à "casser" le chiffrement d'un texte de façon à retrouver le *texte en clair* sans connaître la *clef* qui permet son *déchiffrement* normal.

*Déni de service*

"Impossibilité d'accès à des ressources pour des utilisateurs autorisés ou introduction d'un retard pour le traitement d'opérations critiques." [ISO 7498-2]

*Disponibilité*

Service de sécurité qui assure une protection contre les attaques visant à dégrader ou rendre impossible l'accès à un service.

*Empreinte (digest)*

Aussi appelé condensé.

Chaîne de taille fixe obtenue par application d'une *fonction de hachage* à un ensemble de données.

*Encapsulation, encapsuler*

Technique qui consiste à inclure un paquet d'un protocole à l'intérieur d'un paquet d'un autre protocole afin que ce dernier transporte le premier paquet. L'intérêt peut être soit de rendre possible l'utilisation du protocole encapsulé sur une liaison possédant le protocole encapsulant, soit de faire profiter le protocole encapsulé des services rendus par le protocole encapsulant. La façon la plus "logique" d'utiliser l'encapsulation est d'encapsuler un protocole de niveau supérieur dans un protocole de niveau inférieur, mais il est également possible de faire l'inverse.

*Fonction à sens unique*

Une fonction à sens unique est une fonction facile à calculer mais difficile à inverser.

La cryptographie à clef publique repose sur l'utilisation de fonctions à sens unique à brèche secrète : pour qui connaît le secret (i.e. la clef privée), la fonction devient facile à inverser.

*Fonction de hachage*

Aussi appelée fonction de condensation.

Fonction qui convertit une chaîne de longueur quelconque en une chaîne de taille inférieure et généralement fixe ; cette chaîne est appelée empreinte (*digest* en anglais) ou condensé de la chaîne initiale.

*Fonction de hachage à sens unique*

Fonction de hachage qui est en plus une fonction à sens unique : il est aisé de calculer l'empreinte d'une chaîne donnée, mais il est difficile d'engendrer des chaînes qui ont une empreinte donnée. On demande généralement en plus à une telle fonction d'être sans collision, c'est-à-dire qu'il soit impossible de trouver deux messages ayant la même empreinte.

*ICV (Integrity Check Value)*

"Valeur de vérification d'intégrité". Cette valeur est calculée par l'expéditeur sur l'ensemble des données à protéger. L'ICV est alors envoyée avec les données protégées. En utilisant le même algorithme, le destinataire recalcule l'ICV sur les données reçues et la compare à l'ICV originale. Si elles se correspondent, il en déduit que les données n'ont pas été modifiées.

⇨ Voir le chapitre “Authentication Header (AH)” page 12.

*Implémenter*

Anglicisme, de *to implement* : mettre en œuvre, réaliser.

*Intégrité*

Service de sécurité qui consiste à s’assurer que seules les personnes autorisées pourront modifier un ensemble de données. Dans le cadre de communications, ce service consiste à permettre la détection de l’altération des données durant le transfert.

On distingue deux types d’intégrité :

- L’intégrité en mode non connecté permet de détecter des modifications sur un datagramme individuel, mais pas sur l’ordre des datagrammes.
- L’intégrité en mode connecté permet en plus de détecter la perte de paquets ou leur réordonnancement.

L’intégrité est très liée à l’authentification de l’origine des données, et les deux services sont souvent fournis conjointement.

*MAC (Message Authentication Code)* - Voir *Code d’authentification de message*

*Mascarade (Masquerade, spoofing)*

Acte de prétendre être une autre entité dans le but d’accéder aux ressources de celle-ci ; incident au cours duquel un tiers non autorisé prétend être le véritable utilisateur.

*Message*

Dans le monde des réseaux, un message est une suite de données binaires formant un tout logique pour les tiers communiquants.

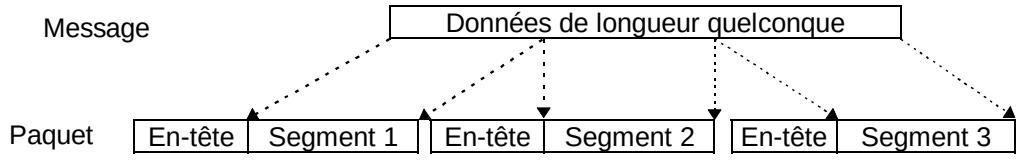
Lorsqu’un message est trop long pour être transmis d’un seul bloc, il est segmenté et chaque segment est envoyé séparément dans un *paquet* distinct.

