

IP security (IPsec) / SSL

Plan

- IPsec
 - Rappels et Présentation
 - Services
 - Architecture
 - Protocole AH
 - Protocole ESP
 - L'association de sécurité
 - Les politiques de sécurité
 - Protocole IKE
 - Conclusions
- SSL
 - Introduction
 - Les sous-protocoles de SSL

Rappels:

Qu'est-ce que la sécurité des réseaux ?

Confidentialité : seuls l'émetteur et le récepteur visé doivent pouvoir comprendre le contenu du message

- L'émetteur chiffre le message
- Le récepteur déchiffre message

Authentification : l'émetteur et le récepteur veulent confirmer l'identité de leur correspondant

Intégrité et non répudiation : l'émetteur et le récepteur veulent s'assurer que le message n'a pas été altéré (durant le transit, ou après) sans que cela n'ait été détecté. On peut prouver que le message vient bien de l'émetteur

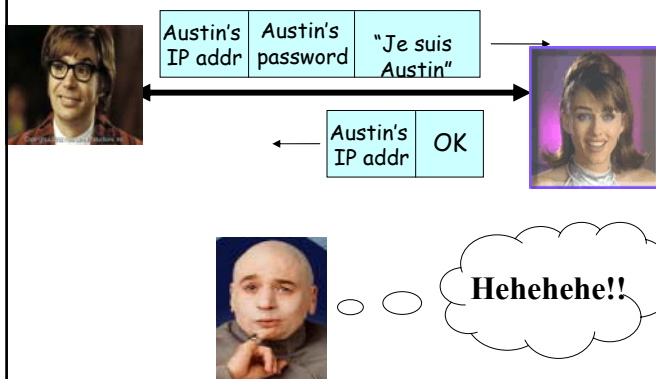
Accès et disponibilité : les services doivent être accessibles et disponibles pour les utilisateurs

Rappels: attaques possibles

- **Écoute (eavesdrop)** : interception de messages
- **Insertion** active de messages dans la connexion
- **Imitation** : falsification (spoof) de l'adresse source dans le paquet (ou tout autre champ du paquet)
- **MITM** : "prise de contrôle" de la connexion en cours en écartant l'émetteur ou le récepteur et en prenant sa place
- **DOS** : empêcher le service d'être utilisé par les autres (ex : en surchargeant les ressources)
- **Rejeu**

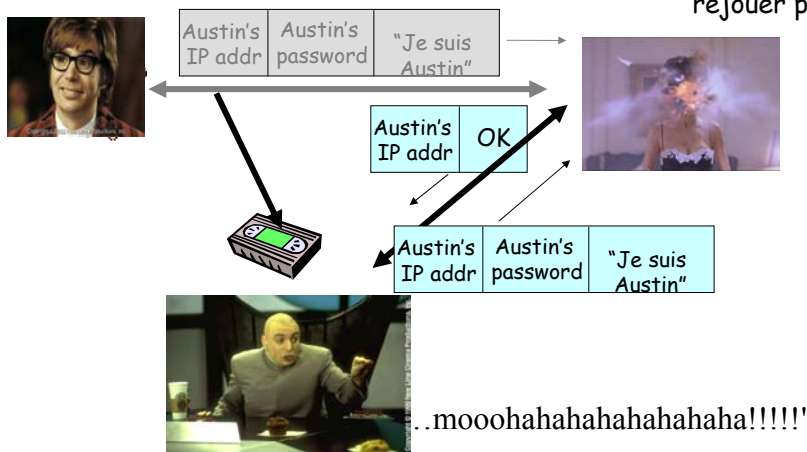
Rappels: Authentification : Protection contre le rejeu

Austin dit "Je suis Austin" et envoie son mot de passe pour le prouver.



Rappels: Authentification : Protection contre le rejeu

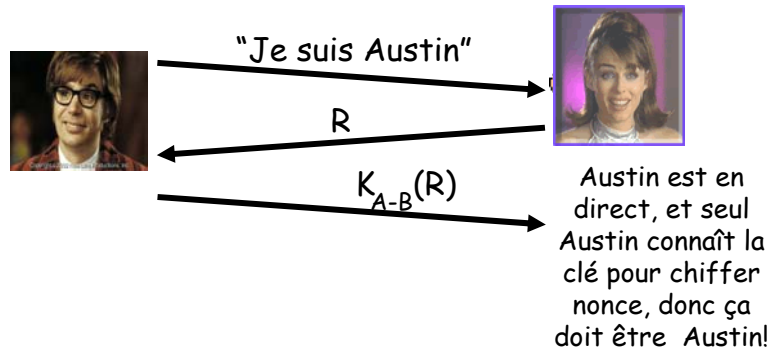
Attaque par rejeu :
Enregistrement du paquet d'Austin pour le rejouer plus tard



Rappels: Authentification : Protection contre le rejeu

Nonce: nombre (R) utilisé *seulement-une-fois*

Vanessa envoie à Austin un **nonce**, R. Austin doit renvoyer R, chiffré avec la clé secrète



Rappels: Signatures numériques

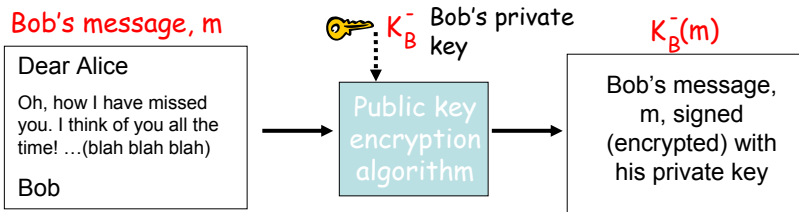
Technique cryptographique analogue aux signatures manuscrites.

- L'émetteur (Bob) signe le document de manière numérique et établit qu'il est le créateur/propriétaire du document.
- **Vérifiable, non falsifiable**: le récepteur (Alice) peut prouver à quelqu'un que Bob et personne d'autre (y compris Alice) a signé ce document

Rappels: Signatures numériques

Signature numérique simple pour le message m :

- Bob signe m en chiffrant avec sa clé privée K_B^- , créant le message "signé" $K_B^-(m)$



Rappels: Signatures numériques (suite)

- Alice
 - reçoit le msg m et la signature numérique $K_B^-(m)$
 - vérifie que m a été signé par Bob en appliquant la clé publique de Bob K_B^+ à $K_B^-(m)$ et vérifie que $K_B^+(K_B^-(m)) = m$.
- Si $K_B^+(K_B^-(m)) = m$, la personne qui a signé m a forcément utilisé la clé privée de Bob.

Alice vérifie ainsi que :

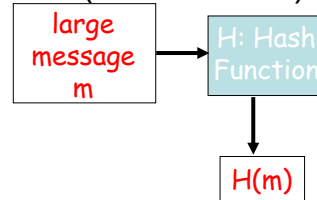
- Bob a signé m .
- Personne d'autre n'a signé m .
- Bob a signé m et pas m' .

Non-répudiation:

- ✓ Alice peut emporter m et la signature $K_B^-(m)$ à un procès et prouver que Bob a signé m .

Rappels: Messages condensés (Condensats)

Le chiffrement par clé publique de longs messages est très onéreux "computationnellement"



But : "empreinte digitale" de longueur fixe et facile à calculer

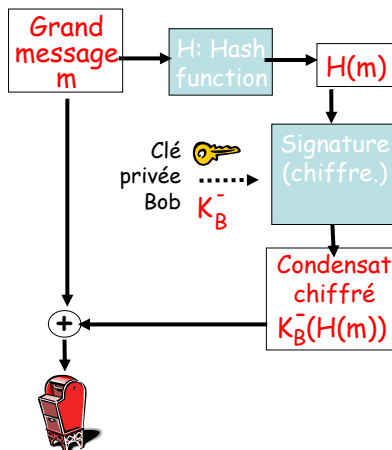
- Appliquer une fonction de hachage H à m , recevoir un message condensé de longueur fixe, $H(m)$.

Propriétés de la fonction de hachage:

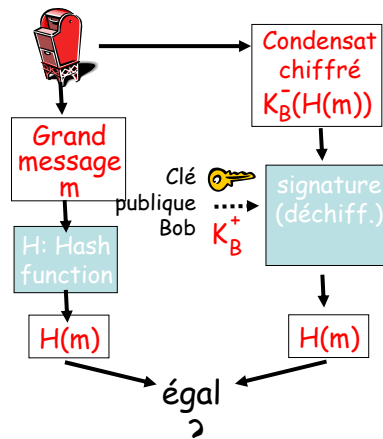
- Produit des messages condensés de taille fixe ("empreinte digitale")
- Étant donné un message condensé x , il est computationnellement impossible de trouver m tel que $x = H(m)$

Rappels: Signature numérique = condensat signé

Bob envoie un message signé numériquement :



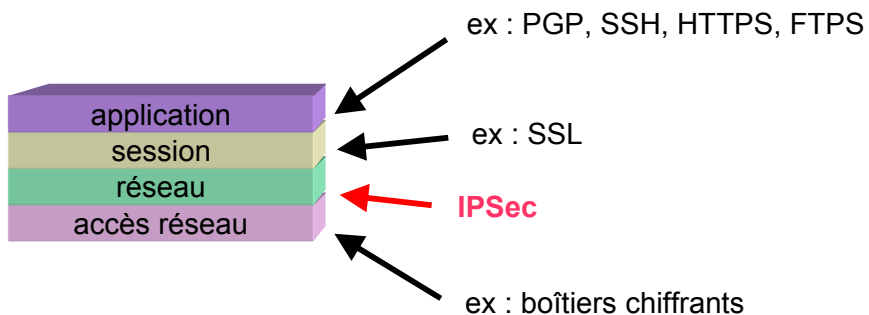
Alice vérifie la signature et l'intégrité du message signé numériquement :



Rappels: Algorithmes de fonctions de hachage

- **Fonction de hachage MD5 largement utilisée (RFC 1321)**
 - Calcule un condensat x de 128 bits.
 - Il est difficile, à partir d'une chaîne aléatoire de 128 bits, de construire un msg m dont le hash MD5 est égal à x.
- **SHA-1 est également utilisé.**
 - Standard américain [NIST, FIPS PUB 180-1]
 - Condensat de 160 bits

Où appliquer la sécurité ?



Motivations de IPSec

- IPSec (IP Security) est intégré dans IPv6
- Motivations de IPv6
 - Grande capacité d'adressage (128 bits)
 - Simplification du routage
 - Sécurisation des communications (IPSec)
 - QoS (pas vraiment)
 - Protocole et architecture pour la mobilité
- 6bone : réseau mondial d'expérimentation IPv6
- Stratégie de migration en cours de développement

Standardisation de IPSec

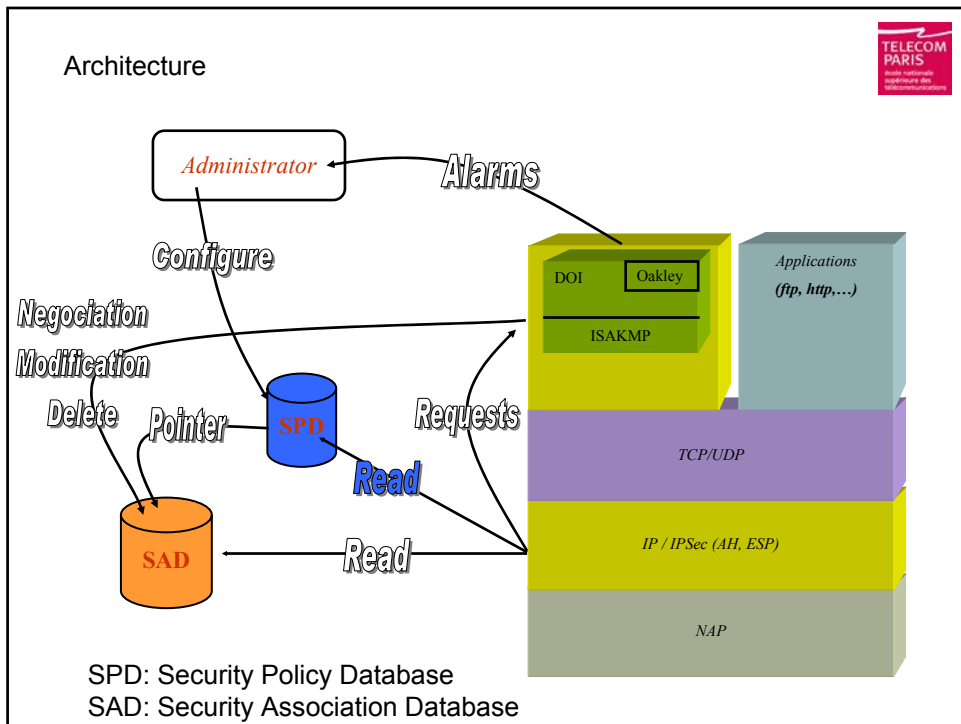
- IPSec = IP security Protocol
 - Standard développé à l'IETF
 - Premier RFC en 1995 sans gestion de clés
 - Deuxième version en Novembre 1998 avec la gestion des clés (IKE)
 - Partie commune entre IPv4 et IPv6 (obligatoire en IPv6)
- Implémentation de IPSec
 - Implémentation Native (dans la pile IP avec IPSec en native)
 - BITS (Bump in the Stack) : logiciel additionnel
 - BITW (Bump in the Wire) : processeur cryptographique externe

Apports d'IPSec

- IPSec
 - Couche réseau pour le **chiffrement** et l'**authentification**
 - Standards ouvert pour offrir des **communications privés et sécurisés**
 - Solution flexible pour **déployer des politiques de sécurité** à grande échelle
- Statut de IPSec
 - Plusieurs RFCs bien définis
 - Plusieurs implémentations (Nortel, Redcreek, Sun Solaris, Microsoft, DEC, Cisco, HP, Telebit, 6Wind, Freeswan, etc.)
 - Plusieurs tests de conformance et d'interopérabilité basés sur des implantations de référence
- Caractéristiques de IPSec
 - Standard pour la confidentialité, l'intégrité, et l'authentification pour les échanges sur le réseau Internet
 - Transparent aux infrastructures du réseau
 - Solution de sécurité de bout en bout incluant routeurs, firewalls, PCs et serveurs

Services de sécurité fournis par IPsec

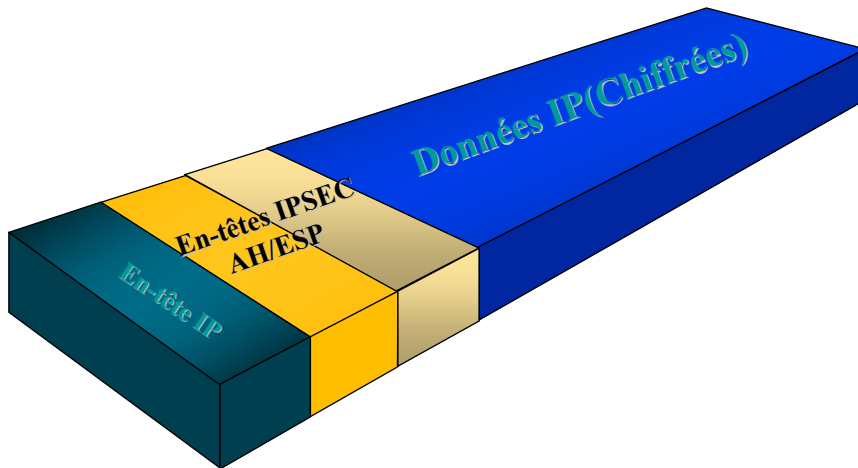
- Confidentialité
- Intégrité
- Authentification de l'origine des données
- Contrôle d'accès & contre analyse de trafic
- Non rejeu



- Protocoles
- AH: Authentication Header
 - ESP: EncapSuled Payload
 - IKE: Internet Key Exchange
 - ISAKMP
 - OAKLEY

Protocole: encapsulation

Authentification, intégrité et chiffrement



IPSec : deux modes

Chaque paquet IP est chiffré et/ou authentifié.

