Fiche technique

IPSec : Techniques

Bénoni MARTIN

Degré de difficulté

L'un des protocoles les plus complexes, complexité notamment dûe au fait que IPSec se base sur d'autres protocoles (AH, ESP, ISAKMP, IKE, ...) qu'il faut donc appréhender avant d'aborder IPSec ; complexité qui se traduit d'ailleurs par le nombre élevé de RFCs traitant du sujet. Pour certains, IPSec est basé sur le trio AH/ESP/IKE ; pour d'autres ce sera plutôt le trio IKE/ISAKMP/Oakley ; pour d'autres encore, IPSec est un ensemble de mécanismes destiné à pallier le manque de sécurité de IPv4, …

IPSec a été développé par l'IETF dans le but de sécuriser TCP/IP au niveau de la couche 3 (couche réseau du modèle OSI), contrairement à SSL/TLS ou SSH qui sécurisent respectivement les couches 6 et 7 (ce qui évite de rattacher IPSec à un port donné -22 pour SSH ou 443 pour HTTPs-). Il peut être implémenté sur des connexions hôte vers hôte, hôte vers passerelle ou passerelle vers passerelle. Le premier type requiert soit le mode transport, soit le mode tunnel, tandis que les deux derniers cas demandent forcément un mode tunnel. Par l'authentification et le chiffrement des paquets IP, IPSec permet de sécuriser toute transmission de données reposant sur TCP.

Nous présenterons donc IPSec comme :

- étant un *patch de sécurité* pour IP (IPSec étant en option sous IPv4, mais obligatoire avec le futur Ipv6),
- étant un protocole pouvant être utilisé sous deux modes (transport et tunnel),
- faisant appel à deux sous-protocoles (AH et ESP),
- et se basant sur plusieurs autres protocoles plus ou moins utilisés dans leur totalité (ISAKMP, IKE, Oakley, Photuris, Skeme et SKIP).

IPSec permet donc principalement :

- *l'authentification* – cette fonctionnalité repose entre autres sur le concept de cookie comme nous le verrons par la suite et est basée sur des clés prépartagées, adresses IP, noms FQDN, certificats X.509, ...,
- *l'intégrité des données* – via l'utilisation d'algorithmes de hachage, nous pouvons vérifier que les données n'ont pas été altérées entre le départ et l'arrivée. Cette intégrité repose sur deux types particuliers de fonctions de hachage : les MACs -cf. Encadré- et les HMACs -cf. Encadré-,

Cet article explique...

- Comment fonctionne IPSec en détail.

Ce qu'il faut savoir...

- Idéalement avoir des connaissances de base sur les protocoles TCP/UDP et IP.
- Des notions de base en cryptographie (clé prépartagée, échange de Diffie-Hellman, certificats et signatures numériques, ...).

La gestion des clés

Les 3 types de clés existantes

On a trois grands types de clés :

- les clés de chiffrement de clés. Servant à chiffrer d'autres clés (par exemple crypter la clé qui va permettre de transmettre la clé symétrique de chiffrement de données), ce type de clé doit être par conséquent très solide (d'où une utilisation recommandée de la cryptographie à clé publique) et a une durée de vie en général assez longue,
- les clés de chiffrement de données. Comme son nom l'indique, ce type de clés permet de crypter les données échangées. Les données à échanger pouvant être très grandes, le cryptage / décryptage doit être le plus rapide possible, d'où le choix de clés symétriques. La « fragilité » de ce type de clés est compensée par le fait que dans la plupart des cas, ces clés changent souvent (elles ne durent pas plus de 10 minutes par défaut par exemple sur un VPN monté sur un Firewall NetASQ),
- les clés maîtresses. Ces clés permettent de générer d'autres clés par dérivation, par exemple pour le chiffrement ou les signatures numériques.

Comment sont gérées les clés ?

La distribution des clés peut se faire soit manuellement soit automatiquement :

- dans la distribution manuelle, l'administrateur configure chaque équipement avec sa clé. Cette technique n'est réalisable que si le réseau est statique et de taille acceptable.
- dans la distribution automatique, les participants pourront utiliser des clés via DNS en utilisant un algorithme asymétrique. Ces clés authentifieront les messages de distribution de clés. Les protocoles les plus utilisés dans cette dernière distribution sont ISAKMP, OAKLEY et IKE.

- *la non-répudiation* – possibilité d'identifier formellement l'émetteur de manière à ce que ce dernier ne puisse nier être l'auteur du message. Cette option repose sur le concept de signature numérique -cf. Encadré-,
- *la confidentialité des données* – via le cryptage, nous pouvons empêcher qu'un attaquant ne puisse lire nos données,
- *l'anti-rejeu* – cette option sera développée en détail lorsque nous parlerons du PFS (protection anti-rejeu).

Ces fonctionnalités sont données à travers l'utilisation de deux sous-protocoles de IPSec :

- l'AH -Authentication Header- qui est conçu pour assurer principalement l'intégrité et l'authentification des données,
- l'ESP -Encapsulating Security Payload- qui assure la confidentialité par cryptage, et aussi éventuellement l'authentification. ESP est largement plus utilisée que AH.

IPSec en détail

Une connexion IPSec repose sur l'usage d'une association de sécurité (SA -Security Association-) unidirectionnelle (il en faudra donc deux par connexion, une pour chaque sens) préalablement établie entre les correspondants et qui va permettre aux deux parties de convenir des différents paramètres de la SA utilisés durant l'échange des données. Trois paramètres l'identifient :

- un index de paramètres de sécurité (SPI -Security Parameters Index). Il s'agit d'une chaîne de 32 bits de signification locale (propre au système qui gère l'association), véhiculée en clair dans les en-têtes AH et ESP. Une SPI de valeur 0 est un cas particulier pour dire qu'aucune SA n'a été encore créée,
- l'adresse de destination, il peut s'agir d'un système d'extrémité ou d'un système intermédiaire (routeur, firewall ou poste de travail),
- l'identifiant de protocole de sécurité (SPId -Security Protocol Identifier-) qui indique la nature de la SA (AH ou ESP).

Cette association de sécurité contient en plus les paramètres suivants :

- les ports source et destination (peuvent aussi jouer le rôle de paramètres pour identifier la SA),
- l'adresse IP source,
- le nom (user ID ou nom système comme un nom FQDN / X.500, …),
- algorithme d'authentification et clés publiques associées éventuelles,
- algorithme de cryptage et clés publiques associées éventuelles,
- durée de vie de la SA,
- mode (tunnel ou transport),
- numéro de séquence,
- fenêtre anti-rejeu si cette option est activée (cette option est décrite en détail dans la suite),
- débordement du numéro d'ordre (drapeau indiquant si le débordement du numéro de séquence doit produire un évènement d'audit et empêcher toute nouvelle transmission sur cette SA),
- le *Path MTU*. Soient I et R respectivement l'Initiateur et le Répondant du tunnel (soit tout simplement respectivement l'extrémité du tunnel qui va initier le tunnel et l'autre extrémité). I va envoyer un paquet ayant comme taille Max{MTUI, MTUR} avec le bit DF à 1 (bit Don't Fragment -pas de fragmentation-). S'il y a un routeur nécessitant de fragmenter le paquet, il retournera un *ICMP destination inatteignable code 4*, ce qui permettra à I de renvoyer un paquet moins grand. Le processus continue jusqu'à que R recoive le paquet et I n'ait plus de message d'erreur ICMP. la dernière valeur du MTU sera le PMTU, soit la taille maximale des

Tableau 1. *Exemple de SAD avec deux SA*

SPI	N° SA	IP src.	IP dest.	Port src.	Port dest.	SPId	Mode	Type	N° SPD	...
156	1	10.0.0.1	Any	Any	23	AH	Trans-port	Sortant	2	...
23	1	10.0.0.8	10.0.0.5	80	Any	ESP	Tunnel	Entrant	34	...