*Non-rejouabilité*

Garantie qu’un adversaire ayant intercepté des messages au cours d’une communication ne pourra pas les faire passer pour des messages valides en les injectant soit dans une autre communication, soit plus tard dans la même communication.

*Paquet*

Un paquet est une suite de données binaires ne pouvant pas dépasser une longueur fixée. Il est obtenu en découpant un *message* en plusieurs segments et en ajoutant à chaque segment un entête contenant un certain nombre d’informations utiles à l’acheminement de ce paquet sur le réseau (options, destinataire...).




La taille maximale d’un paquet dépend du réseau ; un paquet peut correspondre à un message entier si celui-ci est court, mais en général il ne forme pas un tout logique pour le destinataire. Les paquets sont acheminés séparément jusqu’au destinataire, qui attend la réception de tous les paquets pour pouvoir reconstituer le message.

*Passerelle de sécurité (security gateway)*

Une passerelle de sécurité est un système intermédiaire (par exemple un routeur ou un garde-barrière) qui agit comme interface de communication entre un réseau externe considéré comme non fiable et un réseau interne de confiance. Elle fournit, pour les communications traversant le réseau non fiable, un certain nombre de services de sécurité.

Dans IPsec, une passerelle de sécurité est un équipement sur lequel sont implémentés AH et/ou ESP de façon à fournir des services de sécurité aux hôtes du réseau interne lorsqu'ils communiquent avec des hôtes externes utilisant aussi IPsec (soit directement soit par l'intermédiaire d'une autre passerelle de sécurité).

⇨ Voir le chapitre "Équipement fournissant IPsec" page 8.

*Perfect Forward Secrecy (PFS)*

Propriété d'un protocole d'échange de clef selon laquelle la découverte, par un attaquant, du ou des secrets à long terme utilisés ne permet pas de retrouver les clefs de sessions.

⇨ Voir le chapitre "Propriétés des protocoles d'échange de clef" page 17.

*Rejeu*

Action consistant à envoyer un message intercepté précédemment, en espérant qu'il sera accepté comme valide par le destinataire.

*Répudiation*

"Le fait, pour une des entités impliquées dans la communication, de nier avoir participé aux échanges, totalement ou en partie." [ISO 7498-2]

*Réseau privé virtuel*

Réseau composé par un ensemble d'hôtes et d'équipements qui utilisent des protocoles spécifiques pour sécuriser leurs communications.

⇨ Voir le chapitre "Équipement fournissant IPsec" page 8.

*RFC (Request For Comment)*

Littéralement, "Appel à commentaires". C'est en fait un document décrivant un des aspects d'Internet de façon relativement formelle (généralement, spécification d'un protocole). Ces documents sont destinés à être diffusés à grande échelle dans la communauté Internet et servent souvent de référence. On peut les trouver sur la plupart des sites FTP.

*Sans connexion* - Voir *Connexion*

*Scellement (seal)* - Voir aussi *Code d'authentification de message*

Mécanisme de sécurité permettant d'assurer l'intégrité et l'authentification de l'origine des données.

*Security gateway* - Voir *Passerelle de sécurité*

*Signature numérique*

"Données ajoutées à une unité de données, ou transformation cryptographique d'une unité de données, permettant à un destinataire de prouver la source et l'intégrité de l'unité de données et protégeant contre la contrefaçon (par le destinataire, par exemple)." [ISO 7498-2].

Une signature numérique fournit donc les services d'authentification de l'origine des données, d'intégrité des données et de non-répudiation. Ce dernier point la différencie des *codes d'authentification de message*, et a pour conséquence que la plupart des algorithmes de signature utilisent la cryptographie à clef publique.

D'autre part, la signature peut prendre deux formes :

- "transformation chiffrée" : un algorithme cryptographique modifie directement le message (par exemple chiffrement du message avec une clef privée).
- "données annexées" : des données supplémentaires sont adjointes au message (par exemple une empreinte, chiffrée avec une clef privée).

*Somme de contrôle*

Condensé d'un ensemble de données, calculé par l'expéditeur avant l'envoi des données et recalculé par le destinataire à la réception pour vérifier l'intégrité des données transmises.