Il existe deux modes :

- transport : en-tête non modifié
- tunnel : encapsulation dans un nouveau paquet IP

Mode transport :

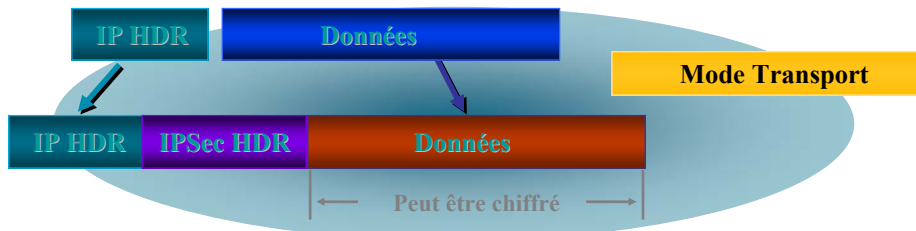
entre 2 correspondants finaux

Mode tunnel :

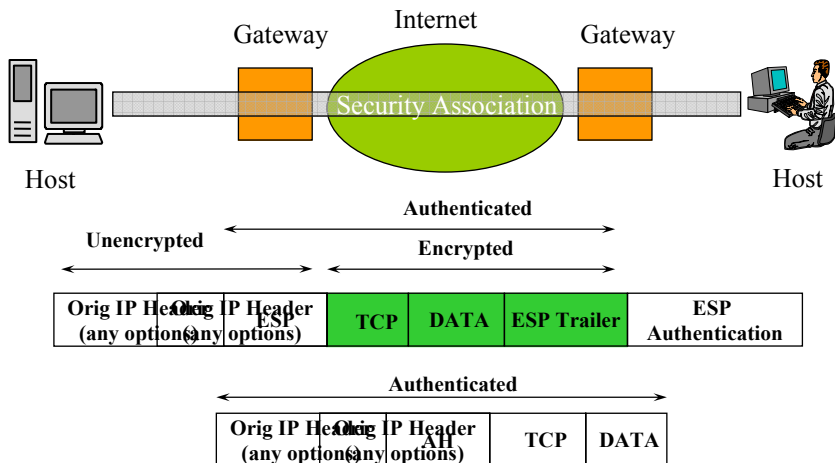
entre 2 passerelles ou entre 2 correspondants finaux
(utilisé pour les VPNs au niveau réseau)

Protocole: mode Transport

- Pour la confidentialité: seules les données sont chiffrées
- Implémenté au dessus de IP

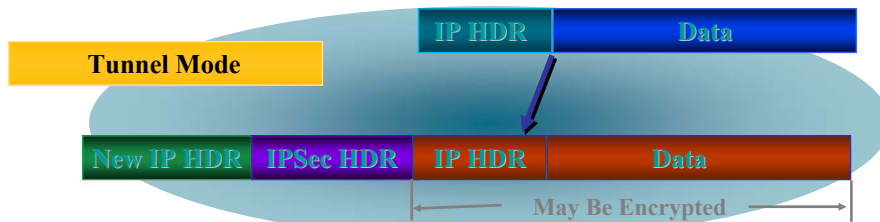


Architecture: IPsec mode transport

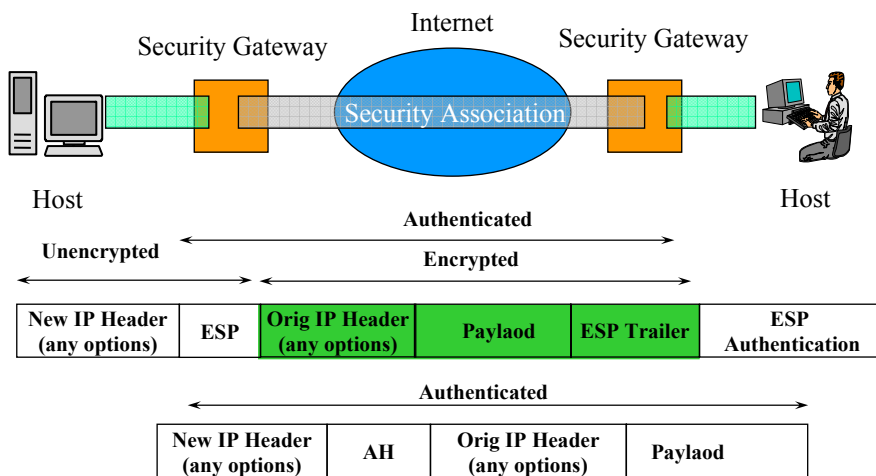


IPSec Mode Tunnel

- Idéal pour les VPNs



Architecture: IPSec Tunnel Mode



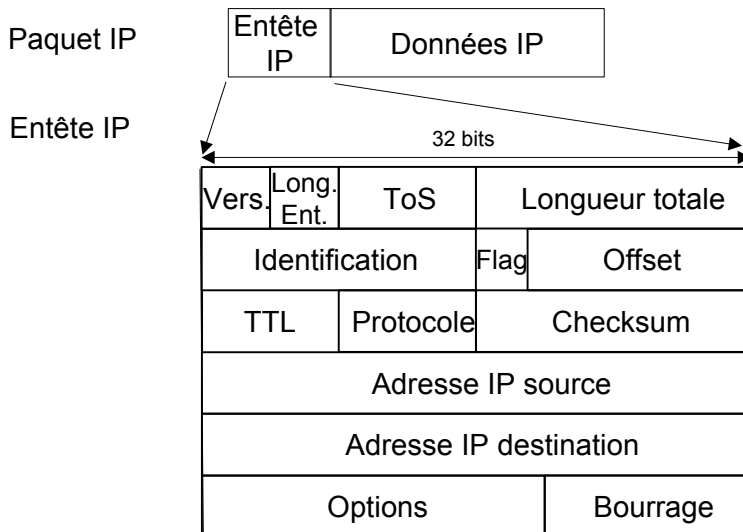
Authentication Header (AH)

- RFC 2402 (novembre 1998)
- Services
 - Intégrité
 - Authentification
 - Protection contre le rejeu
 - Pas de confidentialité

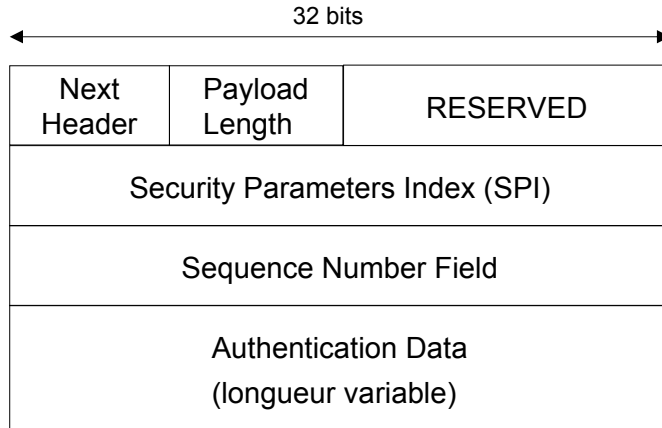
Principe :

ajout d'un bloc supplémentaire dans le paquet IP.

Authentication Header

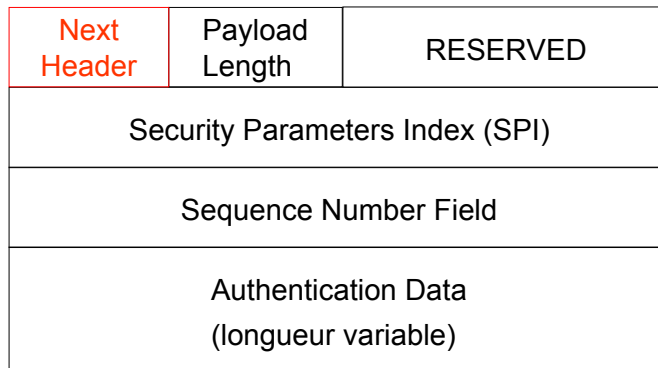


Authentication Header



34

Authentication Header



Next Header : protocole de niveau supérieur (TCP : 6, UDP : 17, ...).

Le champ "Protocol" de l'en-tête IPv4 précédant vaut 51.

35

Authentication Header

Next Header	Payload Length	RESERVED
Security Parameters Index (SPI)		
Sequence Number Field		
Authentication Data (longueur variable)		

Payload Length : longueur du bloc AH = nombre de mots de 32 bits – 2.

Authentication Header

Next Header	Payload Length	RESERVED
Security Parameters Index (SPI)		
Sequence Number Field		
Authentication Data (longueur variable)		

RESERVED : emplacement réservé pour le futur.

Tous les bits doivent être mis à 0.

Authentication Header

Next Header	Payload Length	RESERVED
Security Parameters Index (SPI)		
Sequence Number Field		
Authentication Data (longueur variable)		

SPI : identifiant unique de l'association de sécurité.

Authentication Header

Next Header	Payload Length	RESERVED
Security Parameters Index (SPI)		
Sequence Number Field		
Authentication Data (longueur variable)		

Sequence Number Field : protection contre le jeu (index initialisé à 0).

Elle est obligatoirement activée par l'émetteur.

Le récepteur la prend en compte de manière optionnelle.

Authentication Header

Next Header	Payload Length	RESERVED
Security Parameters Index (SPI)		
Sequence Number Field		
Authentication Data (longueur variable)		

Authentication Data : (résultat du hachage signé).

La signature est faite sur tout le paquet IP

=> authentification de la source et de l'intégrité des données

40

Authentication Header

Il y a deux modes possibles pour appliquer le mécanisme AH :

- mode tunnel (ajout d'un nouvel en-tête IP)
- mode transport (conservation de l'en-tête IP d'origine)

Pour sécuriser le trafic entre deux passerelles, il est nécessaire d'utiliser le mode tunnel.

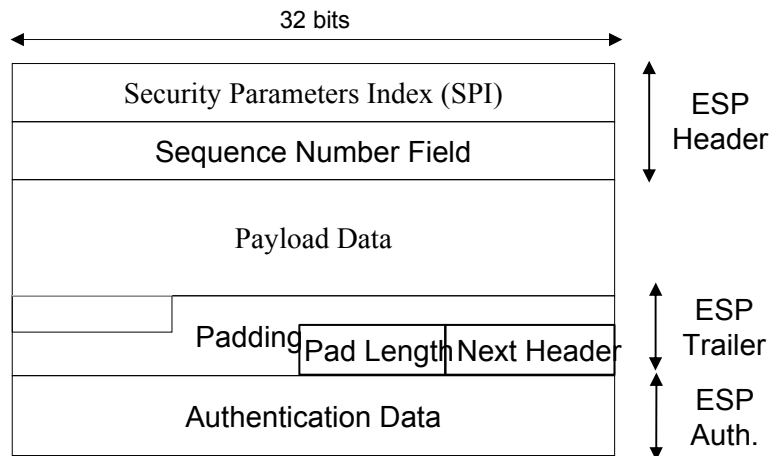
41

Protocol ESP: (Encapsulating Security Payload)

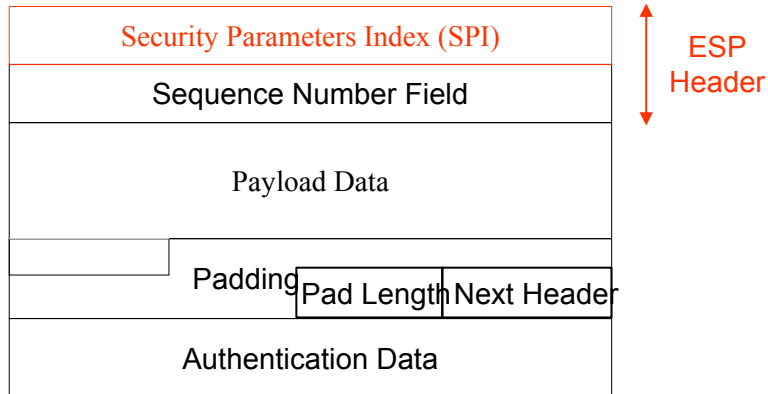
- RFC 2406
- Confidentialité + Intégrité
des données encapsulées par le paquet IP
- Authentification de la source
- Protection contre le jeu

Principe : chiffrement ET encapsulation

Encapsulating Security Payload

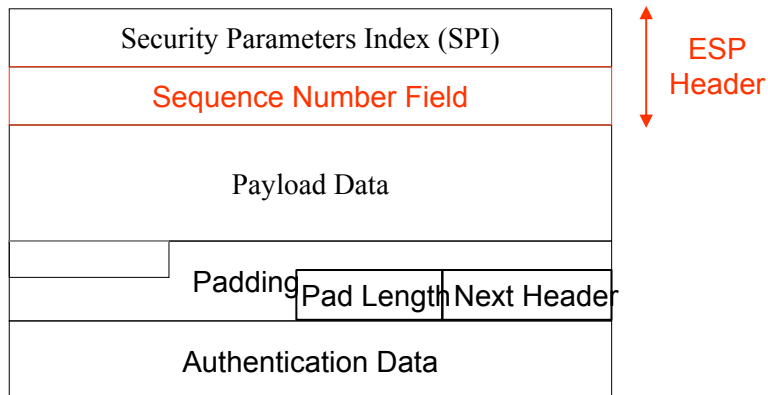


Encapsulating Security Payload



SPI : identifiant unique de l'association de sécurité (id. que pour AH).

Encapsulating Security Payload

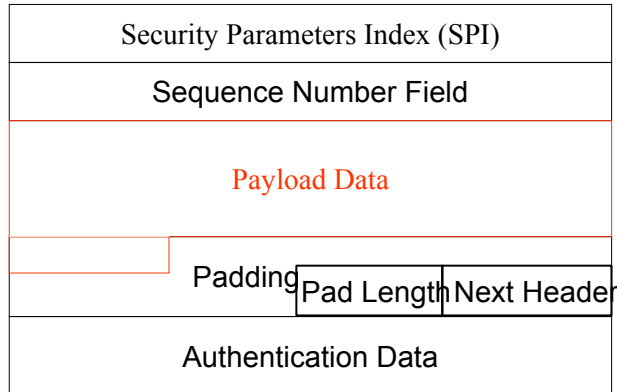


Sequence Number Field : protection contre le rejeu (index initialisé à 0).

Comme pour AH, l'émetteur l'active obligatoirement.

L'émetteur choisit de la prendre en compte ou non.

Encapsulating Security Payload

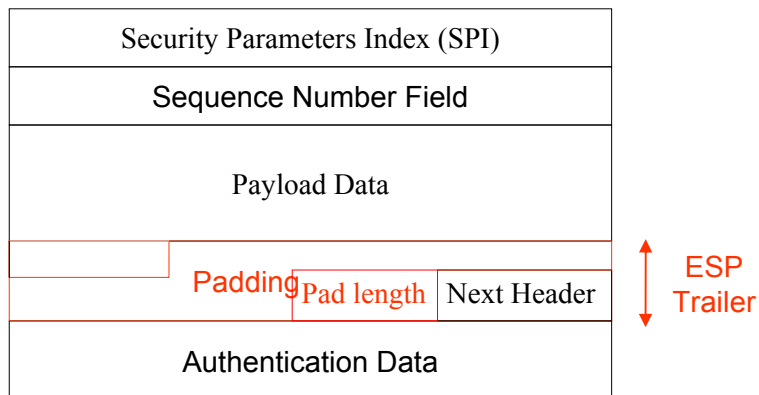


Payload Data : contient les données chiffrées.

Si l'algorithme de chiffrement nécessite des variables d'initialisation, elles sont stockées à cet endroit.