Tableau 2. *Exemple de SPD*

Règle	IP src.	IP dest.	Port src.	Port dest.	Action	SPId	Mode	N° SPD
1	10.0.0.1	Any	Any	23	IPSec	ESP	Tunnel	234
2	10.0.0.8	10.0.0.5	80	Any	Drop	-	-	412
3	10.2.2.1	10.0.0.5	Any	Any	Accept	-	-	234
4	10.2.2.1	10.0.0.3	Any	Any	Reject	-	-	21

paquets pouvant être envoyés sur le futur tunnel,
- lien vers la SPD. C'est l'identifiant qui va permettre de trouver la correspondance dans la SPD à partir de la SAD (cf. ci-dessous).

Remarques :

- nous parlons d'une SA en général. Il existe des SA IPSec, ISAKMP, TLS…, la SA ISAKMP par exemple n'étant définie que par le SPI et le SPId,
- si ESP et AH sont employés, alors deux SA seront nécessaires, une pour chaque type,
- en général, on n'attend pas la fin d'une SA pour commencer à en négocier une nouvelle : le début de cette nouvelle négociation se fait un peu avant la fin de négociation de l'ancienne (c'est ce qui est fait dans les routeurs CISCO, dans les Firewalls NetASQ ou encore dans le démon IKE Pluto de FreeS/WAN via le paramètre rekeymargin). La version 2 de IKE intègre cette fonctionalité en standard (CREATE_CHILD_SA).

La base des associations de sécurité (SAD -Security Association Database-)

Chaque SA va être contenue dans ce que l'on appelle une base des associations de sécurité (SAD -Security Association Database-). Cette base va contenir pour chaque SA les informations qui lui sont relatives, ce qui permettra de savoir comment traiter chaque paquet à envoyer. C'est une simple base de données qui va être consultée par la SPD. Cette base de données contient toutes les informations de la SA dont la liste a été donnée plus haut.

La base de politique de sécurité (SPD -Security Political Database-)

On définit aussi une base de politique de sécurité (SPD -Security Political Database-), qui va permettre de décider pour chaque paquet entrant ou sortant s'il va se voir attribuer des règles de sécurité et même s'il sera autorisé à passer.

La sécurité avec le mécanisme anti-rejeu

Une attaque par rejeu est une attaque dans laquelle un attaquant obtient une copie d'un paquet, le modifie et le renvoie au destinataire initial. Cette réception peut avoir des effets indésirables, provoquer des perturbations, ou au pire des cas être même pris en compte par le destinataire. Pour éviter cela, si l'option anti-rejeu est sélectionnée, l'émetteur doit s'assurer qu'il n'y a pas de bouclage des numéros de séquence (i.e. dès que le numéro de séquence atteint à 232-1, une nouvelle SA est négociée au lieu de revenir à 0 avec la même SA). Le mécanisme anti-rejeu est donné à la figure ci-dessous :

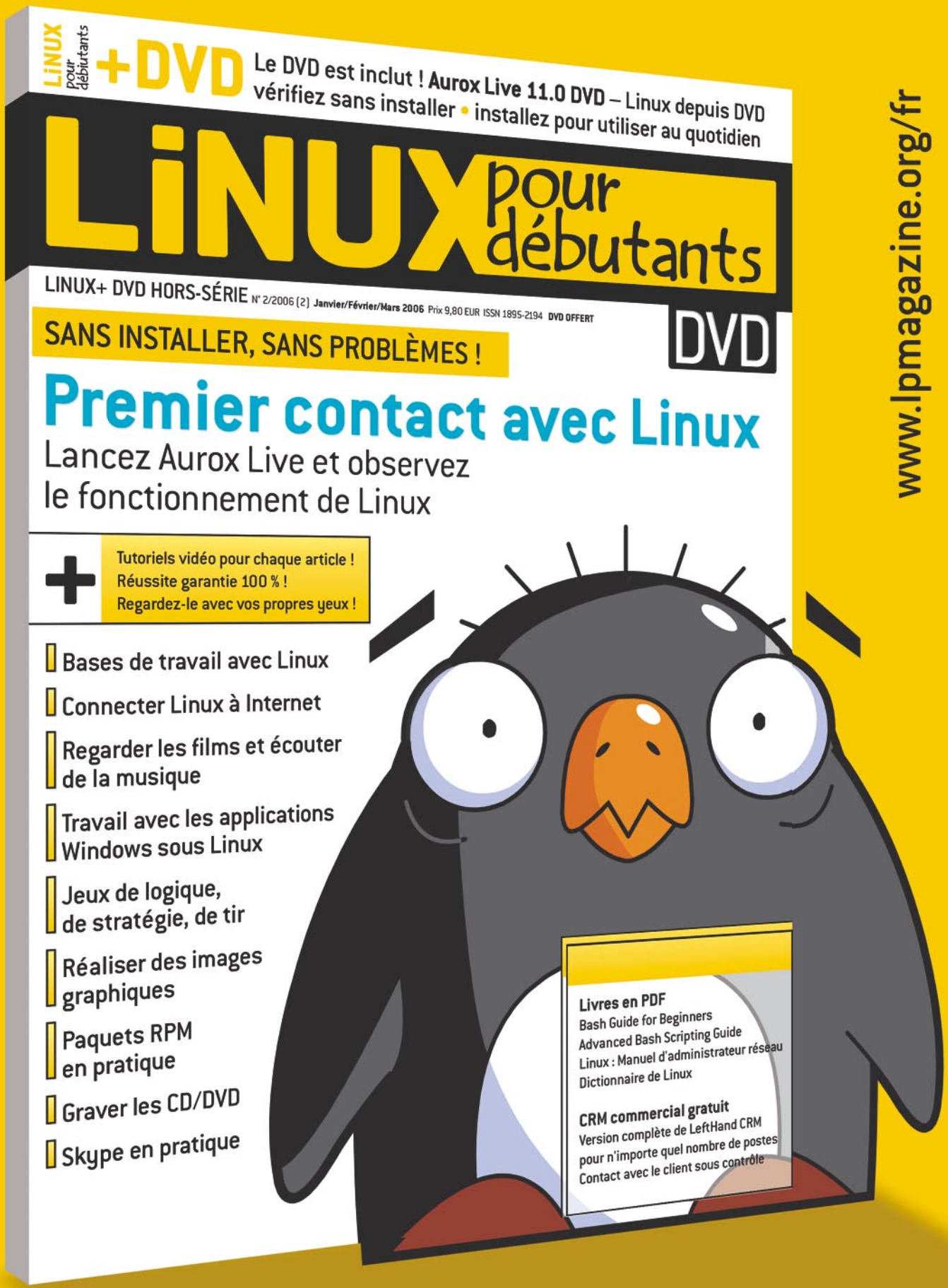
Voici comment il fonctionne : une largeur de fenêtre est établie au départ (lors des négociations de la SA). Le récepteur connaît cette fenêtre qui est un nombre maximum W de paquets IPSec (64 par défaut) car c'est un des renseignements donnés dans la SA correspondante de sa SAD. Cette fenêtre est représentée en vert sur la Figure 1. A un temps *t*, le récepteur va positionner sa fenêtre de manière à ce que celle-ci ait à son extrémité droite ce dernier paquet reçu (noté *N* sur la Figure 1). Pour un paquet arrivant à ce moment-là, nous aurons 3 cas de figure selon son numéro de séquence (appelons ce dernier *n*) :

- soit $n < (N - W)$. Dans ce cas, il est détruit et déclenche éventuellement un audit si le champ correspondant dans la SA le requiert,
- s oit $(N - W) < n < N$. Dans ce cas, il est tout simplement pris en compte et traité (authentification, décryptage, …),
- soit $n > N$. Dans ce cas, la fenêtre avance de manière à que ce dernier paquet se retrouve à son extrémité droite, c'est ce qui est représenté sur la figure ci-dessus, dans la partie du bas.