*SPI (Security Parameter Index)*

Bloc de 32 bits qui, associé à une adresse de destination et au nom d'un protocole de sécurité (par exemple AH ou ESP), identifie de façon unique une *association de sécurité* (SA). Le SPI est transporté dans chaque paquet de façon à permettre au destinataire de sélectionner la SA qui servira à traiter le paquet. Le SPI est choisi par le destinataire à la création de la SA.

⇨ Voir le chapitre "La notion d'association de sécurité" page 5.

*Texte chiffré*

Aussi appelé *cryptogramme*.

Données obtenues par application d'un algorithme de chiffrement. Le contenu sémantique de ces données n'est pas compréhensible.

*Texte en clair*

Données intelligibles, dont la sémantique est compréhensible.

*Tunneling*

Technique consistant à créer un "tunnel" entre deux points du réseau en appliquant une transformation aux paquets à une extrémité (généralement, une *encapsulation* dans un protocole approprié) et en les reconstituant à l'autre extrémité.

*Vecteur d'initialisation (Initialization Vector, IV)*

Bloc de texte de valeur quelconque servant à initialiser un chiffrement avec chaînage de blocs, et donc à faire en sorte que deux messages identiques donnent des cryptogrammes distincts.

*Virtual Private Network (VPN)* - Voir *Réseau privé virtuel*

# Annexe C – Bibliographie commentée

Cette bibliographie liste les principaux documents utilisés pour la réalisation de cette présentation, classés par thème. L'index permet de retrouver rapidement un document d'après sa référence.

La plupart des documents cités ci-dessous sont disponibles en ligne. Ainsi, on trouvera les RFC sur de nombreux sites FTP (par exemple ftp://ftp.normos.org/ietf/rfc/) et les *Internet Drafts* sur le site de l'IETF (http://www.ietf.org/). L'adresse à laquelle trouver un article est indiquée lorsque celui-ci est accessible en ligne. À ce sujet, la page de la compagnie CRYPTOGRAPHY RESEARCH intitulée *Cryptography Author Sites* (http://www.cryptography.com/resources/authors/) liste les pages *web* de nombreux auteurs, permettant ainsi d'accéder à de nombreux articles.

## *C.1. Cryptographie*

### C.1.1. Généralités


[SCH96]
SCHNEIER Bruce, *Cryptographie appliquée - Algorithmes, protocoles et codes source en C - 2ème édition*, International Thomson Publishing France, 1997.
Ce livre est une traduction, le titre original est *Applied Cryptography - Protocols, Algorithms, and Source Code in C - 2nd Edition*.
⇨ Livre de référence sur la cryptographie moderne.

[RSA - FAQ]
RSA LABORATORIES, *Frequently Asked Questions About Today's Cryptography - Version 4.1*, 2000.
☸ http://www.rsasecurity.com/rsalabs/faq/
⇨ Présentation rapide de nombreux points de cryptographie sous forme de FAQ.

[LAB98]
LABOURET Ghislaine, *Introduction à la cryptologie*, document de synthèse personnel, 1998.
☸ http://www.multimania.com/labouret/realisations/cryptologie/


### C.1.2. Codes d'authentification de messages


[TSU92]
TSUDIK Gene, *Message Authentication with One-Way Hash Functions*, ACM Computer Communication Review, Vol. 22, pp. 29-38, 1992.
☸ http://www.isi.edu/~gts/paps/t92.ps.gz
⇨ Méthodes du préfixe secret, du suffixe secret et de l'enveloppe secrète.

[RFC 2104]
KRAWCZYK Hugo, BELLARE Mihir, CANETTI R, *HMAC: Keyed-Hashing for Message Authentication*, RFC 2104, février 1997.
⇨ HMAC.


### C.1.3. Échange de clef


[DOW92]
DIFFIE Whitfield, VAN OORSCHOT Paul C., WIENER Michael J., *Authentication and Authenticated Key Exchanges*, Designs, Codes and Cryptography, 2, pp. 107-125, Kluwer Academic Publishers, 1992.

⇨ Les protocoles d'authentification mutuelle avec échange de clef : définition d'un protocole sûr, les propriétés souhaitées pour de tels protocoles, présentation du protocole STS.

### C.1.4. Sécurité des réseaux et des échanges

[OPP98]
OPPLIGER Rolf, *Internet and Intranet Security*, Artech House, 1998.
⇨ Présentation des principales techniques existantes pour sécuriser les réseaux TCP/IP. Ce livre comporte notamment :
- un rappel sur le modèle OSI et l'architecture TCP/IP,
- une introduction rapide à la cryptographie,
- une présentation de différentes techniques de contrôle d'accès (gardes-barrières, filtres, relais),
- une présentation de protocoles sécurisés au niveau IP (IPsec et les protocoles de gestion des clefs correspondants), au niveau transport (SSH, SSL), et au niveau applicatif.