47

Encapsulating Security Payload

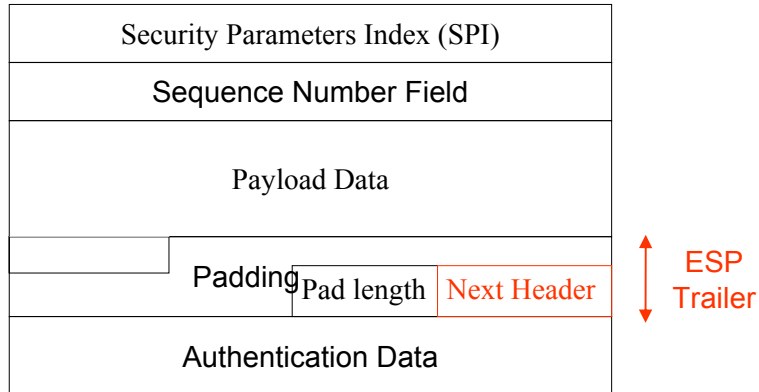


Padding : bourrage (de 0 à 255 bits)

Pad length : longueur du bourrage (1 octet)

48

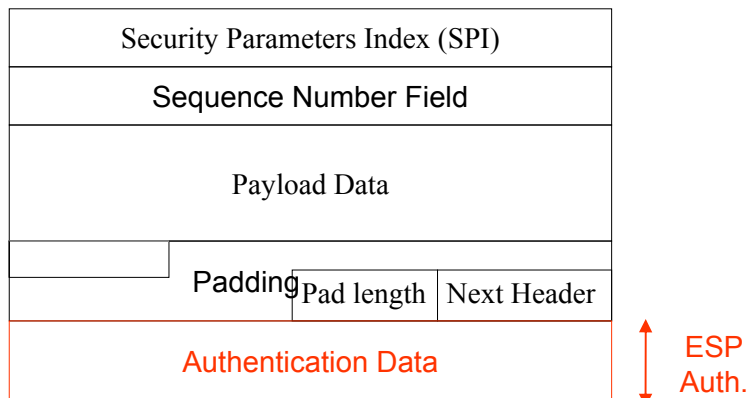
Encapsulating Security Payload



Next Header : protocole de niveau supérieur (TCP, UDP, ...)

49

Encapsulating Security Payload



Authentication Data : données d'authentification du paquet ESP (n'authentifie pas l'en-tête IP).

50

Encapsulating Security Payload

- Deux modes possibles pour appliquer ESP :

→ mode transport
(conservation de l'en-tête IP d'origine)

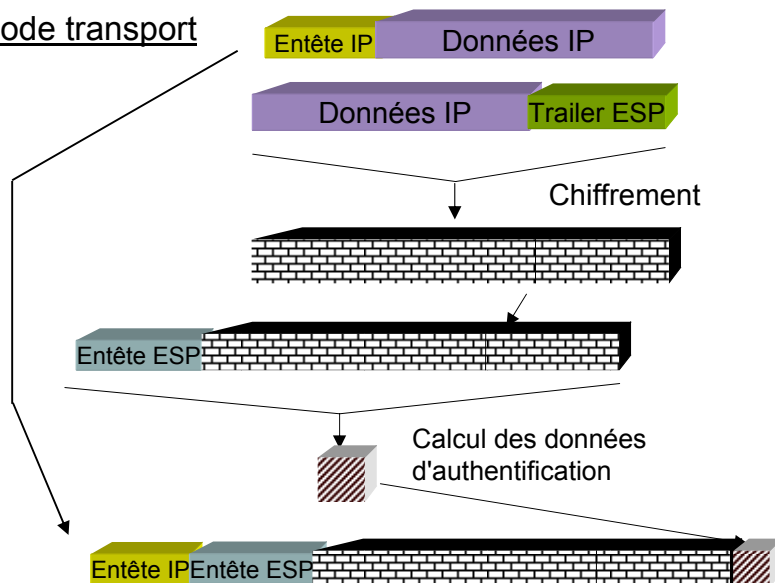
→ mode tunnel
(ajout d'un nouvel en-tête IP)

Pour sécuriser le trafic entre deux passerelles, il est nécessaire d'utiliser le mode tunnel

51

Encapsulating Security Payload

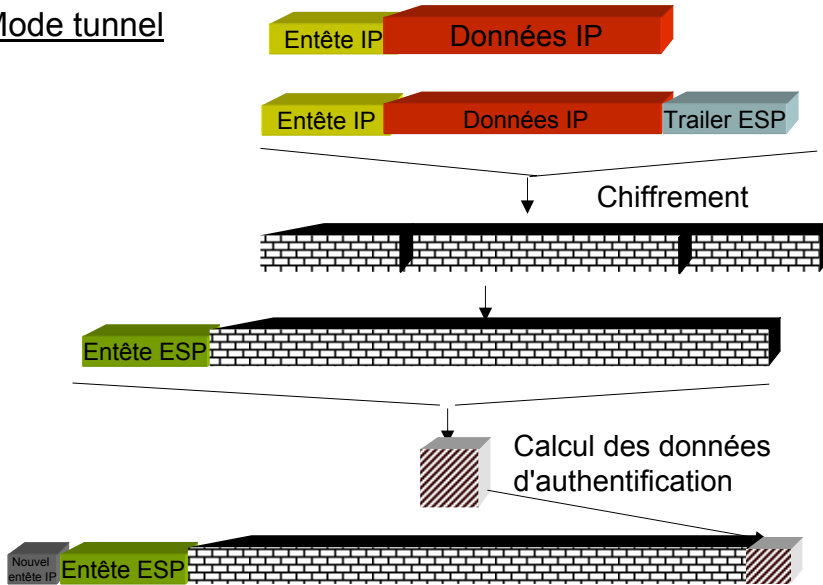
Mode transport



52

Encapsulating Security Payload

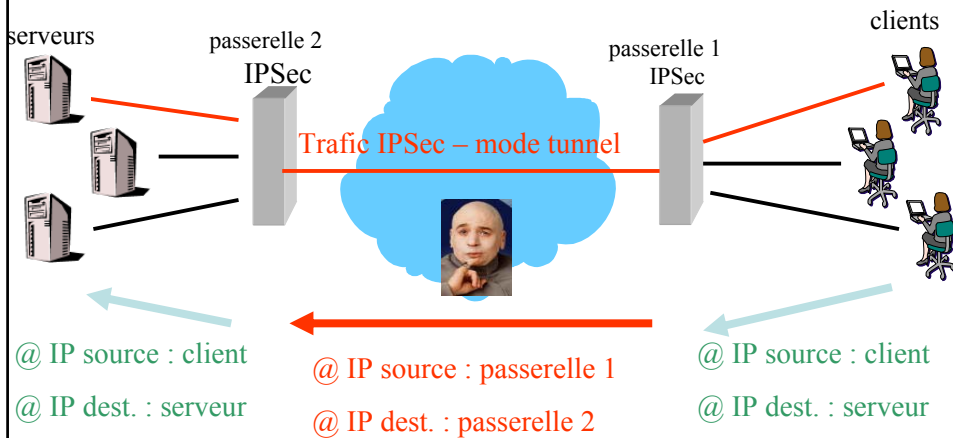
Mode tunnel



53

Encapsulating Security Payload

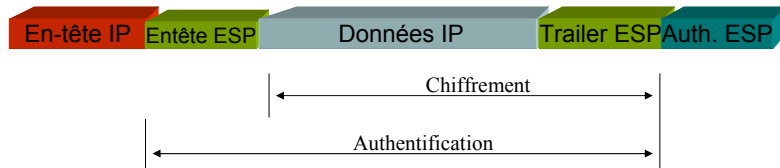
Mode tunnel : protection partielle contre l'analyse de trafic.



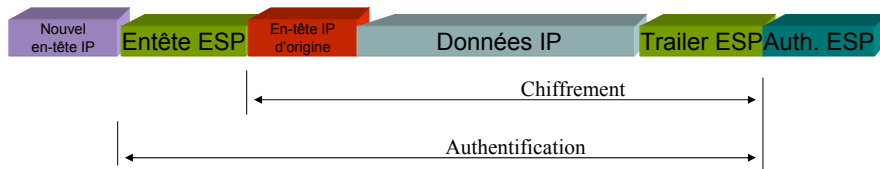
54

Encapsulating Security Payload

Mode transport



Mode tunnel



55

Encapsulating Security Payload

Algorithme obligatoire pour le chiffrement :

- 3DES

Algorithmes obligatoires pour l'authentification :

- HMAC-MD5
- HMAC-SHA-1

56

Les Associations de Sécurité (SA)

Connexion **unidirectionnelle** qui offre des services de sécurité au trafic qui transite par elle.

Ces services sont apportés soit par AH, soit par ESP.

Les SA précisent les "options" de ces mécanismes :
algorithme, activation de la protection anti-rejeu, ...

Chaque SA est identifiée par :

- l'adresse IP destination ;
- l'identifiant du protocole de sécurité (AH ou ESP) ;
- l'index du paramètre de sécurité (SPI).

Les Associations de Sécurité

Il existe 2 modes pour les SA :

- mode transport (entre deux hôtes)
- mode tunnel (entre deux passerelles ou entre deux hôtes)

Si une SA n'est pas suffisante pour protéger le trafic, on peut les associer en "paquets" (ou *bundle*) de SA.

Il existe 2 façons de combiner les SA :

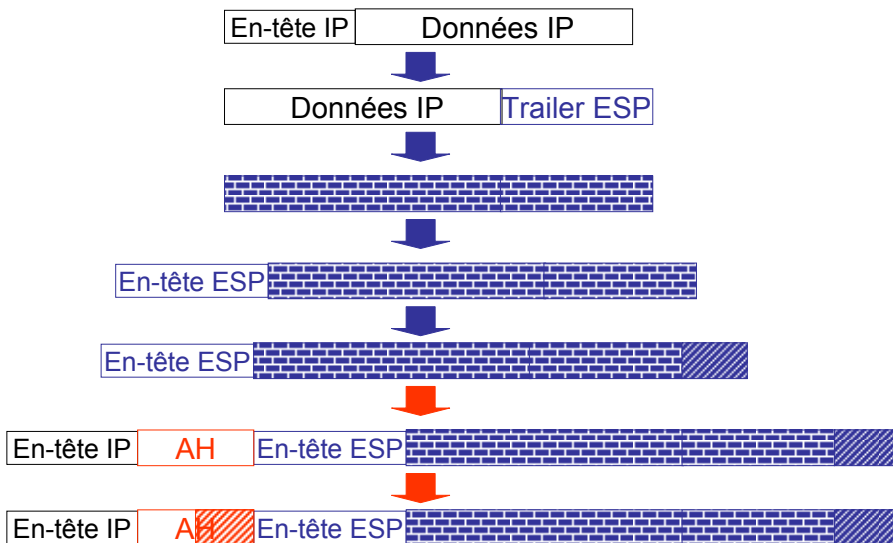
- juxtaposition du transport
- tunnel itératif

Juxtaposition du transport (mode transport)

2 AS peuvent s'appliquer au même datagramme IP.
Les mécanismes (AH et ESP) seront utilisés en mode transport.

On appliquera d'abord ESP, puis AH.

Les Associations de Sécurité

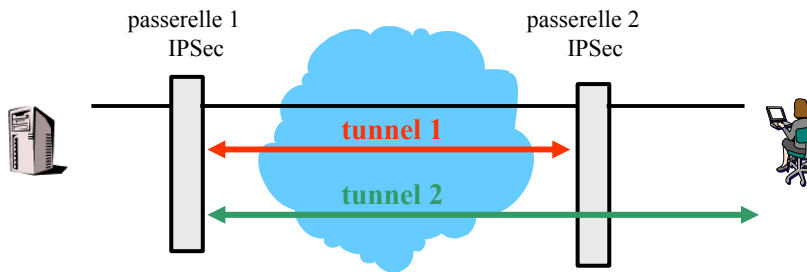


tunnels itératifs (mode tunnel)

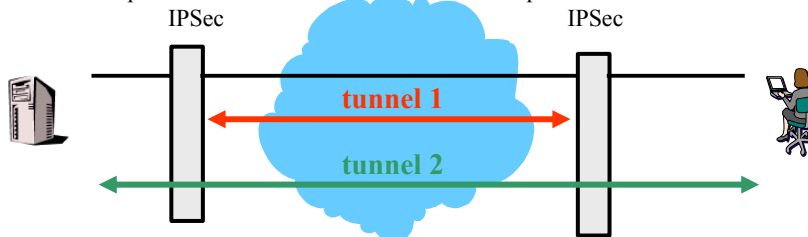
Surtout utilisé pour appliquer plusieurs AS au trafic entre deux passerelles.

Les extrémités de chaque tunnel ne sont pas forcément confondues.

extrémité commune



extrémités disjointes
passerelle 1
IPSec



Les Associations de Sécurité



Pour la mise en place et la gestion des SA, le protocole IPSec fait appel à deux bases de données :

- SPD (Security Policy Database) : indique les politiques qui déterminent le traitement de tout le trafic IP entrant ou sortant
- SAD (Security Association Database) : contient les paramètres qui sont associés à chaque SA active.

Il y a des entrées distinctes dans les bases SPD/SAD pour chaque interface IPSec.

63

Les Associations de Sécurité



SPD

- consultée pour le traitement de tout trafic.

Il y a soit des entrées distinctes pour le trafic entrant et sortant, soit deux bases différentes.

Trois choix de traitement pour un paquet IP :

- rejeter le paquet
- laisser passer le paquet sans protection IPSec
- laisser passer le paquet avec une protection IPSec
 - ⇒ préciser les services de sécurité, les protocoles et les algorithmes à utiliser, ...

On définit l'ensemble du trafic grâce à des sélecteurs...

64

Les Associations de Sécurité

Sélecteurs

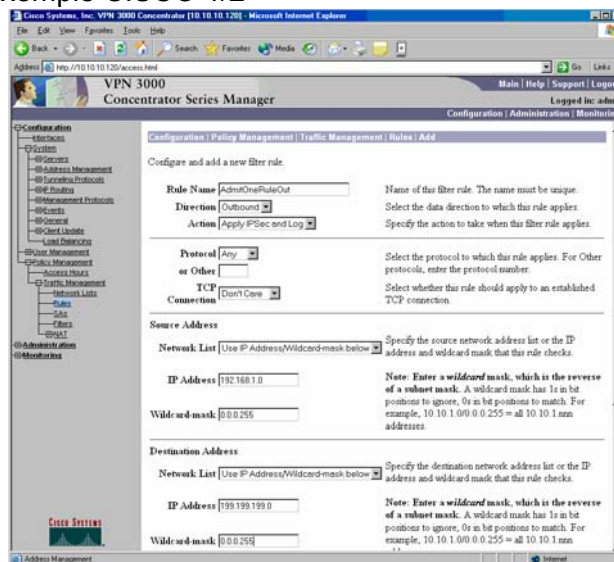
- @ IP destination
- @ IP source
- nom (d'utilisateur ou de système)
- protocole de la couche transport
- ports source et destination

Ils définissent la granularité des SA.

65

Les Associations de Sécurité

SPD : exemple CISCO 1/2



The screenshot shows the Cisco VPN 3000 Concentrator Series Manager web interface. The left sidebar contains a tree view with categories like Configuration, Administration, and Monitoring. The main content area is titled 'Configuration / Policy Management / Rules / Add'. It contains a form for adding a new filter rule. The form fields are as follows:

- Rule Name:** AdminOneRuleOut
- Direction:** Outbound
- Action:** Apply IPSec and Log
- Protocol:** Any
- or Other:** (empty)
- Connections:** Don't Care
- Source Address:**
 - Network List:** Use IP Address/Wildcard-mask below
 - IP Address:** 192.168.1.0
 - Wildcard-mask:** 0.0.0.255
- Destination Address:**
 - Network List:** Use IP Address/Wildcard-mask below
 - IP Address:** 199.199.199.0
 - Wildcard-mask:** 0.0.0.255

Notes on the right side of the form:

- For Rule Name: Name of this filter rule. The name must be unique.
- For Direction: Select the data direction to which this rule applies.
- For Action: Specify the action to take when this filter rule applies.
- For Protocol: Select the protocol to which this rule applies. For Other protocols, enter the protocol number.
- For Connections: Select whether this rule should apply to an established TCP connection.
- For Source Address Wildcard-mask: Note: Enter a wildcard mask, which is the reverse of a subnet mask. A wildcard mask has 1s in bit positions to ignore, 0s in bit positions to match. For example, 10.10.1.0/0.0.0.255 = all 10.10.1.xxx addresses.
- For Destination Address Wildcard-mask: Note: Enter a wildcard mask, which is the reverse of a subnet mask. A wildcard mask has 1s in bit positions to ignore, 0s in bit positions to match. For example, 10.10.1.0/0.0.0.255 = all 10.10.1.xxx addresses.