La sécurité avec l'option PFS

L'option PFS -Perfect Forward Security- est la propriété que la découverte d'un secret à long terme ne compromettra pas les clefs de session qui

Disponible sur **shop.software.com.pl/fr**



www.lpmagazine.org/fr

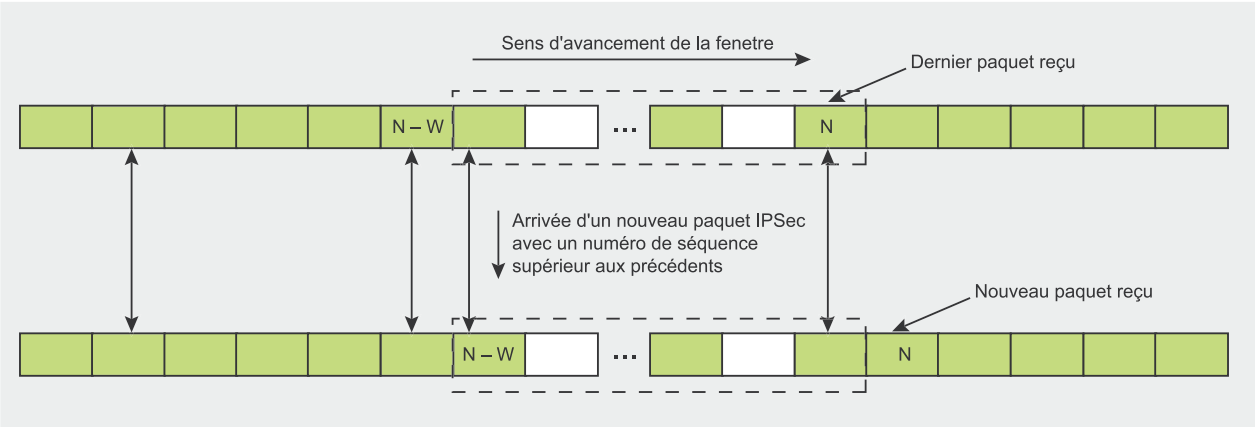


Figure 1. *Mécanisme anti-rejeu avec système de fenêtrage*

auront été dérivées de cette dernière, i.e. le cassage de la clé à long terme ne permet pas d'en déduire les clés de session et donc de déchiffer le trafic crypté avec ces dernières, de même que le cassage d'une clé de session ne permettra pas d'en casser d'autres. Cela se traduit dans les faits par les deux conditions suivantes :

- aucune clé de session (servant donc à crypter des données) ne peut pas être aussi utilisée pour dériver d'autres clés,
- la clé ayant servi à générer la clé de session ne doit pas servir pour d'autres dérivations.

Sous ces conditions, on peut dire que l'option PFS est garantie pour ces deux types de clés, celle de session et celle ayant été utilisée pour la générer.

Tableau 3. *Récaputilatif des services offert par AH et ESP*

	AH	ESP (chiffrement seul)	ESP (chiffrement & authentification)
Contrôle d'accès	Oui	Oui	Oui
Intégrité des données	Oui	Non	Oui
Non-répudiation	Oui	Non	Oui
Anti-rejeu	Oui	Oui	Oui
Confidentialité	Non	Oui	Oui
Confidentialité du flot de trafic	Non	Oui	Oui

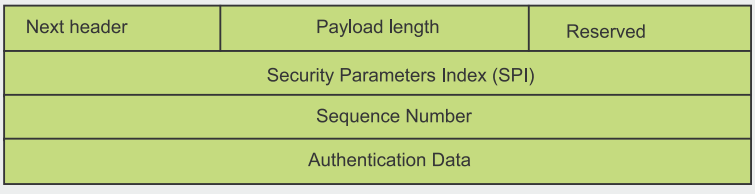


Figure 2. *Format de l'en-tête AH*

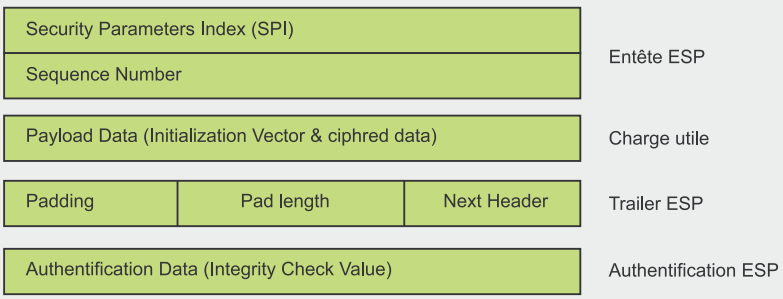


Figure 3. *Format des en-têtes ESP*

Mécanisme de contrôle d'intégrité

Le contrôle d'intégrité se fait via le champ ICV -Integrity Check Value - comme présenté un peu plus bas. Celui-ci est le résultat du hachage de tous les champs de la trame (ceux qui ne sont pas sujets à changement lors du voyage de la trame comme l'adresse réelle source sont gardés tels quels, de même pour ceux dont la valeur à l'arrivée est prévisible comme l'adresse réelle de destination, mais les champs dont la valeur peut changer de manière imprévisible comme la TTL du paquet sont considérés comme nuls pour le calcul de l'ICV), par un algorithme tel que HMAC-MD5 ou HMAC-SHA1.

Les deux modes : tunnel et transport

Dans le mode transport, seules les données en provenance des couches supérieures à la couche IPSec vont être protégées (les données souvent). Ce mode n'est utilisable que entre 2 machines.

Dans le mode tunnel, l'en-tête IP est aussi protégé (que ce soit

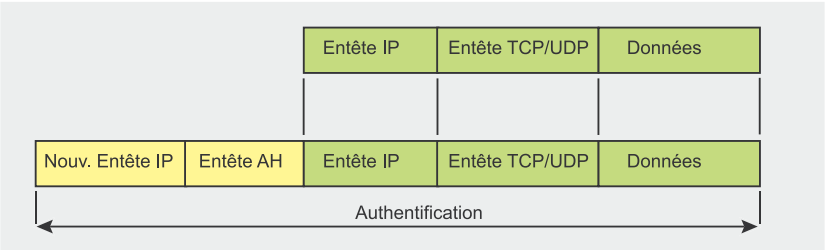


Figure 4. *AH en mode tunnel*

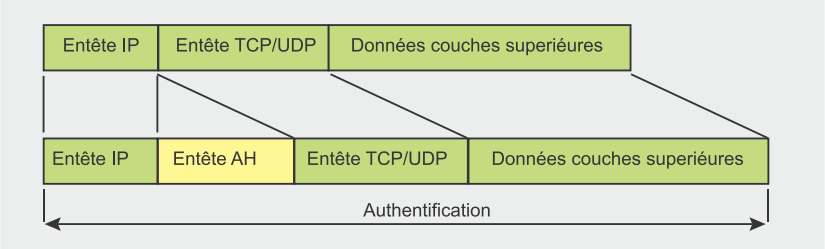


Figure 5. *AH en mode transport*

une simple authentification par vérification de l'intégrité avec AH, ou par cryptage qui va la cacher quand ESP est utilisé) et remplacé par un nouvel en-tête. Ce nouvel en-tête sert à transporter le paquet le long du tunnel, au bout duquel l'ancien en-tête va être rétabli pour pouvoir acheminer le paquet vers sa destination réelle.

Les sous-procoles AH et ESP

Le sous-protocole AH

AH offre les services suivants :

- authentification. On peut savoir si la personne qui dit être l'expéditeur du paquet l'est effectivement ou pas,
- intégrité. L'intégrité est assurée comme on l'a annoncé précédemment par le calcul d'une MAC (paramètre ICV comme nous le verrons ci-dessous et qui est rajouté dans le champ Authentification Data). Elle est étroitement liée avec la non répudiation, et le calcul de la MAC se fait après cryptage des données, ce qui permet du coté du récepteur de vérifier l'authenticité du paquet avant de se lancer pour rien si les paquets ont été altérés, dans la lourde opération de décryptage,
- protection *anti-rejeu* optionnelle. On peut empêcher les attaques de *man-in-the-middle* basiques en numérotant les paquets. Ceci est assuré via le champ *Sequence Number*,
- non-répudiation. Selon les algorithmes utilisés (RSA par exemple).

Elle n'offre cependant pas de confidentialité, i.e. les données peuvent être lues par une tierce personne car elles ne sont pas cryptées.