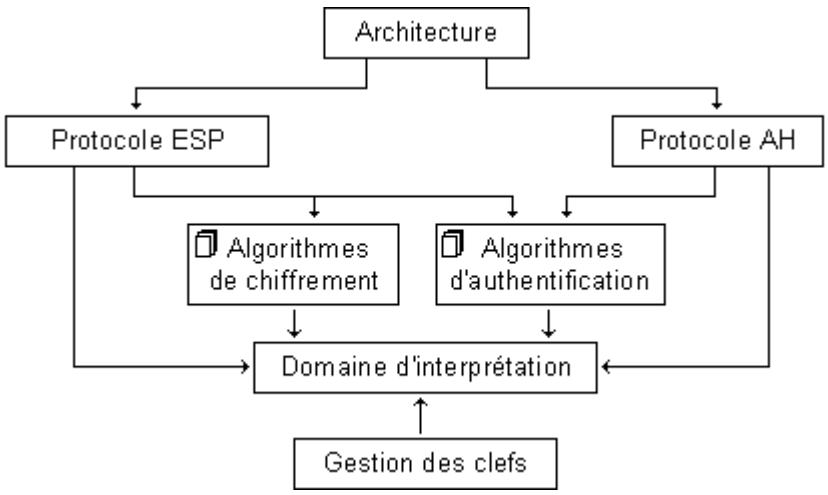
[ISO 7498-2]
International Standardization Organization, *Systèmes de traitement de l'information : interconnexion de systèmes ouverts - Modèle de référence de base - Partie 2 : architecture de sécurité*, NOR Z 70-102 (NF ISO 7498-2), AFNOR, 1989.
⇨ Description générale des services de sécurité OSI et des mécanismes associés, et notamment définitions formelles des termes relatifs à la sécurité des réseaux et des échanges.

## *C.2. IPsec*

La présentation de IPsec dans ce document est basée sur les RFC de novembre 1998 du groupe IPsec à l'IETF.

Pour obtenir la liste des *Internet drafts* relatifs à IPsec, je vous conseille de consulter la page *web* du groupe de travail sur IPsec à l'IETF (http://www.ietf.org/html.charters/ipsec-charter.html) ou directement la page listant les *Internet drafts* relatifs à IPsec (http://www.ietf.org/ids.by.wg/ipsec.html).

[RFC 2411]
THAYER Rodney, DORASWAMY Naganand, GLENN R., *IP Security Document Roadmap*, RFC 2411, novembre 1998.
⇨ Ce document explicite l'organisation des documents qui servent à décrire IPsec.
Voici une reproduction du schéma de cette organisation :



*- Figure 16. Organisation des documents sur IPsec -*

### C.2.1. Architecture

[RFC 2401]
ATKINSON Randall, KENT Stephen, *Security Architecture for the Internet Protocol*, RFC 2401, novembre 1998.
⇨ Présentation de l'architecture d'ensemble de IPsec et notamment de la notion d'association de sécurité (*Security Association, SA*).

### C.2.2. Protocole AH

[RFC 2402]
KENT Stephen, ATKINSON Randall, *IP Authentication Header*, RFC 2402, novembre 1998.

### C.2.3. Protocole ESP

[RFC 2406]
KENT Stephen, ATKINSON Randall, *IP Encapsulating Security Payload (ESP)*, RFC 2406, novembre 1998.

### C.2.4. Algorithmes d'authentification

[RFC 1828]
METZGER P., SIMPSON W., *IP Authentication using Keyed MD5*, RFC 1828, août 1995.
⇨ Keyed-MD5

[RFC 1852]
METZGER P., SIMPSON W., *IP Authentication using Keyed SHA*, RFC 1852, septembre 1995.
⇨ Keyed-SHA

[RFC 2403]
MADSON C., GLENN R., *The Use of HMAC-MD5-96 within ESP and AH*, RFC 2403, novembre 1998.

[RFC 2404]
MADSON C., GLENN R., *The Use of HMAC-SHA-1-96 within ESP and AH*, RFC 2404, novembre 1998.

[RFC 2857]
PROVOS Niels, KEROMYTIS Angelos, *The Use of HMAC-RIPEMD-160-96 within ESP and AH*, RFC 2857, juin 2000.