66

SPD : exemple CISCO 2/2

VPN 3000 Concentrator Series Manager

Modify a configured Security Association.

SA Name: AdminOneSA Specify the name of this Security Association (SA).

Inheritance: Admin One Select the granularity of this SA.

IPsec Parameters

Authentication: ESP/MAC/HMAC-128 Select the packet authentication algorithm to use.

Encryption Algorithm: DES-168 Select the ESP encryption algorithm to use.

Encapsulation Mode: Tunnel Select the Encapsulation Mode for this SA.

Perfect Forward Secrecy: Group 2 (1024-bits) Select the use of Perfect Forward Secrecy.

Lifetime Measurement: Time Select the Lifetime measurement of the IPsec keys.

Data Lifetime: 10000 Specify the data lifetime in kilobytes (KB).

Time Lifetime: 28800 Specify the time lifetime in seconds.

IKE Parameters

IKE Peer: 0.0.0.0 Specify the IKE Peer for a LAN-to-LAN IPsec connection.

Negotiation Mode: Aggressive Select the IKE Negotiation mode to use.

Digital Certificate: None (Use Freshened Keys) Select the Digital Certificate to use.

Certificate: Entire certificate chain Choose how to send the digital certificate to the IKE peer.

Transmission: Identity certificate only

IKE Proposal: AdminOneIKE Select the IKE Proposal to use as IKE initiator.

Apply Cancel

67

SAD

Elle contient les paramètres de chaque SA active :

- compteur de numéro de séquence (SNC, Sequence Number Counter)
- algorithme d'authentification d'AH
- algorithme de chiffrement d'ESP
- durée de vie de l'AS

68

Les Associations de Sécurité

SAD : exemple Microsoft

The screenshot shows the 'IP Security Monitor' window. At the top, it displays 'Security Associations' with a table containing one entry:

Policy Name	Security	Filter Name	Source Address	Dest. Address	Protocol	Src. Port	Dest.
{A8F942DF...	ESP D...	<No Name...	DKALIN-03.spa...	ipsecmage.jp...	TCP	0	445

Below the table are 'IPSEC Statistics' and 'ISAKMP/Oakley Statistics'.

IPSEC Statistics

Active Associations	1
Confidential Bytes Sent	151,633,118
Confidential Bytes Received	147,303,714
Authenticated Bytes Sent	155,388,742
Authenticated Bytes Received	151,071,845
Bad SPI Packets	0
Packets Not Decrypted	0
Packets Not Authenticated	0
Key Additions	176

ISAKMP/Oakley Statistics

Oakley Main Modes	145
Oakley Quick Modes	176
Soft Associations	0
Authentication Failures	0

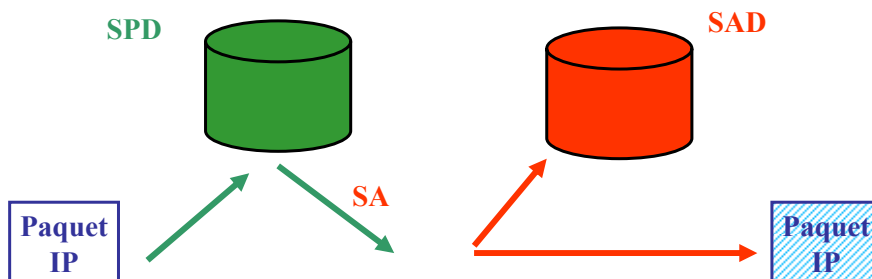
At the bottom, a status bar indicates 'IP Security is enabled on this computer.'

69

Les Associations de Sécurité

Résumé

Chaque paquet IP est traité par la SPD qui lui fait correspondre une ou plusieurs SA, repertoriées dans la SAD.




70

Les Associations de Sécurité

Gestion des SA

Pour créer des SA, il est nécessaire que les deux parties soient d'accord sur

- les paramètres à utiliser,
- en particulier pour **échanger les clés de session**.



configuration
manuelle



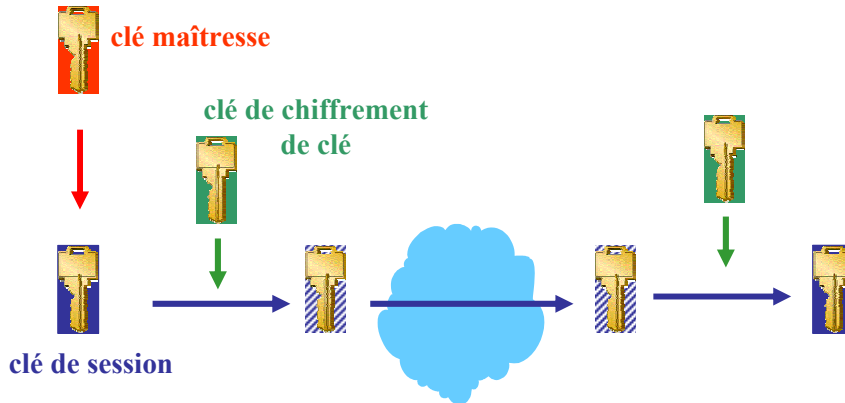
négociation
IKE

La gestion des clés

Les 3 types de clés :

- **clés de chiffrement de clés** : elles servent à chiffrer d'autres clés. Elles ont une durée de vie longue.
- **clés maîtresses** : elles servent à générer d'autres clés.
- **clés de session** (ou clés de chiffrement) : elles servent à chiffrer les messages. Elles ont en général une durée de vie courte.

La gestion des clés



73

La gestion des clés

Techniques pour l'échange de clé:

- **transport de clés** : on échange une clé chiffrée (cf. transparent précédent)
- **génération de clés** : on partage un secret sans entente préalable.
 - Alors, l'algorithme le plus utilisé est Diffie-Hellman.

74

La gestion des clés

Propriétés des protocoles d'échanges de clé :

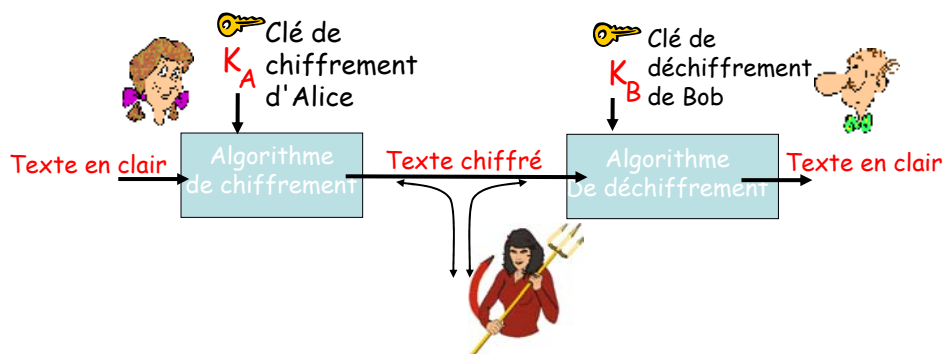
- **Perfect Forward Secrecy** : les clés de sessions passées ne peuvent pas être retrouvées si un secret à long terme est découvert.
- **Back Traffic Protection** : la génération des clés est telle que chaque clé est indépendante des clés passées.

La gestion des clés

Diffie-Hellman (1976)

- basé sur la cryptographie à clé publique
- permet de partager un secret sans entente préalable

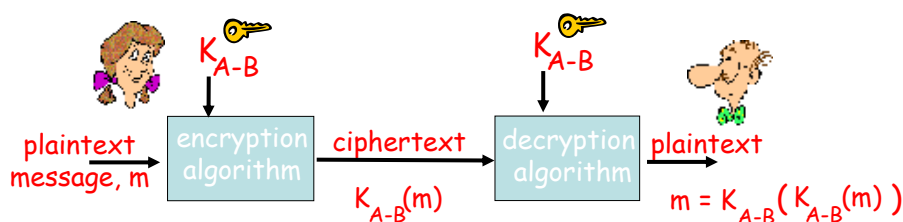
Vocabulaire de cryptographie



Cryptographie à clé symétrique : clés identiques pour l'émetteur et le récepteur

Cryptographie à clé publique : clé de chiffrement publique, clé de déchiffrement secrète (privée)

Cryptographie à clé symétrique



Cryptographie à clé symétrique : Bob et Alice partagent une même clé (connue) : K_B

- Ex : la clé consiste à connaître le mode de substitution dans un chiffrement par substitution monoalphabétique
- Q: Comment Bob et Alice se mettent-ils d'accord sur la valeur de la clé ?

Cryptographie à clé publique

Crypto à clé symétrique

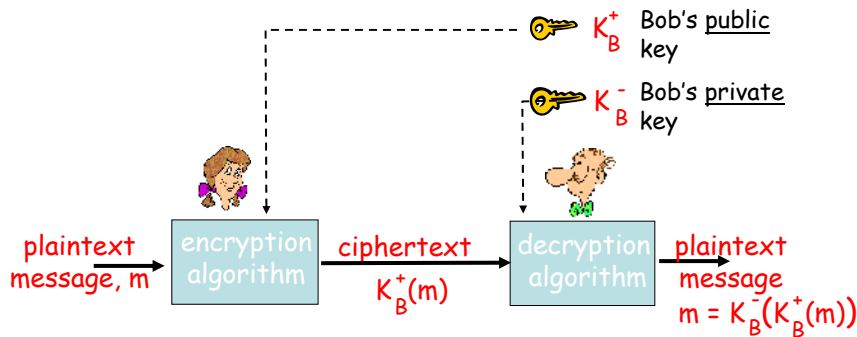
- Nécessite le partage d'une clé entre l'émetteur et le récepteur
- Q: comment se mettre d'accord sur la clé au départ (surtout s'ils ne se sont jamais rencontrés) ?

Crypto à clé publique

- Approche radicalement différente [Diffie-Hellman76, RSA78]
- L'émetteur et le récepteur ne partagent *pas* de clé secrète
- clé de chiffrement *publique* connue de *tous*
- la clé de déchiffrement *privée* n'est connue que du récepteur



Cryptographie à clé publique



Algorithmes de chiffrement par clé publique

Besoins :

- ① Besoin de K_B^+ et de K_B^- telles que

$$K_B^-(K_B^+(m)) = m$$

- ② À partir de la clé publique K_B^+ ,
il devrait être impossible de
calculer la clé privée K_B^-

RSA : algorithme de Rivest, Shamir, Adelson

Intermédiaires de confiance

Problème des clés symétriques

- Comment 2 entités établissent-elles une clé secrète partagée à travers un réseau ?

Solution :

- Centre de distribution de clé de confiance (KDC) agissant comme un intermédiaire entre les entités

Problème des clés publiques :

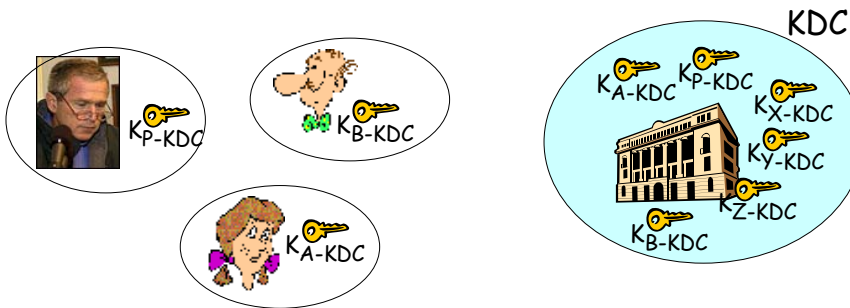
- Quand Alice obtient la clé publique de Bob (à partir d'un site Web, d'un e-mail, d'une disquette), comment sait-elle que c'est la clé publique de Bob, et pas celle de Trudy ?

Solution :

- Autorité de certification de confiance (CA)

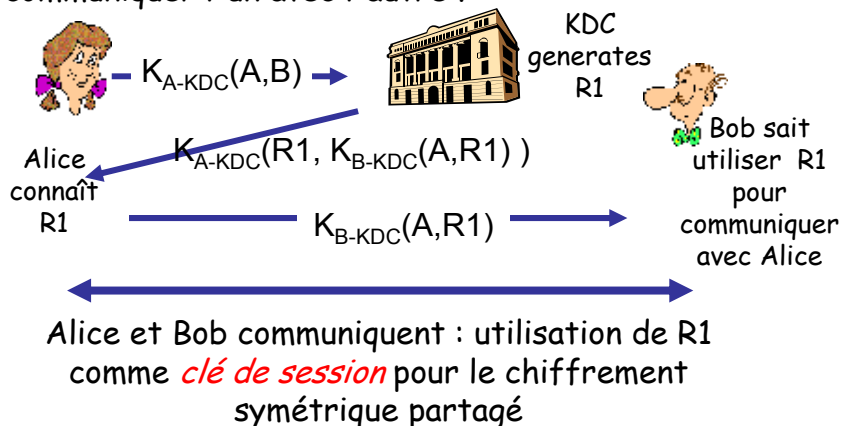
Key Distribution Center (KDC)

- Alice et Bob doivent partager une clé symétrique.
- KDC** : le serveur partage une clé secrète différente avec *chaque* utilisateur enregistré (nombreux utilisateurs)
- Alice et Bob possèdent leur propre clé symétrique K_{A-KDC} K_{B-KDC} , pour communiquer avec le KDC.



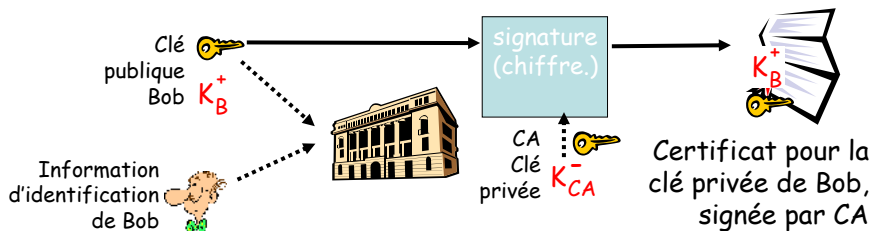
Key Distribution Center (KDC)

Q: Comment le KDC permet-il à Bob et Alice de déterminer une clé secrète symétrique partagée pour communiquer l'un avec l'autre ?



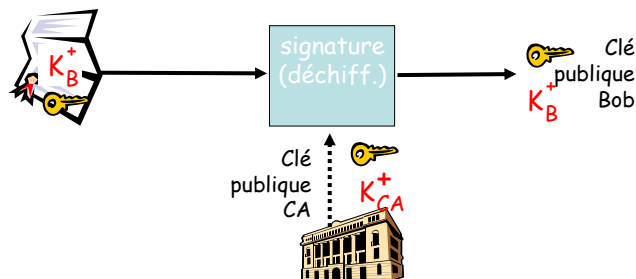
Autorités de certification

- **Certification authority (CA)** : relie une clé publique à une entité particulière E.
- E (personne, routeur) enregistre sa clé publique auprès du CA.
 - E fournit une “preuve d'identité” au CA
 - Le CA crée un lien certifié entre e et sa clé publique
 - Le certificat contenant la clé publique de E est signé numériquement par le CA – le CA dit “ceci est la clé publique de E”



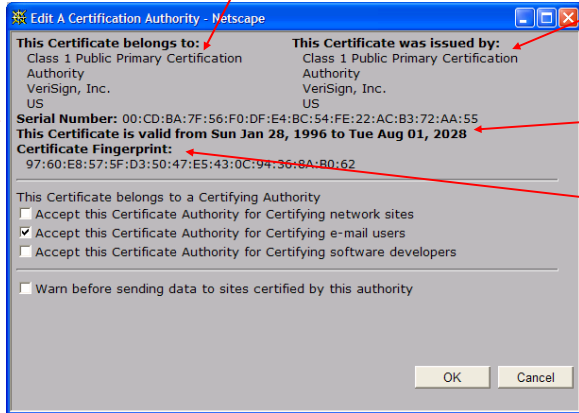
Autorités de certification

- Quand Alice veut la clé publique de Bob :
 - Elle obtient le certificat de Bob (par Bob ou autre).
 - Elle applique la clé publique du CA au certificat de Bob et obtient la clé publique de Bob.