Sur la Figure 2, nous voyons les champs suivants :

- next Header (32 bits). Champ identifiant l'en-tête suivant,
- payload Length. Ce champ décrit la taille du AH, exprimé en multiples de 32 bits moins 2,
- reserved (16 bits). Ce champ est réservé pour une utilisation ultérieure. Il doit être fixé à 0 sans quoi le paquet est éliminé,
- SPI (32 bits). Nous en avons parlé plus haut. Il est sélectionné par le système de destination car c'est ce dernier qui va en avoir besoin pour savoir comment traiter le paquet qui lui arrive,
- sequence Number (32 bits). Ce champ est le même que celui que l'on trouve dans l'ESP. Ce champ est toujours présent,
- authentication Data (multiple de 32 octets). Ce champ contient la variable ICV et est identique au champ de même nom dans ESP (cf. plus haut). Ce champ peut ne pas être présent si cette option n'a pas été choisie dans la SA correspondante.

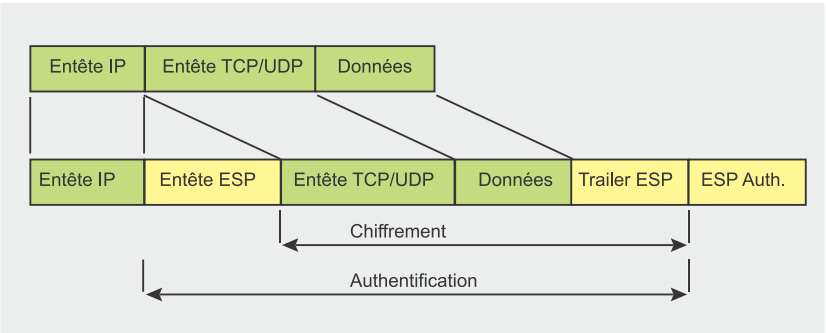


Figure 6. *ESP en mode transport*

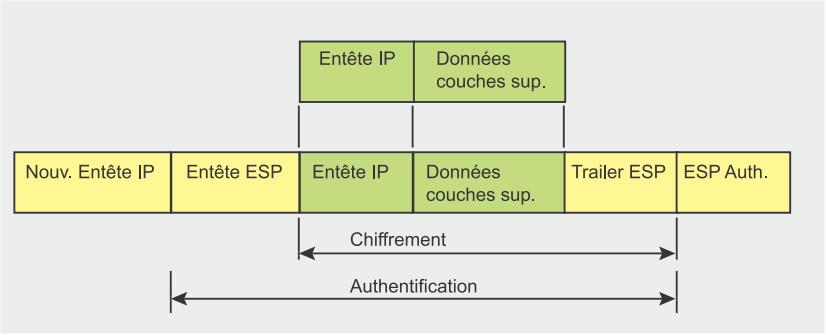


Figure 7. *ESP en mode tunnel*

Pour l'authentification et l'intégrité, les algorithmes possibles sont en général HMAC-RIPEMD-160, HMAC-MD5, HMAC-SHA-1, HMAC-DES, Keyed MD5, ...

Le « sous-protocole » ESP -Encryption Security Payload-

Cette transformation offre en plus ceux de AH, les services suivants :

- confidentialité grâce au cryptage des données. Remarquons la possibilité de choisir un algorithme de chiffrement nul, ce qui revient à ne faire aucun cryptage et qui est donc très dangereux,
- protection des identités. Cette option ne peut être opérationnelle qu'en mode tunnel, et en mode autre que le mode agressif lors de la phase I ISAKMP.

Remarques :

- les fonctionnalités *intégrité* et *non-répudiation* vont de pair, ce qui fait que des fois on appelle *authentification* l'ensemble de ces deux fonctionnalités. Cette dernière fonctionnalité est assurée via le champ ICV -Integrity Check Value- comme nous allons le voir,
- la fonctionnalité de *anti-rejeu* ne peut être sélectionnée que si la *non-répudiation* l'est. Cette dernière est choisie ou non par le récepteur des paquets (en clair comme nous le verrons plus bas, les paquets IPSec contiennent toutes les informations nécessaires via le champ *Sequence Number* pour pouvoir faire la vérification *anti-rejeu*, mais cette vérification n'est faite que si le récepteur l'a décidé),
- avec ESP, même si authentification et confidentialité sont toutes les deux des options, au moins l'une des deux doit être sélectionnée (en effet, même si ESP demande forcément de choisir un algorithme de cryptage, on a toujours la possibilité de choisir l'algorithme nul… ce qui revient à ne pas appliquer de confidentialité).

Dans la Figure 3, nous voyons les champs suivants :

- SPI (32 bits). Nous en avons parlé plus haut. Il est sélectionné par le système de destination car c'est ce dernier qui va en avoir besoin pour savoir comment traiter le paquet qui lui arrive. Ce champ est toujours présent,
- sequence Number (32 bits). Chaque paquet est numéroté par ce champ de 32 bits. Ce champ est initialisé à 0 dès qu'une nouvelle SA est établie, et le premier paquet envoyé sur le réseau aura un numéro de séquence de 1. Incrémenté de 1 à chaque nouveau paquet envoyé, sa limite sera donc de 232. Deux cas apparaissent alors quand cette limite est atteinte : soit la protection anti-rejeu est activée par le récepteur auquel cas une nouvelle SA est générée avant que ne numéro de séquence maximum de 232 est atteint et le compteur du numéro de séquence est réinitialisé à 0 ; soit cette protection n'est pas activée et dans ce cas, on reprend la numérotation des paquets à 1 avec la même SA. Cette option est toujours mise en place par l'émetteur, mais ne sera vérifiée et prise en compte par le récepteur que si ce dernier le souhaite (donc si cette option est choisie de son côté). En pratique, durant la phase d'échange de paramètres de SA, le récepteur peut dire à l'émetteur s'il a activé l'option anti-rejeu, ce qui évite à l'émetteur de faire du travail inutile. D'autre part, cette option ne peut être activée que si la non répudiation l'est aussi. Ce champ est toujours présent,
- payload Data (0-255 bits). Nous trouvons dans ce champ, si l'algorithme choisi le nécessite (DES par exemple), le paramètre IV -Initialization Vector-. Ce champ est toujours présent,
- padding (0-255 bits). La nécessité du bourrage intervient lors de l'utilisation d'algorithmes de cryptage nécessitant des chiffrement par blocs comme DES par exemple. Dans ce cas, il arrive souvent que la longueur des données à chiffrer ne soit pas un multiple entier de cette longueur de bloc, on rajoutera du bourrage de manière à avoir une longueur à crypter qui soit un multiple entier de la longueur du bloc. Ce champ est …très souvent présent !
- pad Lengh. Dans ce champ, nous trouvons la longueur du champ

Tableau 4. *Récapitulation des fonctionnalités des modes transport et tunnel*

	Mode transport	Mode tunnel
AH	Authentifie la charge utile IP et certains champs de l'en-tête IP et les en-têtes d'extension IPv6.	Authentifie le paquet IP tout entier (en-tête plus certaines informations IP) plus certains champs de l'en-tête IP externe et des en-têtes d'extension IPv6 externes.
ESP (chiffrement seul)	Chiffre la charge utile IP et tout en-tête d'extension IPv6 suivant l'en-tête ESP.	Chiffre le paquet IP tout entier.
ESP (chiffrement & authentification)	Chiffre a charge utile IP et tout en-tête d'extension IPv6 suivant l'en-tête ESP. Authentifie la charge utile IP mais pas l'en-tête IP.	Chiffre le paquet IP tout entier. Authentifie le paquet IP.

Chez votre marchand de journaux

Également disponible sur **www.buyitpress.com**

précédent, ce qui nous permet de savoir quels bits sont à ignorer (ceux de bourrage). Ce champ est toujours présent,

- next Header (8 bits). Ce champ permet de savoir quel est le type d'informations contenues dans le champ « Payload Data » ; IPv4/IPv6, ICMP, IP, IGRP, ... Ce champ est toujours présent,
- authentication Data (variable). Ce champ contient la variable ICV qui est calculée sur toute la trame moins ce champ-ci (i.e. « Authentication Data ») et qui permet donc d'assurer l'intégrité des données transmises. Ce champ peut ne pas être présent si cette option n'a pas été choisie dans la SA correspondante.