### C.2.5. Algorithmes de chiffrement

[RFC 1829]
KARN P., METZGER P., SIMPSON W., *The ESP DES-CBC Transform*, RFC 1829, août 1995.

[RFC 2405]
DORASWAMY Naganand, MADSON C., *The ESP DES-CBC Cipher Algorithm With Explicit IV*, RFC 2405, novembre 1998.

[RFC 2451]
ADAMS R., PEREIRA R., *The ESP CBC-Mode Cipher Algorithms*, RFC 2451, novembre 1998.
⇨ CAST-128, RC5, IDEA, Blowfish, DES triple

[RFC 2410]
KENT Stephen, GLENN R., *The NULL Encryption Algorithm and Its Use With IPsec*, RFC 2410, novembre 1998.

### C.2.6. Gestion des clefs

#### a/ SKIP

Le développement autour de SKIP se fait actuellement chez *Sun Microsystems* (www.sun.com) ; Les informations sur SKIP sont accessibles par le site www.skip.org.

[AMP97]
AZIZ Ashar, MARKSON Tom, PRAFULLCHANDRA Hemma, *Simple Key-Management for Internet Protocols (SKIP)*, Sun Microsystems Laboratories, avril 1997.
☸ http://www.skip.org/spec/SKIP.html
⇨ Spécifications techniques de SKIP ; ce document reprend l'*Internet draft* de SKIP, qui a expiré en 1997.

[draft-ietf-ipsec-skip-07]
AZIZ Ashar, MARKSON Tom, PRAFULLCHANDRA Hemma, *Simple Key-Management for Internet Protocols (SKIP)*, Internet Draft, a expiré, août 1996.
☸ http://www.skip.org/drafts/draft-ietf-ipsec-skip-07.txt
⇨ Le dernier *Internet draft* de SKIP en date.

[AZI98]
AZIZ Ashar, *SKIP Extension for Perfect Forward Secrecy (PFS)*, Sun Microsystems Laboratories, février 1998.
☸ http://www.skip.org/spec/EPFS.html
⇨ Extension de SKIP pour fournir la propriété de *perfect forward secrecy.*

#### b/ PHOTURIS

[RFC 2522]
SIMPSON W. A., KARN Phil, *Photuris: Session-Key Management Protocol*, RFC 2522, mars 1999.

[RFC 2523]
SIMPSON W. A., KARN Phil, *Photuris: Extended Schemes and Attributes*, RFC 2523, mars 1999.

#### c/ SKEME

[KRA96]
KRAWCZYK Hugo, *SKEME: A Versatile Secure Key Exchange Mechanism for Internet*, Symposium on Network and Distributed System Security, février 1996.
☸ http://info.isoc.org/conferences/ndss96/krawczyk.ps

#### d/ IKE (ISAKMP/Oakley)

[RFC 2408]
MAUGHAN Douglas, SCHERTLER Mark, SCHNEIDER Mark, TURNER Jeff, *Internet Security Association and Key Management Protocol (ISAKMP)*, RFC 2408, novembre 1998.
⇨ ISAKMP

[RFC 2412]
ORMAN Hilarie, *The OAKLEY Key Determination Protocol*, RFC 2412, novembre 1998.
⇨ Oakley

[RFC 2409]
HARKINS D., CARREL D., *The Internet Key Exchange (IKE)*, RFC 2409, novembre 1998.
⇨ IKE = ISAKMP/Oakley

#### e/ Domaine d'interprétation

[RFC 2407]
PIPER Derrell, *The Internet IP Security Domain of Interpretation for ISAKMP*, RFC 2407, novembre 1998.

### C.2.7. Cryptanalyse

[FS99]
FERGUSON Niels, SCHNEIER Bruce, *A Cryptographic Evaluation of IPsec*, décembre 1999.
☸ http://www.counterpane.com/ipsec.html

[BEL97]
BELLOVIN Steven M., *Probable Plaintext Cryptanalysis of the IP Security Protocols*, Proceedings of the Symposium on Network and Distributed System Security, pp. 155-160, février 1997.
☸ http://www.research.att.com/~smb/papers/probtxt.ps

[BEL96]
BELLOVIN Steven M., *Problem Areas for the IP Security Protocols*, Proceedings of the Sixth Usenix Unix Security Symposium, pp. 1-16, juillet 1996.
☸ http://www.research.att.com/~smb/papers/badesp.ps