Un certificat contient :

- Un numéro de série (unique pour chaque émetteur)
- info sur le propriétaire du certificat, y compris l'algorithme et la valeur de la clé elle-même



- info sur l'émetteur du certificat
- Dates de validité
- Signature numérique de l'émetteur

La gestion des clés

Solution :

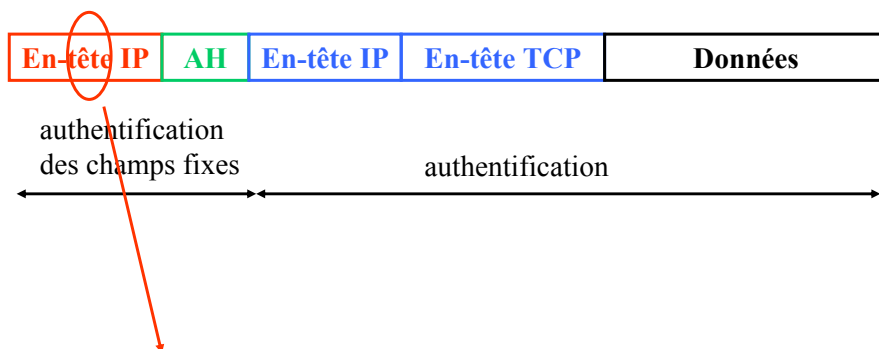
- échanger des valeurs publiques authentifiées
- ou
- authentifier les valeurs publiques après l'échange

Les clés publiques doivent pouvoir être reliées de manière sûre à un individu (ou à un serveur, une institution, ...).

IPSec et NAT

La traduction d'adresse

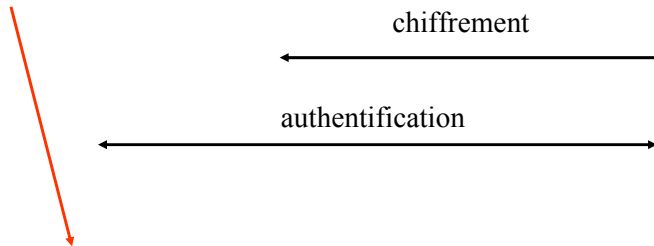
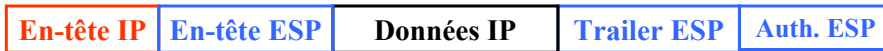
AH mode tunnel :



La NAT modifie certains champs considérés comme fixes (adresses IP) : elle ne peut pas s'appliquer.

La traduction d'adresse

ESP mode transport :



L'en-tête IP n'est pas authentifiée : le mécanisme de traduction d'adresse STATIQUE est donc possible.

Il peut y avoir des problèmes avec les checksums TCP.

96

La traduction d'adresse

Pour calculer le checksum, la pile TCP/IP utilise un « pseudo en-tête » (« pseudo header »), qui inclut l'adresse IP source et l'adresse IP destination.

En mode transport, l'en-tête IP est conservé, et l'adresse IP sera modifiée par la NAT.

97

La traduction d'adresse

ESP mode tunnel :



chiffrement

authentification

L'en-tête IP n'est pas authentifiée : le mécanisme de traduction d'adresse STATIQUE est donc possible.

98

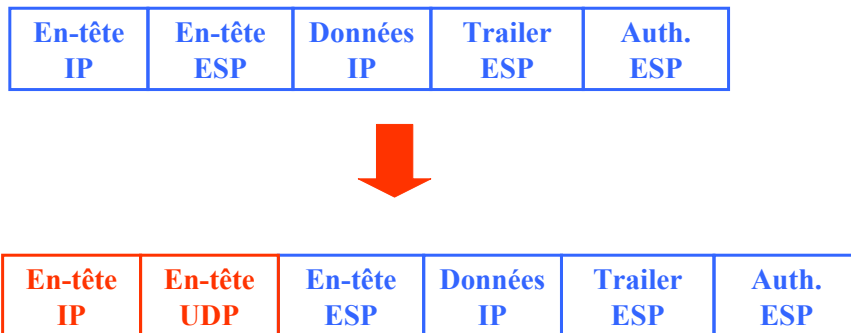
La traduction d'adresse

		Mécanisme	
		AH	ESP
Mode	Transport	NON	STATIQUE
	Tunnel	NON	STATIQUE

99

La traduction d'adresse

Une solution pour utiliser la NAT (ou la NAPT) est d'encapsuler IPSec sur de l'UDP :



100

La traduction d'adresse

L'encapsulation d'IPSec dans le protocole TCP est également possible, mais présente les inconvénients suivants :

- les en-têtes sont plus importants (20 octets contre 8)
- trafic vulnérable aux attaques de type *TCP Reset*

101

La fragmentation de paquets

La fragmentation des paquets

IPSec et le MTU (Maximum Transfer Unit)

Paquet original :

En-tête IP	Données IP
---------------	---------------

20 octets 1480 octets

1500 octets

Paquet après l'application du mécanisme ESP en mode tunnel :

Nouvel en-tête IP	En-tête ESP	En-tête IP	Données IP	Trailer ESP	Auth. ESP
----------------------	----------------	---------------	---------------	----------------	--------------

20 octets

16 octets

20 octets

1480 octets

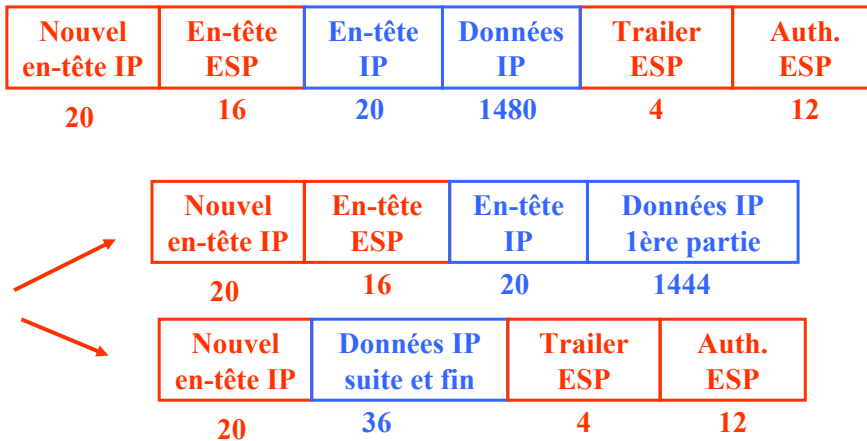
4 octets

12 octets

➔ 1552 octets

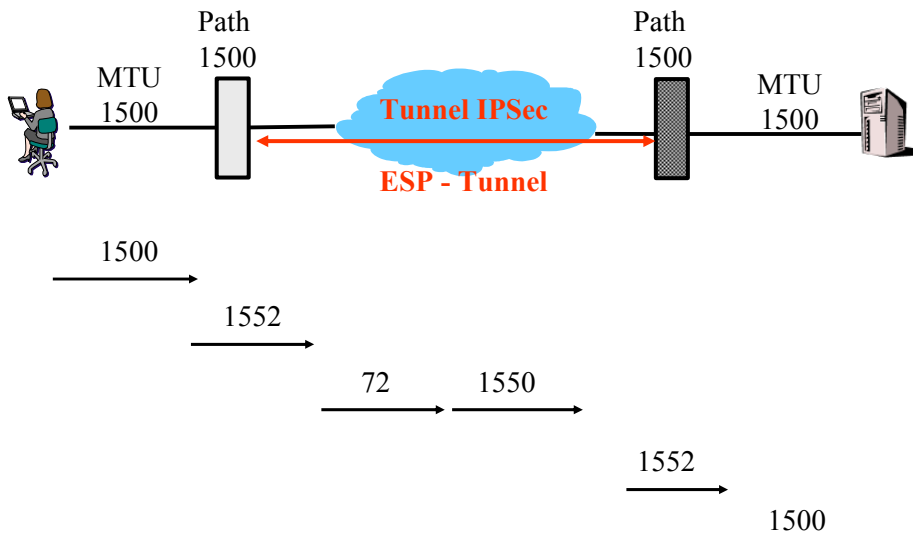
La fragmentation des paquets

Le paquet de 1552 octets est alors découpé en 2 paquets : 1 paquet de 1500 octets, et un autre de 72 octets :



105

La fragmentation des paquets



106

La fragmentation des paquets



La fragmentation des paquets IPSec est possible, mais elle a un coût conséquent en terme de performances.

Il est préférable de réduire le MTU.

Si le *flag DF* est à 1, on ne fragmente pas le paquet, mais on recherche le MTU du chemin -> PMTU discovery

107

La fragmentation des paquets



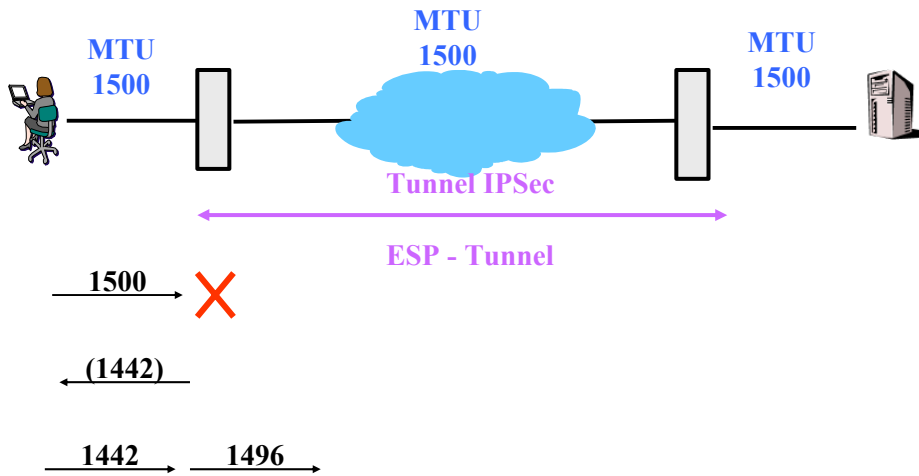
PMTUD : Path MTU Discovery

RFC 1191(Novembre 1990)

Envoi d'un paquet ICMP (type 3 – code 4) qui contient la valeur du MTU pour la portion suivante (*Next Hop MTU*).

108

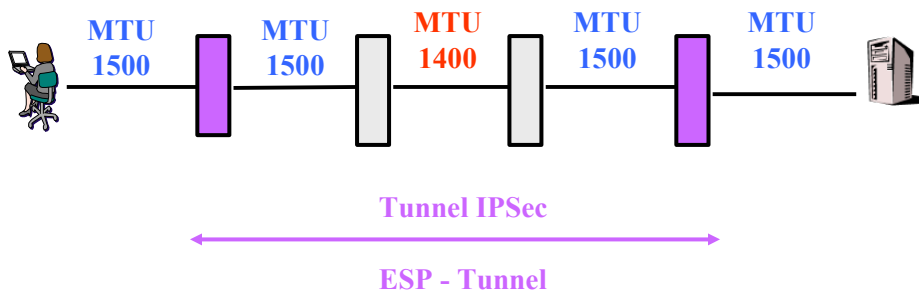
La fragmentation des paquets



109

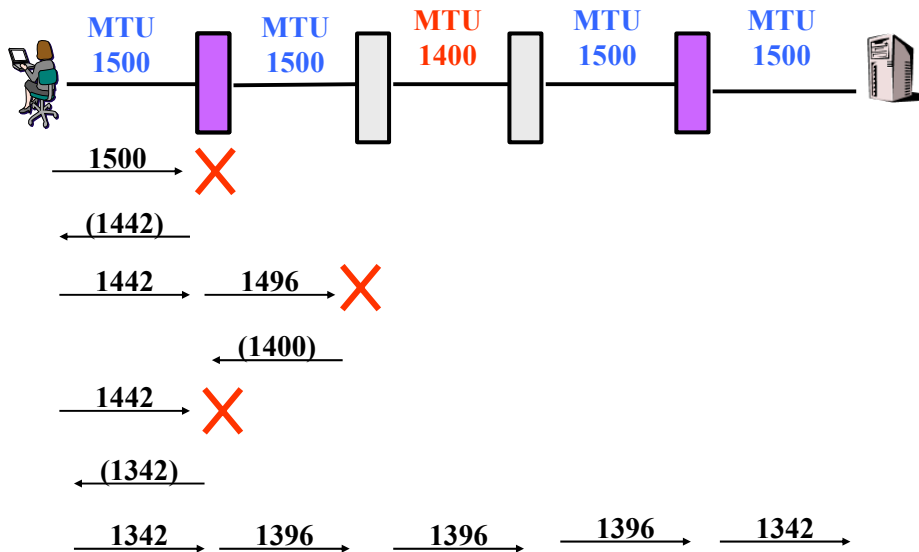
La fragmentation des paquets

PMTUD : Path MTU Discovery



110

La fragmentation des paquets



111

La fragmentation des paquets

Dans un trafic IPSec "normal" :

- il n'y a pas de fragmentation
- le MTU est découvert dynamiquement

112

La fragmentation des paquets



Problème : les messages ICMP peuvent être filtrés par les pare-feux.

Une solution consiste à choisir des MTU suffisamment bas à chaque extrémité du chemin.

113

Protocoles d'échange de clé



Protocoles d'échange de clés



Oakley

RFC 2412

Plusieurs options pour distribuer les clés:

- Diffie-Hellman classique
- Chiffrer une clé puis la distribuer
- Dériver une nouvelle clé d'une clé existante

115

Protocoles d'échange de clés



Trois étapes :

- l'échange de cookies
- l'échange de valeurs publiques
- l'authentification

Oakley permet la négociation d'un grand nombre de paramètres.

116

ISAKMP



ISAKMP (Internet Security Association and Key Management Protocol)

Pour

- l'établissement
 - la modification
 - la suppression
- des Associations de Sécurité

ISAKMP est un cadre générique qui doit être accompagné d'un *domaine d'interprétation* (DOI - Domain Of Interpretation).

117

ISAKMP



ISAKMP comprend deux phases :

- **l'établissement d'une SA ISAKMP**
 - authentification des tiers,
 - génération des clés,
 - échanges ISAKMP
- **la négociation des paramètres d'une SA** pour un mécanisme donné (AH ou ESP par exemple)
 - le trafic de cette phase est sécurisé par la SA ISAKMP

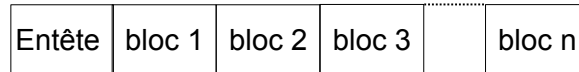
Rq : Une SA ISAKMP est bidirectionnelle

Une SA ISAKMP a une durée de vie plus longue qu'une SA IPSec

118

ISAKMP

Les messages ISAKMP



2 cookies :

- protection contre le déni de service
- identifiants

+ Next Payload

Nombre variable de blocs

119

ISAKMP

Il existe 13 types de blocs :

- | | |
|---------|-----------------------------|
| • SA | <i>Security Association</i> |
| • P | <i>Proposal</i> |
| • T | <i>Transform</i> |
| • KE | <i>Key Exchange</i> |
| • ID | <i>Identification</i> |
| • CERT | <i>Certificate</i> |
| • CR | <i>Certificate Request</i> |
| • HASH | <i>Hash</i> |
| • SIG | <i>Signature</i> |
| • NONCE | <i>Nonce</i> |
| • N | <i>Notification</i> |
| • D | <i>Delete</i> |
| • VID | <i>Vendor ID</i> |

120

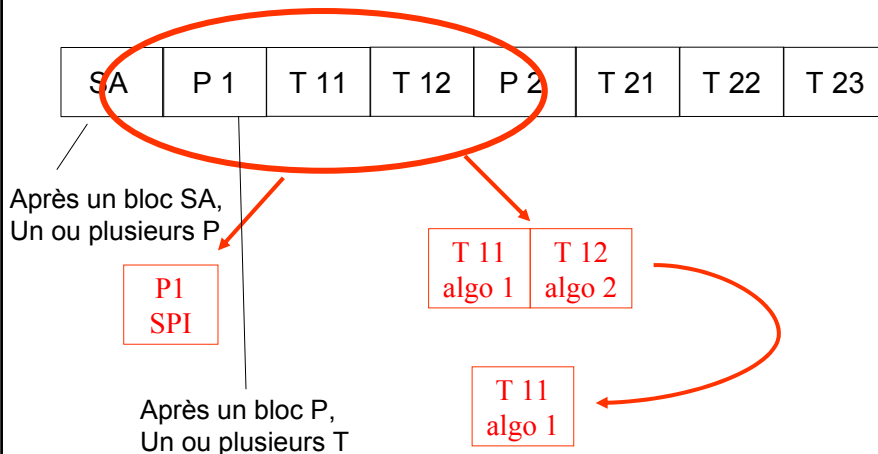
ISAKMP

- SA (*Security Association*) : indique le contexte de la négociation.
Paramètre DOI (Domain of Interpretation):
0 pour ISAKMP
1 pour IPSec
- P (*Proposal*) : mécanisme de sécurité de l'on désire utiliser (AH, ESP) et le SPI associé à la SA.
Chaque bloc est numéroté. S'il y a plusieurs mécanismes pour une même SA, les blocs portent le même numéro.
- T (*Transform*) : indique une transformation (algorithme de chiffrement, fonction de hachage, ...).
Ces blocs sont également numérotés.