Pour le chiffrement, les algorithmes valides pour une négociation sont par exemple : DES CBC, Triple DES, RC 5, IDEA & IDEA Triple, Blowfish, CAST, NULL (ce n'est pas une blague, la possibilité de ne spécifier aucun chiffrement peut être parfois utile ... mais très dangereuse aussi), ...

Comme nous le voyons, nous n'avons que des algorithmes symétriques, ce qui est expliqué par le fait que le chiffrement des données par des algorithmes asymétriques demanderaient beaucoup plus de temps et de ressources machines.

Pour l'authentification et l'intégrité, des algorithmes possibles sont : HMAC-RIPEMD-160, HMAC-MD5, HMAC-SHA-1, HMAC-DES, Keyed MD5, NULL (même remarque que précédemment), ...

Remarques :

- dans le cas où l'authentification et le chiffrement sont sélectionnés, le chiffrement se fait avant l'authentification. Ceci car dans le cas, il est plus facile de découvrir que les données ont été altérées (il suffit de lire le ICV), alors que si on authentifie avant de crypter, il faudrait décrypter d'abord pour pouvoir lire le ICV et voir si les données ont été altérées ou non. De plus cela permet de réduire les risque d'une attaque DOS (l'acceptation se fait plus rapidement comme nous venons de le voir dans le premier cas), et permet également le traitement en parallèle des paquets reçus (pendant que le paquet I est déchiffré après avoir été détecté comme « bon », le paquet u+1 va passer à la détection via lecture de son ICV),
- en général, pour des tunnels en point-à-point, les HMACs sont préférés. Pour des connexions multicast (par exemple un serveur central VPN qui fait office de plaque tournante pour plusieurs VPNs qui partent de lui), on préfèrera les fonctions de hachage basées sur des algorithmes asymétriques.

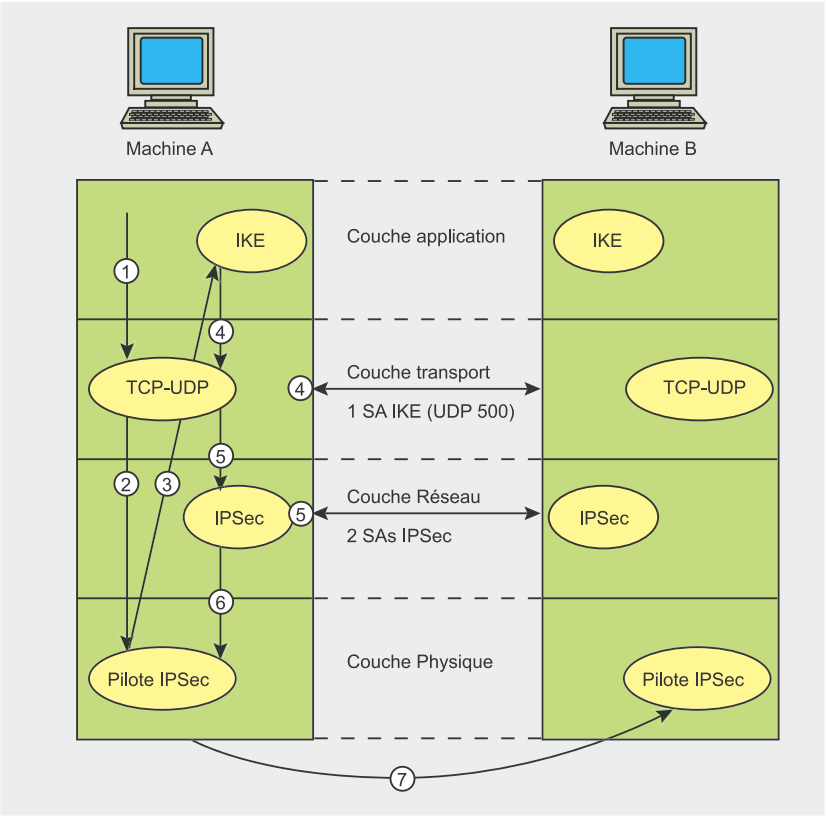


Figure 8. *Gestion du trafic IPSec vu du modèle OSI*

Déchiffrement du trafic entrant

Lorsque la couche IPSec reçoit un paquet remonté du réseau, elle va regarder les entêtes pour voir si le paquet a été sécurisé, et si oui, quelles sont les caractéristiques de la SA. Elle demande ensuite à la SAD les caractéristiques de cette SA pour décrypter/authentifier le paquet. Une fois déchiffré, la SPD sera consultée pour vérifier que la SA associée au paquet correspondait bien aux politiques de sécurité.
Nous avons donc dans l'ordre les étapes suivantes pour traiter un paquet entrant :

- réassemblage-. Ce qui se fait dans la plupart des cas à cause de la fragmentation lors du voyage au travers des réseaux,
- lecture de la SAD-,
- vérification du numéro de séquence-,
- vérification du champ ICV-,
- lecture de la SPD-,
- décryptage-,
- décompression éventuelle-. La décompression doit être réalisée après tout traitement (et non avant comme pour les paquets sortants) comme le décryptage, l'authentification, …

Récapitulatif des services offerts AH et ESP

Remarque : Si ESP et AH doivent être appliqués au même paquet, ESP sera fait avant AH.

Les 4 possibilités pour IPSec

- 1ère possibilité : AH en mode transport,
- 2ème possibilité : AH en mode tunnel,
- 3ème possibilité : ESP en mode transport. Les deux en-têtes importants sont le ESP Header qui contient le SPI et le numéro de séquence ; et l'authentification ESP qui contient les données d'authentification,
- 4ème possibilité : ESP en mode tunnel. Dans l'en-tête *New IP Header*, nous avons l'en-tête IP *temporaire* contenant l'adresse IP du routeur ou de l'équipement vers lequel la trame est envoyée au cours de son voyage. Le champ suivant est l'en-tête ESP (ESP Header) qui contiendra la SPI associée à la SA, ainsi que le numéro de séquence. À droite, nous avons l'en-tête ESP Authentification qui va contenir les données d'authentification.

Récapitulation des fonctionnalités des modes transport et tunnel

En utilisant ESP, il est possible quoique non recommandé d'utiliser le cryptage sans authentification.

La configuration avancée de IPSec : strict, claim, exact et obey

On peut aussi conditionner le comportement du serveur IPSec en phase 1 lors de la négociation des options PFS et durée de vie de SA (dans le but d'accélérer les négociations en les restreignant par exemple) :

- strict. Ce mode n'accepte que les options égales ou plus strictes que les siennes (PFS plus élevée, durée de vie de SA plus courte),
- claim. Ce mode n'accepte que les options égales ou moins strictes que les siennes (PFS moins élevée, durée de vie de SA plus longue),
- exact. Ce mode n'accepte que les options aussi strictes que les siennes (même niveau de PFS, durée de vie de SA strictement égale),
- obey. Ce mode accepte les options quelles qu'elles soient (niveau de PFS, durée de vie de SA).