121

ISAKMP: enchaînement des blocs SA, P, T

La proposition retenue fournira le SPI à l'association de sécurité concernée:



122

ISAKMP



- KE (*Key Exchange*) :
 - pour le transport des données nécessaires à la génération de la clé de session.
- ID (*Identification*) :
 - pour l'identification des parties.
 - Contient un champ *ID Type* (ex: une @IP pour ISAKMP)
- CERT (*Certificate*) :
 - transport des certificats, ou toute information s'y rattachant.
- CR (*Certificate Request*) :
 - pour réclamer un certificat à son interlocuteur.
- HASH (*Hash*) :
 - contient le résultat de l'application d'une fonction de hachage.

123

ISAKMP



- SIG (*Signature*) :
 - même rôle que le bloc *HASH*, mais il est utilisé dans le cas d'une signature.
- NONCE (*Nonce*) : transport de l'aléa.
- N (*Notification*) : pour transmettre les messages d'erreur ou d'informations sur les négociations en cours.
 - 2 champs possibles : *Notify Message Type* et *Notify Data*.
- D (*Delete*) : pour supprimer une SA et indiquer qu'elle n'est plus valable.
- VID (*Vendor ID*) : réservé aux programmeurs pour distinguer 2 instances d'implémentation.

124

ISAKMP

Les types de messages

A partir des blocs précédents, le protocole ISAKMP définit des types d'échanges (*Exchange Types*).

Il y a 5 types d'échanges par défaut :

- Base Exchange
- Identity Protection Exchange
- Authentication Only Exchange
- Aggressive Exchange
- Informational Exchange

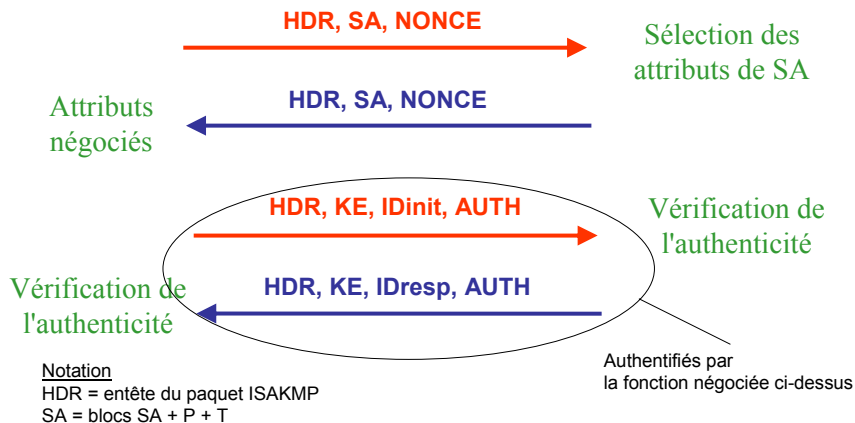
125

ISAKMP

Base Exchange

Initiator

Responder

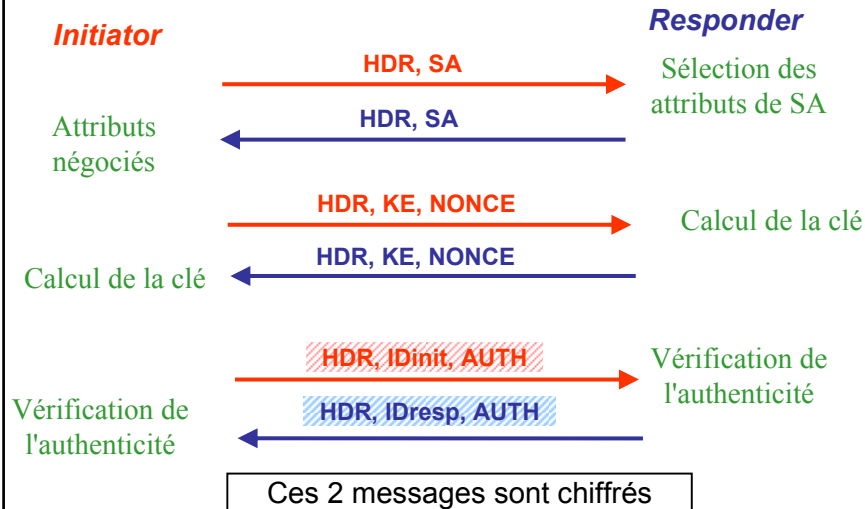


Pas de protection de l'anonymat: identités échangées avant partage d'un secret pour les chiffrer

126

ISAKMP

Identity Protection Exchange



127

Protection de l'identité avec 2 messages supplémentaires

ISAKMP

Authentication Only Exchange



Intérêt de cette approche:

La génération d'une clé est un processus gourmand en ressources système, et doit être évité si elle n'est pas nécessaire.

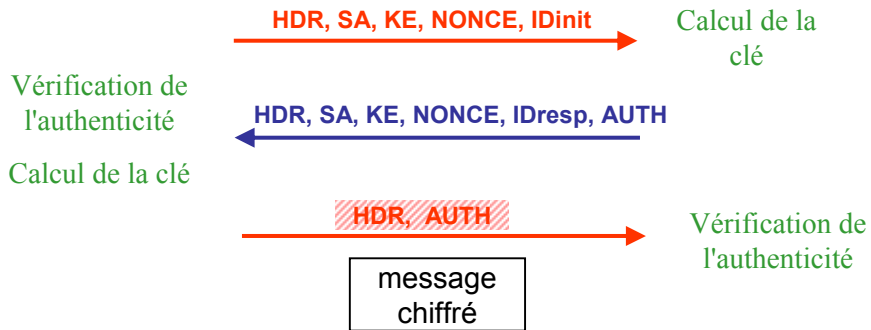
128

ISAKMP

Aggressive Exchange

Initiator

Responder



Un seul message avec négociation de la SA,
d'authentification et d'échange de clé => agressive
l'anonymat des tiers n'est pas protégé.
Il n'y a pas de choix possible dans la négociation de la SA.

129

ISAKMP

Informational Exchange

Initiator

Responder



130

IPSec DOI



Domaine d'interprétation pour IPSec

RFC 2407

Ce document définit les paramètres négociés et les conventions pour l'utilisation du protocole ISAKMP dans le cadre d'IPSec.

Exemple : Bloc P - définition du protocole de sécurité

Dans le cadre de l'IPSec DOI, ce bloc peut prendre 4 valeurs :

- ISAKMP
- AH
- ESP
- IPCOMP (compression des données au niveau IP)

131

IPSec DOI



Domaine d'interprétation pour IPSec

Exemple : Bloc T - définition de l'algorithme

Pour AH, il y a 3 choix possibles :

- MD5
- SHA
- DES

Pour ESP :

- | | |
|--------|------------|
| • DES | • BLOWFISH |
| • 3DES | • 3IDEA |
| • RC5 | • RC4 |
| • IDEA | • NULL |
| • CAST | |

132

IPSec DOI

Domaine d'interprétation pour IPSec

Exemple : Bloc ID - définition de l'identité du tiers

- DES
- sous-réseau IPv4
- plage d'adresses IPv4 (ou IPv6)
- FQDN
- user FQDN
- X.500 Distinguished Name
- X.500 General Name
- Key ID

133

IKE

IKE

RFC 2409

Utilise ISAKMP pour construire un protocole pratique.

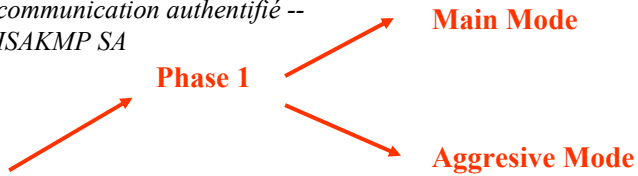
IKE comprend cinq modes :

- principal (*Main Mode*)
- agressif (*Aggressive Mode*)
- rapide (*Quick Mode*)
- nouveau groupe (*New Group Mode*)
- Informationnel (*informational mode*)

134

IKE

*Établissement d'un canal de
communication authentifié --
ISAKMP SA*



*Négociation des SA des
services utilisés et leur
paramètres*

135

IKE

Phase 1 : Main Mode

6 messages sont générés par le mode *Main Mode* durant la phase 1 en vue d'établir :

- 4 paramètres :
 - un algorithme de chiffrement,
 - une fonction de hachage,
 - une méthode d'authentification
 - un groupe pour Diffie-Hellman
- 3 clés :
 - une pour le chiffrement,
 - une pour l'authentification
 - une pour la dérivation d'autres clés

Main Mode est une instance de l'échange ISAKMP *Identity Protection Exchange*.

136

IKE

Phase 1 : Main Mode

Initiator

Responder



Négociation des paramètres IKE



Génération des valeurs DH et des aléas



Authentification mutuelle



Ces 2 messages
sont chiffrés

137

IKE

Phase 1 : Aggressive Mode

Le mode *Aggressive Mode* est une variante du mode *Main Mode* qui ne contient que 3 messages.

C'est une instance de l'échange ISAKMP *Aggressive Exchange*.

138

IKE

Phase 2 : Quick Mode

Les échanges de cette phase sont protégés en confidentialité et en authenticité grâce à la SA ISAKMP établie lors de la phase 1.

La phase 2 a pour but de mettre en oeuvre les SA (ou les "paquets" de SA) pour IPSec. Chaque négociation donne lieu à deux SA (les SA IPSec étant unidirectionnelle).

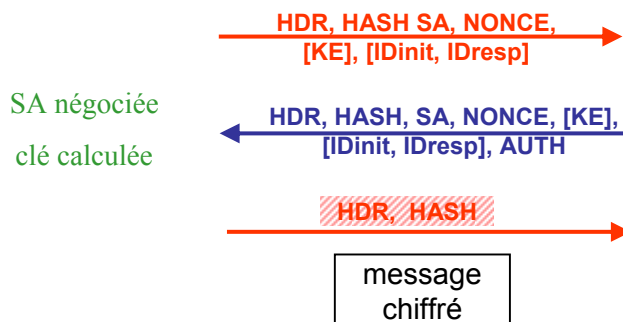
139

IKE

Phase 2 : Quick Mode

Initiator

Responder



140

IKE

Phase 2 : Quick Mode

Un nouvel échange de valeurs Diffie-Hellman a lieu pour respecter la propriété "Perfect Forward Secrecy".

Cet échange est optionnel.

141

IKE

Phase 2 : Quick Mode

Initiator

SPI 101	SPI 102	Empreinte
ESP	ESP	valeurs DH
IDEA	3DES	Aléa
MD5	SHA-1	Idi, IDr

Responder

SPI 102	Empreinte
ESP	valeurs DH
3DES	Aléa
SHA-1	Idi, IDr



142

IKE

New Group Mode

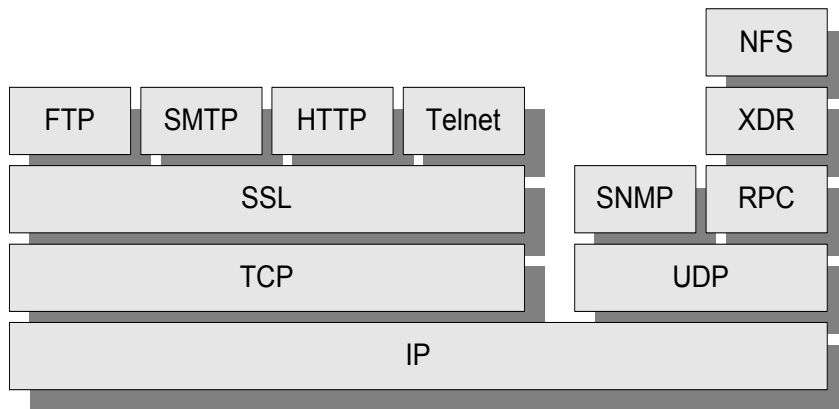
Ce mode sert à négocier le groupe Diffie-Hellman si ce dernier n'a pas été établi durant le *Main Mode*.

SSL/TLS: le transport sécurisé

SSL : Introduction

- SSL
 - Protocole de négociation
 - défini par *netscape* et intégré au browser
- Versions
 - Première version de SSL testée en interne
 - Première version de SSL diffusée : V2 (1994)
 - Version actuelle V3
- Standardisation
 - Standard à l'IETF au sein du groupe Transport Layer Security (TLS). TLS v1.0 correspond à SSL 3.1 (RFC 2246)
 - Standard au sein du WAP Forum Wireless Transport Layer Security (WTLS)

SSL : Architecture



XDR : eXternal Data Representation (RFC 1832)

Ports au dessus de SSL (1/2)

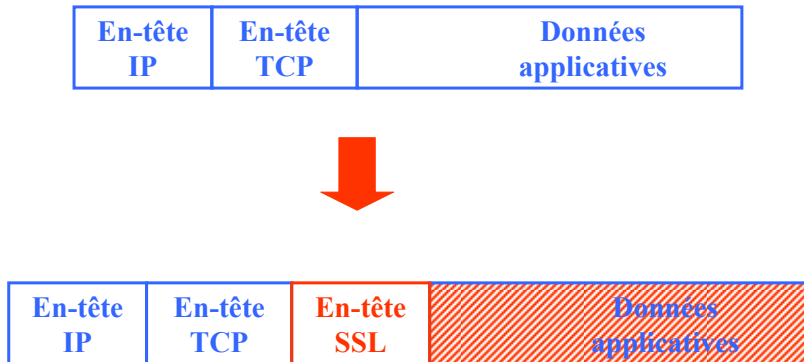
Protocole sécurisé	Port	Protocole non sécurisé	Application
HTTPS	443	HTTP	Transactions requête-réponse sécurisées
SMTP	465	SMTP	Messagerie électronique
NNTP	563	NNTP	News sur le réseau Internet
SSL-LDAP	636	LDAP	Annuaire X.500 allégé
SPOP3	995	POP3	Accès distant à la boîte aux lettres avec rapatriement des messages

Ports au dessus de SSL (2/2)

Protocole sécurisé	Port	Protocole non sécurisé	Application
FTP-DATA	889	FTP	Transfert de fichiers
FTPS	990	FTP	Contrôle du transfert de fichiers
IMAPS	991	IMAP4	Accès distant à la boîte aux lettres avec ou sans rapatriement des messages
TELNETS	992	Telnet	Protocole d'accès distant à un système informatique
IRCS	993	IRC	Protocole de conférence par l'écrit

Encapsulation SSL

Protocole SSL classique



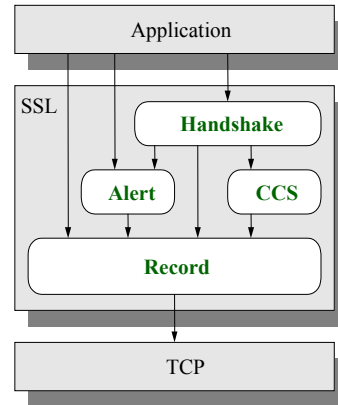
149

SSL : Services

- Authentification
 - Serveur (obligatoire), client (optionnel)
 - Utilisation de certificat X509 V3
 - A l'établissement de la session.
- Confidentialité
 - Algorithme de chiffrement symétrique négocié,
 - clé générée à l'établissement de la session.
- Intégrité
 - Fonction de hachage avec clé secrète : $\text{hmac}(\text{clé secrète}, h, \text{Message})$
- Non Rejeu
 - Numéro de séquence

SSL : Protocoles

- SSL se base sur des sous protocoles
 - SSL handshake (authentification mutuelle du serveur et du client, négociations des algorithmes, négociations des clés de session)
 - SSL Change Cipher Spec
 - SSL Alert (envoi de messages d'erreur - *warning / fatal*)
 - SSL Record (confidentialité et intégrité des données)

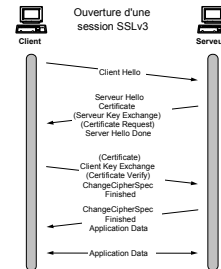


Handshake (1/6)

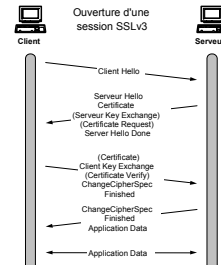
- Authentification du serveur et éventuellement du client,
- Négociation des algorithmes de chiffrement et de hachage, échange d'un secret,
- Génération des clés.
- Pas de consultation systématique d'une CRL

Handshake (2/6)

Message	Type de message	Sens de transmission	Signification
HelloRequest	optionnel	serveur → client	Ce message demande au client d'entamer le Handshake.
ClientHello	obligatoire	client → serveur	Ce message contient : le numéro de version du protocole SSL ; le nombre aléatoire : client_random ; l'identificateur de session : session_ID ; la liste des suites de chiffrement choisies par le client ; la liste des méthodes de compression choisies par le client.
ServerHello	obligatoire	serveur → client	Ce message contient : le numéro de version du protocole SSL ; un nombre aléatoire : serveur_random ; l'identificateur de session : session_ID ; une suite de chiffrement ; une méthode de compression.

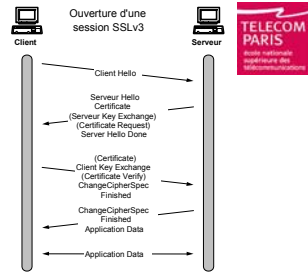


Handshake (3/6)



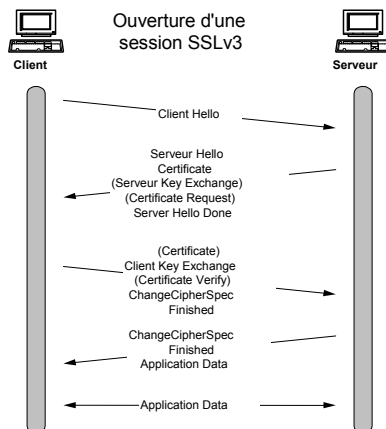
Certificate	Optionnel	serveur → client client → serveur	Ce message contient le certificat du serveur ou celui du client si le serveur le lui réclame et que le client en possède un.
ServerKeyExchange	Optionnel	serveur → client	Ce message est envoyé par le serveur que s'il ne possède aucun certificat, ou seulement un certificat de signature.
CertificateRequest	Optionnel	serveur → client	Par ce message, le serveur réclame un certificat au client.
ServerHelloDone	Obligatoire	serveur → client	Ce message signale la fin de l'envoi des messages ServerHello et subséquents.

Handshake (4/6)

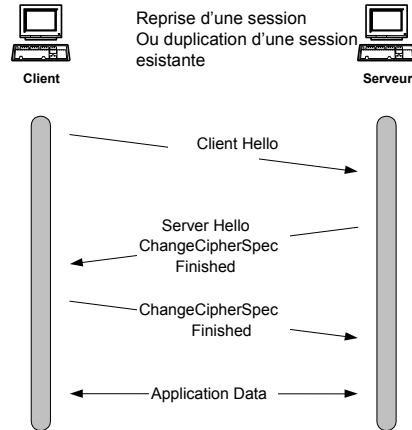


ClientKeyExchange	Obligatoire	client → serveur	Ce message contient le PreMasterSecret crypté à l'aide de la clé publique du serveur.
CertificateVerify	Optionnel	client → serveur	Ce message permet une vérification explicite du certificat du client.
Finished	obligatoire	serveur → client client → serveur	Ce message signale la fin du protocole Handshake et le début de l'émission des données protégées avec les nouveaux paramètres négociés.

Handshake (5/6)



Handshake (6/6)



ChangeCipherSpec (CCS)

- *ChangeCipherSpec* signale au *Record* toute modification des paramètres de sécurité,
- Constitué d'un message (1 octet)

Le protocole *Record*

- Reçoit les données des couches supérieures : (*Handshake*, *Alert*, CCS, HTTP, FTP ...), et les transmet au protocole TCP.
- Après application de :
 - la fragmentation des données en blocs de taille maximum de 2^{14} octets
 - la compression des données, fonction prévue mais non supportée actuellement
 - la génération d'un condensat pour assurer le service d'intégrité
 - le chiffrement des données pour assurer le service de confidentialité

Le protocole *Alert*

- Le protocole *Alert* peut être invoqué :
 - par l'application, par exemple pour signaler la fin d'une connexion
 - par le protocole *Handshake* suite à un problème survenu au cours de son déroulement
- par la couche *Record* directement, par exemple si l'intégrité d'un message est mise en doute

Le protocole *Alert* (2)

Message	Contexte	Type
bad_certificate	échec de vérification d'un certificat	fatal
bad_record_mac	réception d'un MAC erroné	fatal
certificate_expired	certificat périmé	fatal
certificate_revoked	certificat mis en opposition (révoqué)	fatal
certificate_unknown	certificat invalide pour d'autres motifs que ceux précisés précédemment	fatal
close_notify	interruption volontaire de session	fatal
decompression_failure	les données appliquées à la fonction de décompression sont invalides (par exemple, trop longues)	fatal
handshake_failure	impossibilité de négocier des paramètres satisfaisants	fatal
illegal_parameter	un paramètre échangé au cours du protocole Handshake dépasse les bornes admises ou ne concorde pas avec les autres paramètres	fatal
no_certificate	réponse négative à une requête de certificat	avertissement ou fatal
unexpected_message	arrivée inopportune d'un message	fatal
unsupported_certificate	le certificat reçu n'est pas reconnu par le destinataire	avertissement ou fatal

SSL : charges (1/2)

- Les choix pour les calculs de la charge cryptographique de SSL:
 - algorithme de chiffrement du protocole record : DES 64 bits en mode CBC ;
 - algorithme de chiffrement asymétrique : RSA 1024 bits ;
 - fonction de hachage : MD5 ;
 - itinéraire de certification comprenant une seule étape ;
 - certificat du serveur : autorité de certification unique, déjà connue du client (un seul certificat dans le message *Certificate*) ;
 - taille des informations contenues, du message *Certificate* : 500 Koctets (notons que la taille des informations du certificat est dans la plupart des cas inférieure) ;
 - seul le serveur est certifié.

SSL : liste non exhaustive de serveurs

Nom de l'API	Fournisseur	Adresse
AOLserver 2.3	America Online, Inc.	http://www.aolserver.com
Alibaba 2.0	Computer Software Manufacturers	http://www.csm.co.at/alibaba/
Apache 1.3	The Apache Group	http://www.apache.org
Commerce Server/400 1.00	INTERNET, Inc.	http://www.inetmi.com
Enterprise Server 3.0	Novonyx	http://www.novonyx.com
Enterprise Web Secure/VM	Beyond-Software Incorporated	http://www.beyond-software.com
Internet Information Server	Microsoft Corp.	http://www.microsoft.com/iis
Java Server 1.1	Sun Microsystems	http://www.java.sun.com
Lotus Domino Go Webserv	IBM	http://www.ibm.com
4.6.1		
Netscape Enterprise Server 3	Netscape Communications	http://www.netscape.com
Oracle Web Application Serv	Oracle Corp.	http://www.oracle.com/products
3.01		
Roxen Challenger 1.2b I	Idonex	http://www.roxen.com
SSLava	Phaos Technologies	http://www.phaos.com/main.htm
WebSite Professional 2.2	O'Reilly Software	http://www.website.oreilly.com/
WebTen 2.1	Tenon Intersystems	http://www.tenon.com/products/webten
Zeus Web Application Serv	Zeus Technology	http://www.zeustech.net

SSL : liste de suite de chiffrement supportée par un serveur

Serveur et Version			Apache SSLLeay 08.0	Jigsaw 2.0 Beta 1	Microsoft IIS/4.0	Netscape Entreprise3.0L	Netscape Entreprise 3.0F	SSLava Beta 1
Suite	Export	Code						
RSA	RC4-40 MD5	✓	0x03	•	•	•	•	•
	RC4-128 MD5		0x04	•	•	•		•
	RC4- 128 SHA		0x05	•	•	•		•
	RC2 CBC-40 MD5	✓	0x06	•	•	•	•	
	IDEA CBC SHA		0x07	•	•			
	DES40 CBC SHA	✓	0x08	•	•			•
	DESCBC SHA		0x09	•	•	•		•
	3DES EDE CBC SHA		0x0A	•	•	•		
DH et DSA	DES40 CBC SHA	✓		•				
	DES CBC SHA		0x0C	•				
	3DES EDE CBC SHA		0x0D	•				
DH et RSA	DES40 CBC SHA	✓	0x0E	•				
	DES CBC SHA		0x0F	•				
	3DES EDE CBC SHA		0x10	•				
DHE et DSA	DES40 CBC SHA	✓	0x11	•				
	DES CBC SHA		0x12	•				
	3DES EDE CBC SHA		0x13	•				
DHE et RSA	DES40 CBC SHA	✓	0x14	•	•			
	DES CBC SHA		0x15	•	•			
	3DES EDE CBC SHA		0x16	•	•			

SSL : liste non exhaustive d'APIs

Nom de l'API	Fournisseur	Adresse
AOLserver 2.3	America Online, Inc.	http://www.aolserver.com
Alibaba 2.0	Computer Software Manufacturers	http://www.csm.co.at/alibaba/
Apache 1.3	The Apache Group	http://www.apache.org
Commerce Server/400 1.00	INTERNET, Inc.	http://www.inetmi.com
Enterprise Server 3.0	Novonyx	http://www.novonyx.com
Enterprise Web Secure/VM	Beyond-Software Incorporated	http://www.beyond-software.com
Internet Information Server	Microsoft Corp.	http://www.microsoft.com/iis
Java Server 1.1	Sun Microsystems	http://www.java.sun.com
Lotus Domino Go Webserv 4.6.1	IBM	http://www.ibm.com
Netscape Enterprise Server 3	Netscape Communications	http://www.netscape.com
Oracle Web Application Serv 3.01	Oracle Corp.	http://www.oracle.com/products
Roxen Challenger 1.2b 1	Idonex	http://www.roxen.com
SSLava	Phaos Technologies	http://www.phaos.com/main.htm
WebSite Professional 2.2	O'Reilly Software	http://www.website.oreilly.com/
WebTen 2.1	Tenon Intersystems	http://www.tenon.com/products/webten
Zeus Web Application Serv	Zeus Technology	http://www.zeustech.net

Attaques classiques

Attaque classique	Parade SSL
Casser les clés	Taille des clés
Attack replay	Nonces (connection id)
Man in the middle	Certificats servent à passer les clés
Attaque à clair ouvert	clés + Aléas

SSL avantages/inconvénients



Avantages :

- il n'y a plus de problème de NAT/NAPT
- les logiciels clients supportant ces protocoles sont de plus en plus répandus

Inconvénients :

- il faut que l'application supporte la librairie SSL
- le protocole est souvent implémenté de manière incomplète (pas d'authentification client par certificat pour IMAPS et SMTPS)

174

SSH



SSH (Secure Shell)

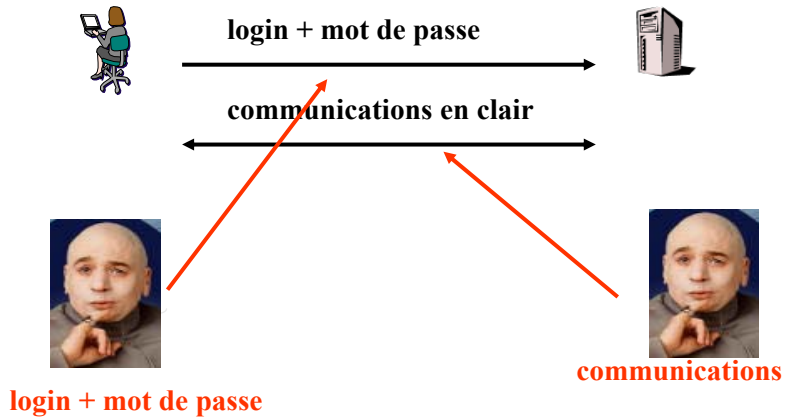
Niveau application

Protocole client / serveur

A l'origine, SSH est utilisé comme un service telnet sécurisé : le mot de passe ne circule pas en clair et les échanges sont chiffrés.

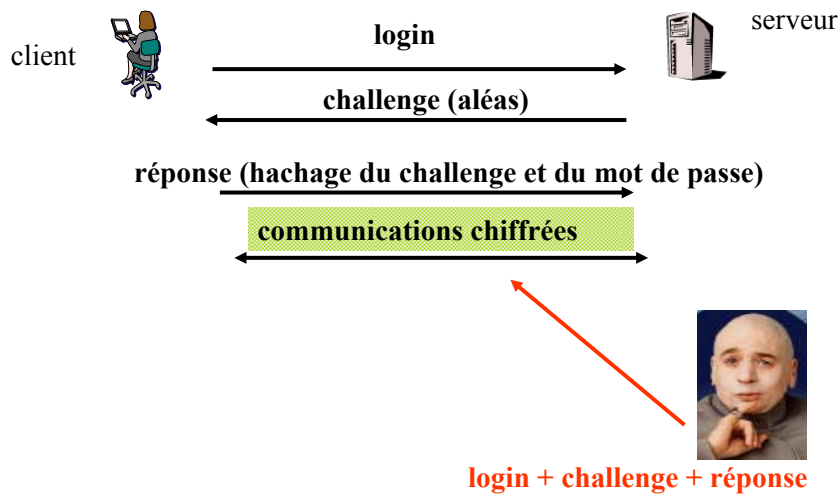
175

Telnet



176

SSH



177

Avantages de SSH

- chiffrement de la session et mot de passe unique (*OTP*)
- le client SSH est disponible sur un très grand nombre de plateformes
- la sécurité est assurée depuis le client jusqu'au serveur (*end-to-end security*)

178

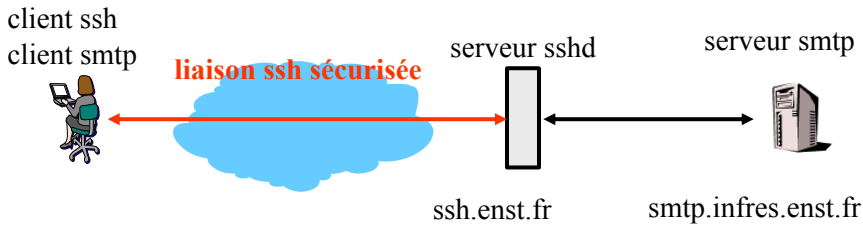
SSH: port forwarding

Le *Port Forwarding* (ou transfert de port) permet d'utiliser une liaison SSH pour transporter des protocoles non sécurisés (POP, NNTP, ...).

On peut alors construire un VPN basé sur des liaisons SSH.