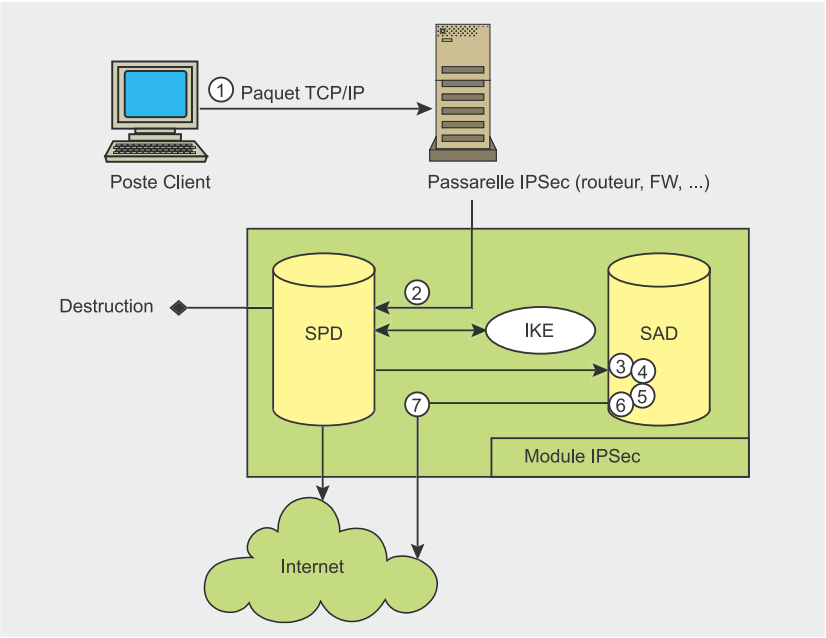


Figure 9. *Gestion du trafic sortant avec IPSec*

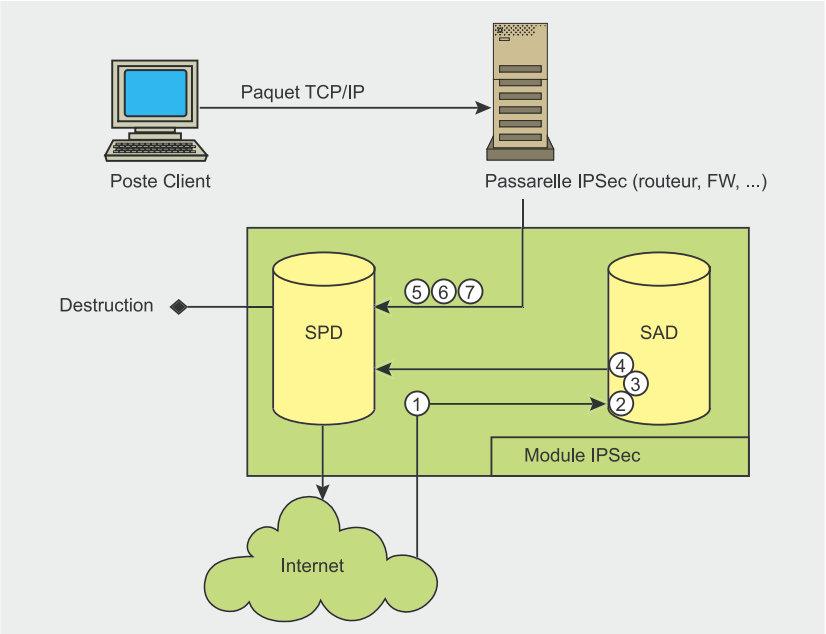


Figure 10. *Gestion du trafic entrant avec IPSec*

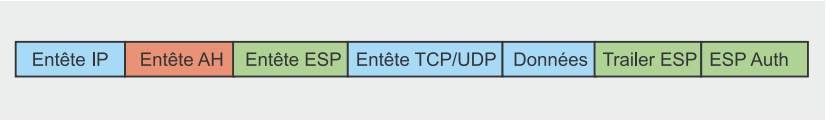


Figure 11. *Contiguïté en mode transport*

Chiffrement du paquet sortant

Lorsque un paquet à envoyer est transmis à la couche IPSec, celle-ci consulte la SPD pour savoir comment traiter ces données car elle a trois choix :

- destruction. Le paquet est tout simplement détruit,
- transmission sans sécurisation. Le paquet est transmis sans appliquer de politique de sécurité,
- transmission avec sécurisation. Le noyau applique une politique de sécurité.

Dans tous les cas, c'est la SPD qui gère cela : elle prend le numéro de SA correspondant et va en chercher les caractéristiques dans la SAD. Si la SA existe déjà, le trafic se voit appliquer ces mécanismes, si la SA n'existe pas encore, IPSec fera appel à IKE pour établir une nouvelle SA avec les caractéristiques demandées.

Nous avons donc dans l'ordre les étapes suivantes pour traiter un paquet sortant :

- lecture de la SPD-. En fonction des adresses source et destination, et des ports source et destination, la SPD nous donne les règles pour le paquet correspondant : soit il est détruit, soit il est transmis sans faire intervenir IPSec, soit il est traité avec IPSec. Dans ce dernier cas, nous savons aussi le sous-protocole (AH/ESP) et le mode (tunnel/transport) à utiliser, ainsi que la SA correspondante. S'il n'y a pas de SA correspondante, on passe le relais à IKE pour en créer une,
- lecture de la SAD-. La SPD indiquant aussi la SA correspondante dans la SAD, on va ensuite chercher dans cette dernière les options de transfert (algorithmes de cryptage, authentification, durée de vie de la SA, …),
- compression éventuelle-. La compression (avec le protocole IPComp) doit être faite avant tout traitement IP (authentification, cryptage, fragmentation,…),
- cryptage-. Avec les informations précédentes de la SAD, on peut maintenant crypter la partie de la

Tableau 5. *Notations IKE*

SA	Ce sont les propositions de la SA : l'Initiateur propose un choix d'algorithmes, et le Récepteur renvoi la combinaison choisie.
CKY_X	Ce sont les cookies de l'Initiateur (CRY_I) et du Récepteur (CRY_R) placés dans l'en-tête ISAKMP.
HASH	C'est la charge utile du hachage : HASH_I précise que c'est le hachage envoyé par l'Initiateur et HASH_R celle envoyée par le Récepteur. Elle authentifie la charge utile IP mais pas l'en-tête IP.
gxi, gxr	Ce sont les valeurs publiques de Diffie-Hellman, respectivement de l'Initiateur et du Récepteur.
gxy	C'est la clé secrète obtenue par échange Diffie-Hellman.
No_I, No_R	Ce sont les aléas, respectivement générés par I et R.
ID_I, ID_R	Ce sont les identités utilisées pour l'authentification, respectivement de I et de R.
X*	Signifie que le champ X est crypté

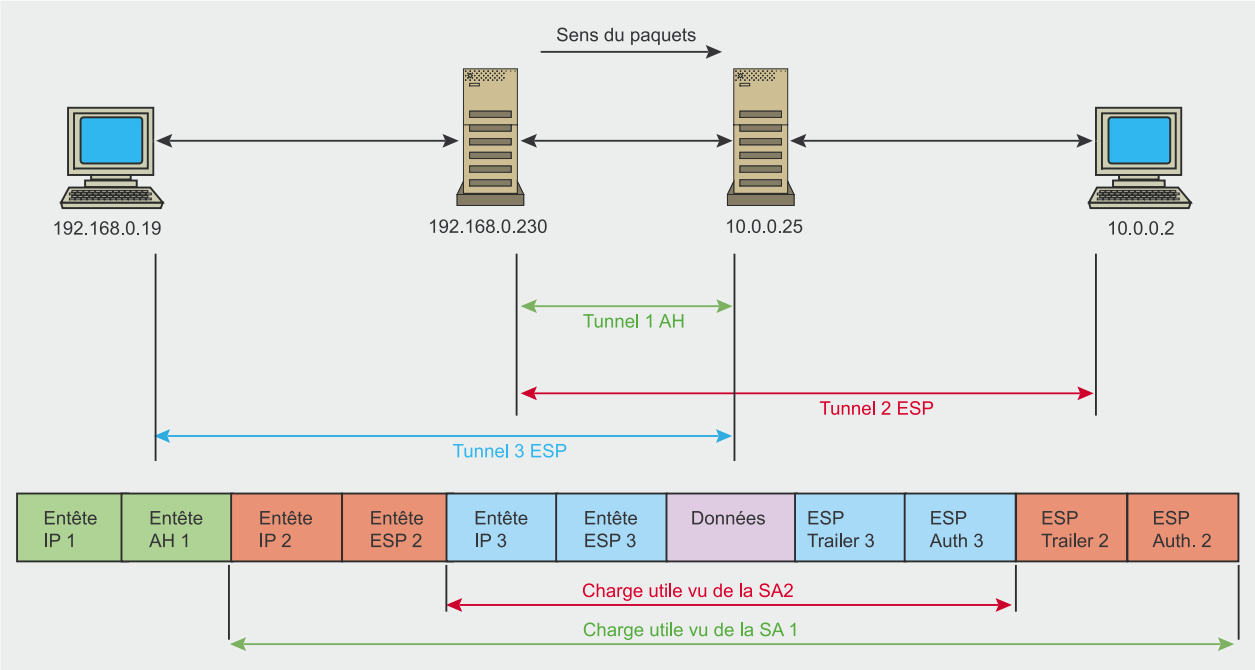
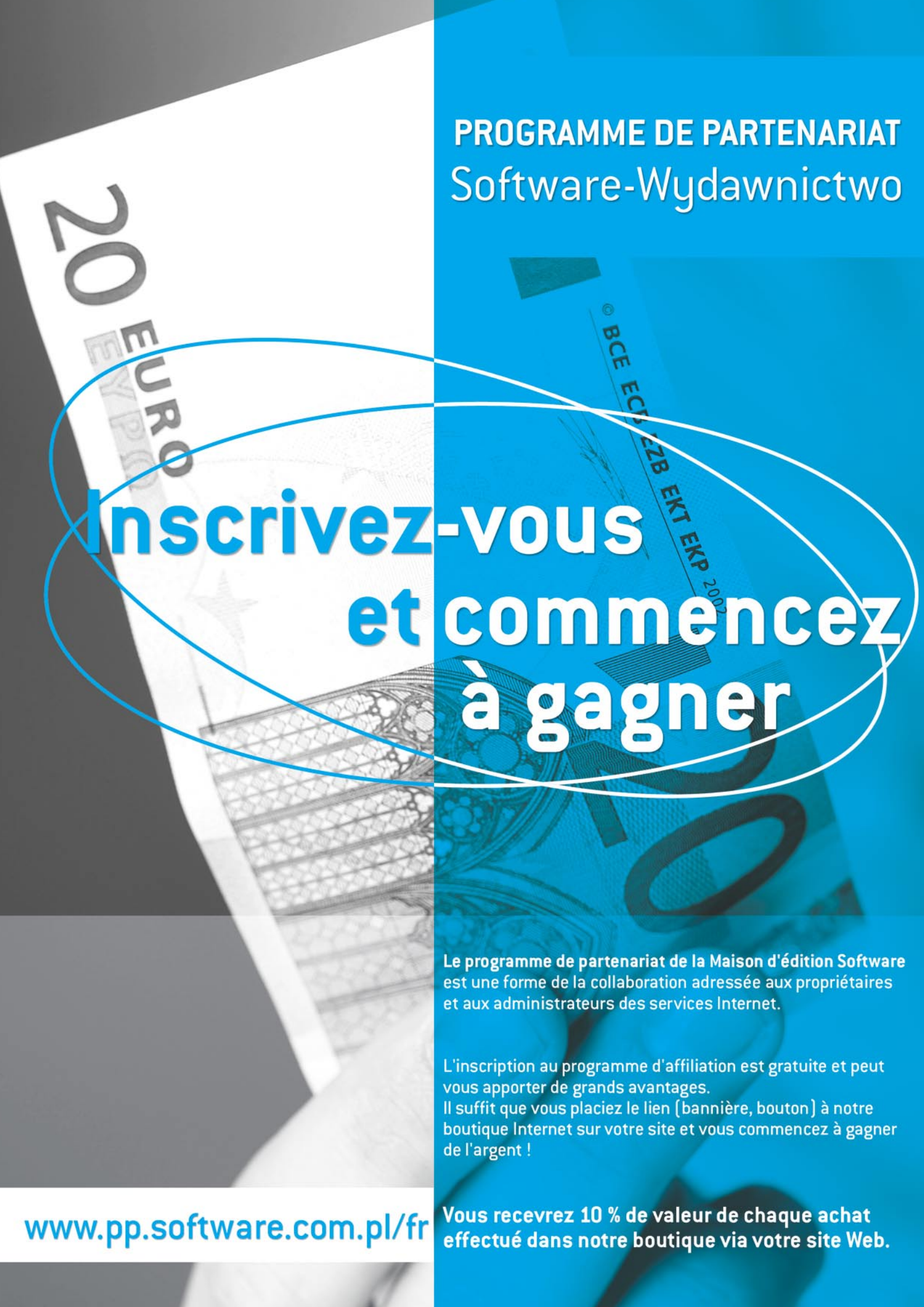


Figure 12. *Tunnels itérés*



PROGRAMME DE PARTENARIAT
Software-Wydawnictwo

Inscrivez-vous et commencez à gagner

Le programme de partenariat de la Maison d'édition Software est une forme de la collaboration adressée aux propriétaires et aux administrateurs des services Internet.

L'inscription au programme d'affiliation est gratuite et peut vous apporter de grands avantages.
Il suffit que vous placiez le lien (bannière, bouton) à notre boutique Internet sur votre site et vous commencez à gagner de l'argent !

Vous recevrez 10 % de valeur de chaque achat effectué dans notre boutique via votre site Web.

www.pp.software.com.pl/fr

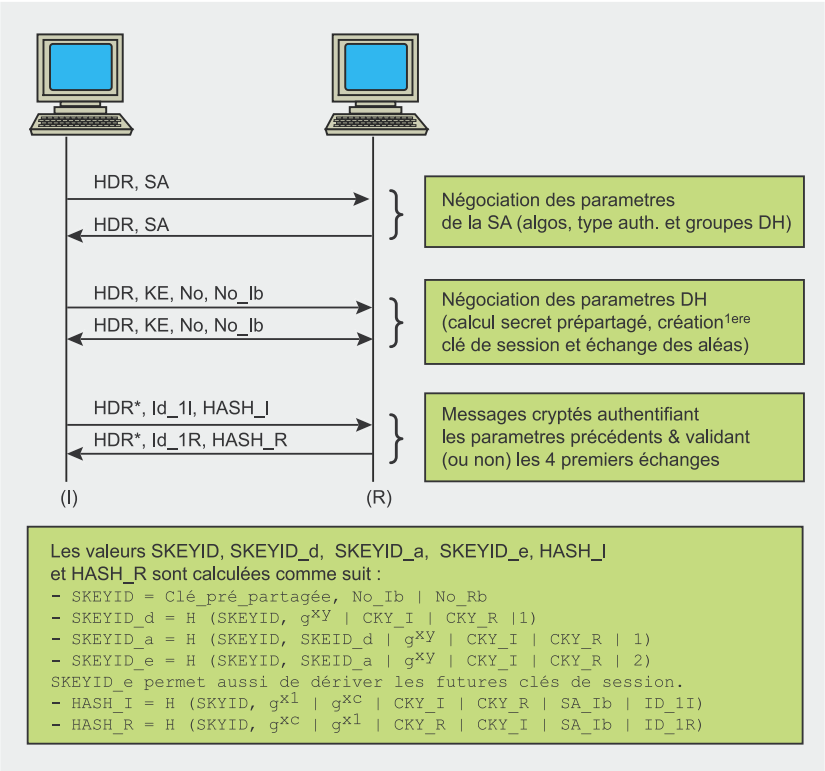


Figure 13. *Phase 1 : Les 6 échanges en mode principal*

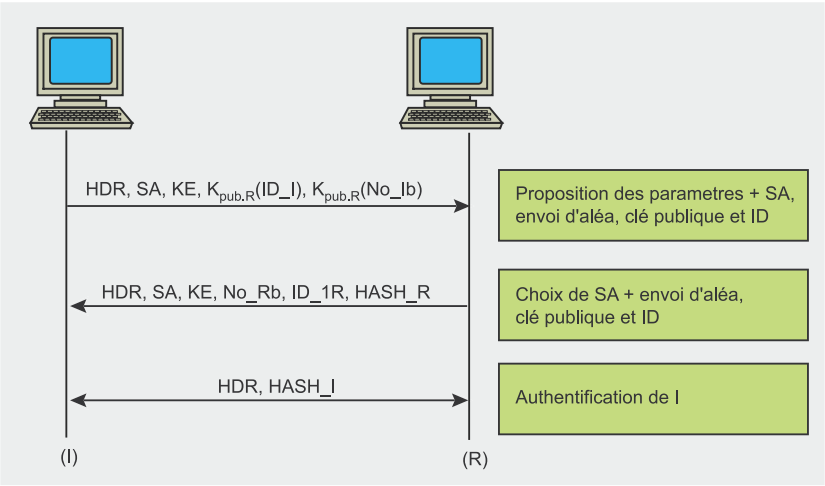


Figure 14. *Phase 2 : Les 3 échanges en mode rapide*

requête nécessaire (cette partie dépend comme nous l'avons vu du mode et du sous-protocole comme vu plus haut). Le cryptage se fait en 3 étapes principales : encapsulation AH/ESP, rajout de bourrage si nécessaire et finalement cryptage,

- création du numéro de séquence-. On rajoute le numéro de séquence de la requête en cours (dans la SA, on trouve le numéro de séquence du paquet précédent ayant utilisé la même SA, il suffit donc de l'incrémenter de 1) dans l'en-tête AH/ESP pour permettre le réassemblage et permet au récepteur de vérifier qu'il n'y a pas eu de rejeu de paquets (si ce dernier a activé cette option de son côté),
- création du champ ICV-. Création de ce champ permettant l'authentification qui va permettre au récepteur de vérifier que le paquet n'a pas été altéré en cours de route (intégrité). Cette valeur prend en compte les champs comme vu précédemment,
- fragmentation-. La SAD contenant aussi la PMTU (comme nous l'avons décrit plus haut), nous saurons si nous avons besoin ou non de fragmenter le paquet avant de l'envoyer sur le réseau.

Cas de plusieurs SAs concurrentes (SA's bundles)

Cas 1. Contiguïté en mode transport (transport adjency). Ce mode permet d'appliquer à la fois AH et ESP, mais n'est possible que en mode transport comme le montre la Figure 11 (ce qui le rend donc rarement utilisé).

Cas 2. Itérations de tunnels. Ce mode permet de monter des tunnels se recoupant entre les deux extrémités finales, comme le montre la Figure 12. Par exemple, si nous avons un FreeS/WAN installé sur la machine 192.168.0.230, l'itération sera faite comme suit :

```
[root@c0rt0W1nch] # ipsec spigrp
inet 10.0.0.2 0x3c1691a1 esp inet
10.0.0.25  0x432d3446
```

Exemple de montage d'un tunnel IPSec classique

Ce processus se compose de 2 phases. Nous les presenterons ci-dessous.

Phase 1 : Les 6 échanges en mode principal

Cette phase a 3 objectifs :

- négociation des paramètres de sécurité. Les deux extrémités du tunnel doivent se mettre d'accord sur les paramètres qui vont être utilisés pour crypter les deux points suivants de la phase 1, ainsi que toute la phase 2. Ces paramètres sont les clés de chiffrement, les algorithmes et la méthode d'authentification (clés pré-partagées, certificat,…),
- établissement de la clé pré-partagée,
- authentification des utilisateurs.

Disponible sur : shop.software.com.pl/fr

www.lpmagazine.org/fr

Au cours de cette phase, on a deux modes Oakley possibles :

- mode principal. Ce mode protège l'identité des deux parties et se fait en 6 messages. Les deux premiers permettent de négocier la politique de sécurité, les deux suivants échangent la clé partagée de Diffie-Hellman et éventuellement toute autre donnée auxiliaire pour cet échange, tandis que les deux derniers messages permettent l'authentification,
- mode agressif. Ce mode ne protège pas l'identité des deux parties et se fait en 3 messages (plus rapide donc). Les deux premiers permettent non seulement comme précédemment de se mettre d'accord sur la politique de sécurité à adopter, mais aussi permettent l'échange de Diffie-Hellman, le transfert de toute autre donnée nécessaire pour cet échange et l'échange des identités des deux parties. Le second message permet aussi en plus d'authentifier la machine serveur (donc pas celle qui initie la connexion, mais l'autre). Le troisième message identifie principalement l'initiateur de la connexion.

Phase 2 : Les 3 échanges du Mode Rapide

Dans cette phase, tous les échanges sont protégés avec les clés échangées lors de la phase 1. Cette phase permet de monter la négociation de la SA Ipsec :

- paramètres (protocole ESP ou AH, algorithme d'authentification (SHA1 ou MD5), et algorithme de chiffrement (si ESP),
- clés à utiliser pour la protection des paquets IP.

Au cours de cette phase, on a deux options pour la génération des clés IPSec :

- mode de base. Dans ce mode, les clés sont celles qui ont été générées lors de la phase 1,
- perfect Forward Secrecy. Dans ce mode, un nouvel échange Diffie-Hellman permet de générer de nouvelles clés IP.

Conclusion

IPSec reste le plus utilisé en matière de VPN grâce aux avantages que nous avons vu : flexibilité et modularité, sécurité totalement transparente pour les applications,

Cependant IPSec reste très complexe (et on dit souvent que la complexité est l'ennemie de la sécurité), pose des soucis de NAT, et reste victime des entorses propriétaires qui nuisent à l'interopérabilité.

Tous les listings sont à consulter sur *www.hakin9.org/fr* (l'onglet : listings). ●

Sur Internet

- J. PLIAM, Authentication Vulnerabilities in IKE and Xauth with Weak Pre-Shared Keys,
- P. KNIGHT, Dynamic Routing inside IPsec VPNs, Nortel networks, BlackHat 2002,
- G. LABOURET, IPSEC : Présentation technique, Hervé Schauer Consultants, 2000,
- *http://www.kb.cert.org/vuls/id/886601* – CERT Coordination Center (CERT/CC), Vulnerability Note VU#886610, Carnegie Mellon Software Engineering Institute,
- J. CHIRILLO, Hack Attacks Revealed, Washington D.C., Wiley, 2002,
- *http://www.cisco.com/warp/public/707/cisco-sn-20030422-ike.html* - Cisco Systems, Cisco Response to Internet Key Enchange Issue, 2003,
- *http://www.nta-monitor.com/ike-scan/whitepaper.pdf* – R. HILL. NTA Monitor UDP Backoff Pattern Fingerprinting White Paper, NTA Monitor LTD, 2003,
- *http://www.ima.umn.edu/~pliam* - J. PILAM, Authentication Vulnerabilities in IKE and Xauth with Weak Preshared Secrets, Institute for Mathematics and its Applications,
- M. THURMAN & R. ENNO, PSK Cracking Using IKE Aggressive Mode, ERNW Enno Rey Netzwerke GmbH,
- RFC 1828. IP Authentication using Keyed MD5,
- RFC 2202. Test Cases for HMAC-MD5 and HMAC-SHA-1,
- RFC 2401. Security Architecture for the Internet Protocol,
- RFC 2402. IP Authentication Header -AH-,
- RFC 2403. The Use of HMAC-MD5-96 within ESP and AH,
- RFC 2404. The Use of HMAC-SHA-1-96 within ESP and AH,
- RFC 2405. The ESP DES-CBC Cipher Algorithm with Explicit IV,
- RFC 2406. IP Encapsulating Security Payload (ESP),
- RFC 2407. The Internet IP Security Domain of Interpretation for ISAKMP,
- RFC 2408. Internet SA and Key Management Protocol (ISAKMP),
- RFC 2409. The Internet Key Exchange (IKE),
- RFC 2410. The NULL Encryption Algorithm and Its Use With Ipsec,
- RFC 2411. IP Security Document Roadmap,
- RFC 2412. The OAKLEY Key Determination Protocol,
- RFC 2522. Photuris - Session-Key Management Protocol,
- RFC 2709. Security Model with Tunnel-mode IPSec for NAT Domains,
- RFC 3173. IP Payload Compression Protocol (IPComp),

À propos de l'auteur

Ayant travaillé dans le domaine de la sécurité depuis plus de 4 ans maintenant, d'abord pour des banques puis chez un constructeur de VPN et de Firewalls, l'auteur est actuellement expatrié au Gabon comme Architecte de Systèmes d'Information pour un opérateur de téléphonie mobile. Son travail lui a permis plusieurs approches de la sécurité : développement d'applications, sécurisation de réseaux, sécurisation de portails Internet et Intranet, ... Il passe son temps libre sur son site web personnel traitant de cryptographie, sécurité, télécommunications, réseaux et de physique.