179

SSH: port forwarding



180

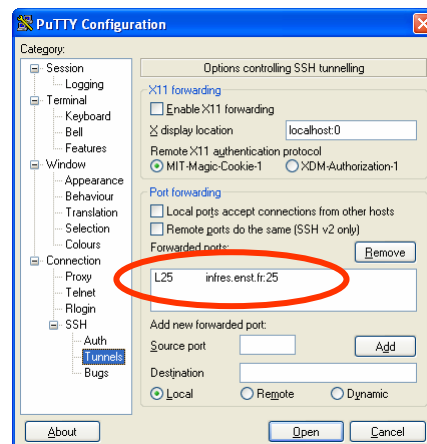
SSH: port forwarding

Exemple de configuration sur un poste Linux :

```
$ ssh -L 4444:smtp.infres.enst.fr:25  
ssh.enst.fr
```

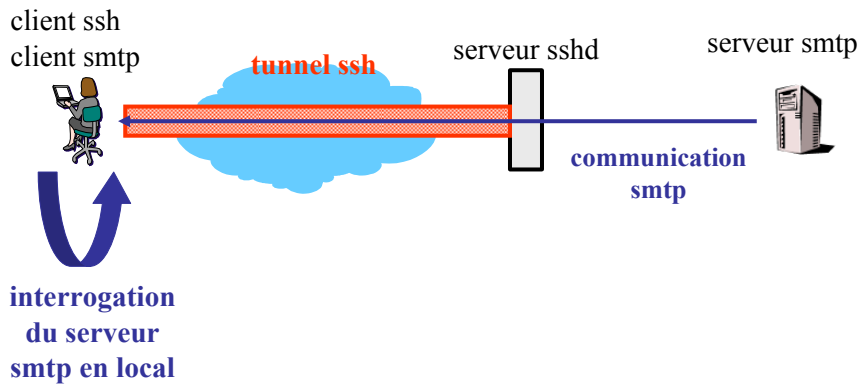
puis on configure le client nntp de la manière suivante :

- serveur : localhost
- port : 4444



181

SSH: port forwarding



182

SSH: port forwarding

Avantages : - on conserve les applications en « standard » (clients et serveurs)

Inconvénients : - gestion manuelle des clés

- le numéro de port utilisé par le protocole doit être fixe.
- tous les protocoles ne sont pas triviaux à implémenter (ex : ftp/ftp-data)

183

Résumé

- IPSec
- La distribution de clés
- Le transport sécurisé

Questions ?

VPN - Réseaux Privés Virtuels

Tables des matières

1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Eléments de comparaison des VPN
6. Exemples d'implémentation

Rappel du plan



1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Eléments de comparaison des VPN
6. Exemples d'implémentation

199

Présentation des VPN



Le rôle d'un réseau privé virtuel (VPN, Virtual Private Network) est d'étendre un réseau privé en exploitant de manière sécurisée un réseau public.

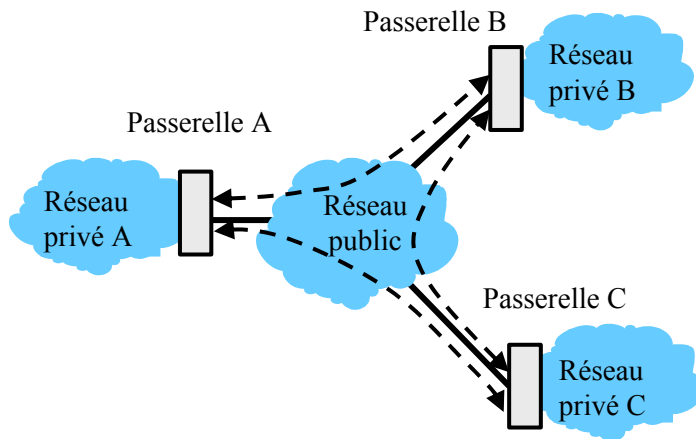
Un VPN permet par exemple :

- de relier plusieurs entités ;
- de sécuriser les connexions depuis un FAI ;
- d'intégrer à un réseau privé des utilisateurs itinérants.

200

Présentation des VPN

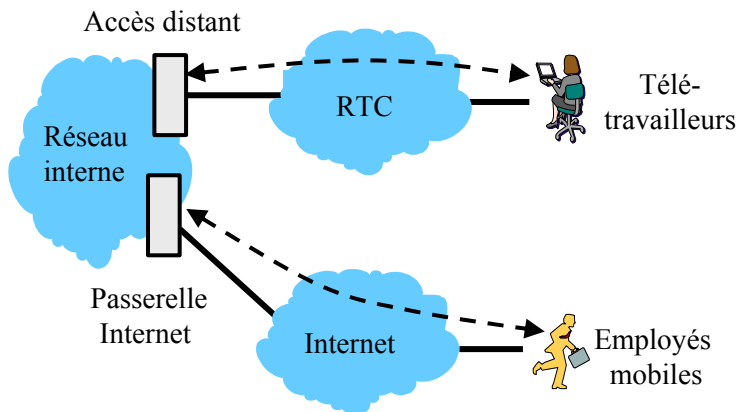
Exemple 1



201

Présentation des VPN

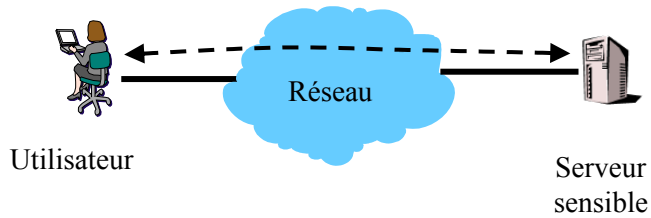
Exemple 2



202

Présentation des VPN

Exemple 3



203

Présentation des VPN

Exemple 4



204

Présentation des VPN



Auparavant, il n'y avait que deux solutions :

- relier deux entités distantes
par une ligne spécialisée (LS) => cher !
- utiliser le RTC => pas de sécurité !

Un VPN permet d'étendre un WAN ou de connecter des utilisateurs itinérants à moindre coût, tout en garantissant la sécurité des échanges et la qualité de service.

205

Présentation des VPN



Méthodes :

- Chiffrement des communications

=> confidentialité
- Authentification mutuelle des correspondants et contrôle d'intégrité des données

=> authenticité

206

Présentation des VPN



On distingue trois types de VPN :

- les VPN à accès distant (Remote Access VPN) qui permet de relier les télétravailleurs et les employés mobiles ;
- les VPN Intranet qui met en relation plusieurs sites à l'intérieur d'une même organisation ;
- les VPN Extranet qui ouvre une partie de l'intranet de l'entreprise aux partenaires (clients, fournisseurs, ...).

207

Présentation des VPN



Il y a deux modes d'établissement d'un VPN :

- mode "compulsory" (passerelle)
- mode "voluntary" (client)

208

Présentation des VPN

mode “compulsory” :



209

Présentation des VPN

mode “voluntary” :



On ne dépend plus du fournisseur d'accès, mais il est nécessaire d'avoir des configurations des postes clients précises et à jour.

210

Présentation des VPN

A quel niveau déployer un VPN ?

Application
Présentation
Session
Transport
Réseau
Liaison
Physique

211

Présentation des VPN

A quel niveau déployer un VPN ?

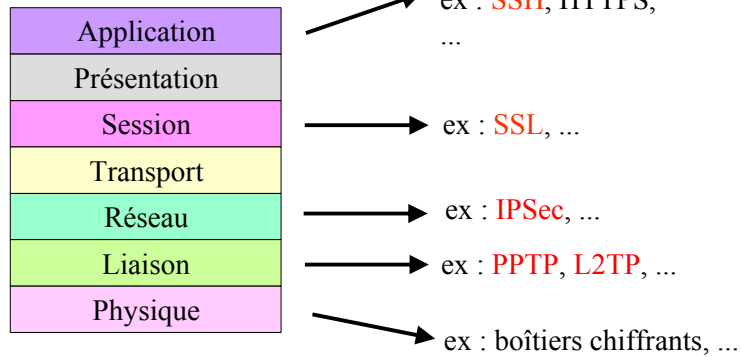
Répondre aux questions suivantes :

- quel contenu doit être sécurisé ?
- qui utilisera le VPN ?
- avec quels moyens ?
- à partir d'où ?

212

Présentation des VPN

A quel niveau déployer un VPN ?



213

Présentation des VPN

Boîtier chiffant : chiffre tous les paquets IP qui transitent.



214

La plupart des VPN se basent sur l'utilisation de “tunnels”.

La tunnelisation (ou *tunneling*) repose sur l'utilisation d'un réseau public, où les échanges sont sécurisés.

Il y a encapsulation des données à transporter dans les paquets du protocole de tunnelisation.

1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Eléments de comparaison des VPN
6. Exemples d'implémentation

Rappel du plan



1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Eléments de comparaison des VPN
6. Exemples d'implémentation

217

Rappels



Le protocole PPP (Point-to-point Protocol)

RFC 1331

C'est un protocole de liaison qui permet l'échange de données sur une liaison point à point.

Il utilise HDLC comme base d'encapsulation pour les protocoles réseaux (IP, IPX, NetBEUI, ...).

PPP peut être associé à IPCP (Internet Protocol Control Protocol - RFC 1332) pour associer à un hôte distant une adresse IP dynamique.

218

Rappels



Format de la trame PPP :

Fanion 01111110	Adresse 11111111	Contrôle 00000011	Protocole 16 bits	Données
	FCS 16 bits	Fanion 01111110		

219

Rappels



GRE (Generic Routing Encapsulation)

RFC 1701, puis RFC 2784.

GRE s'utilise pour encapsuler un protocole dans un autre (par ex. : IP dans IP).

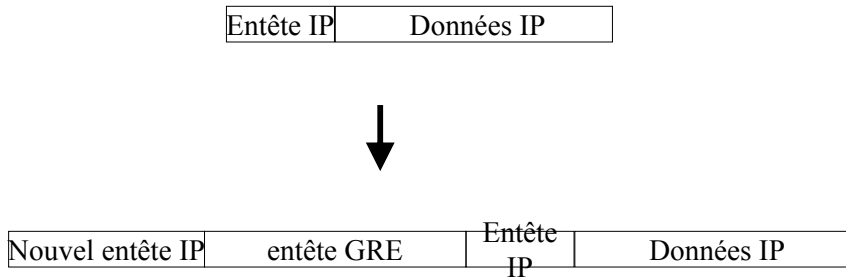
En-tête GRE :

C	Reserved0	Ver	Protocol Type
	Checksum (optional)		Reserved1 (Optional)

220

Rappels

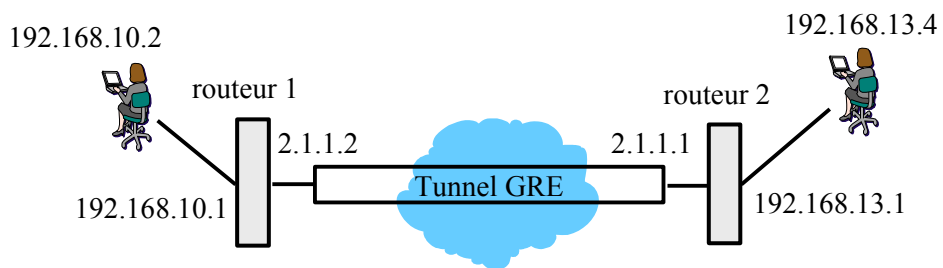
Encapsulation GRE :



221

Rappels

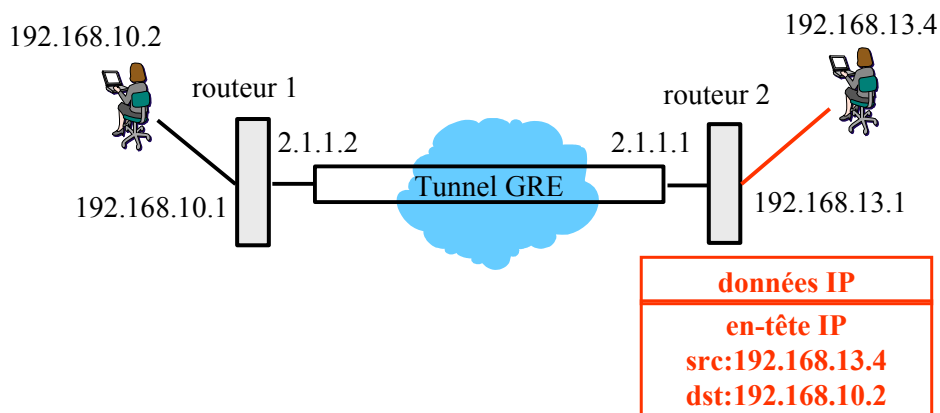
Exemple d'utilisation du protocole GRE :



222

Rappels

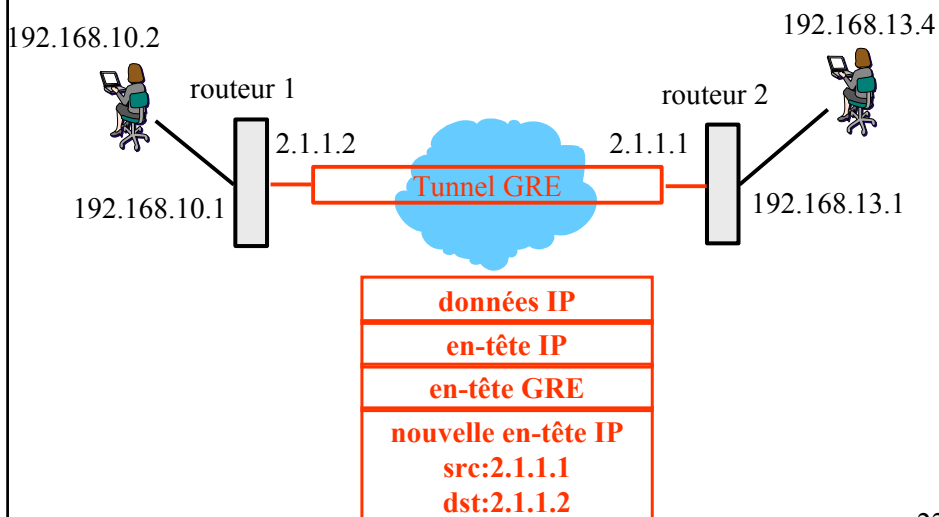
Exemple d'utilisation du protocole GRE :



223

Rappels

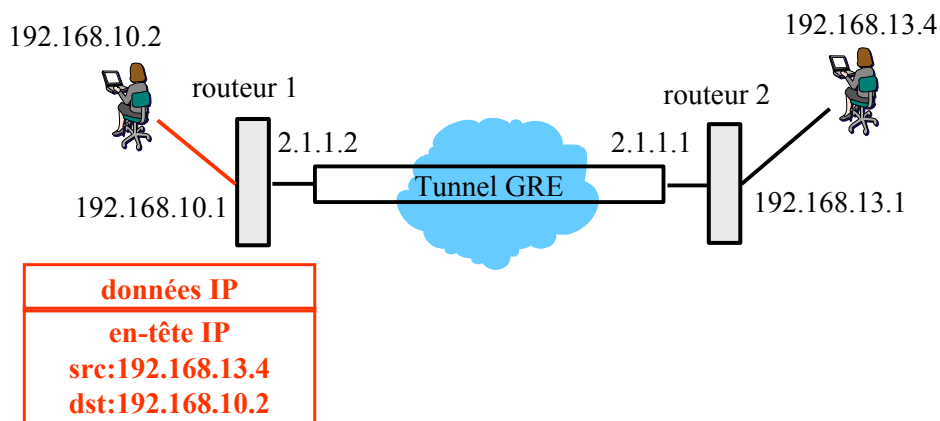
Exemple d'utilisation du protocole GRE :



224

Rappels

Exemple d'utilisation du protocole GRE :



225

Rappel du plan

1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Éléments de comparaison des VPN
6. Exemples d'implémentation

226

PPTP



PPTP (Point-to-Point Tunneling Protocol) est un protocole développé par Ascend, Microsoft, 3COM, ECI Telematics et US Robotics.

RFC 2637

C'est un protocole de niveau 2 qui permet d'encapsuler des trames PPP dans des paquets IP, afin de les transférer sur un réseau IP.

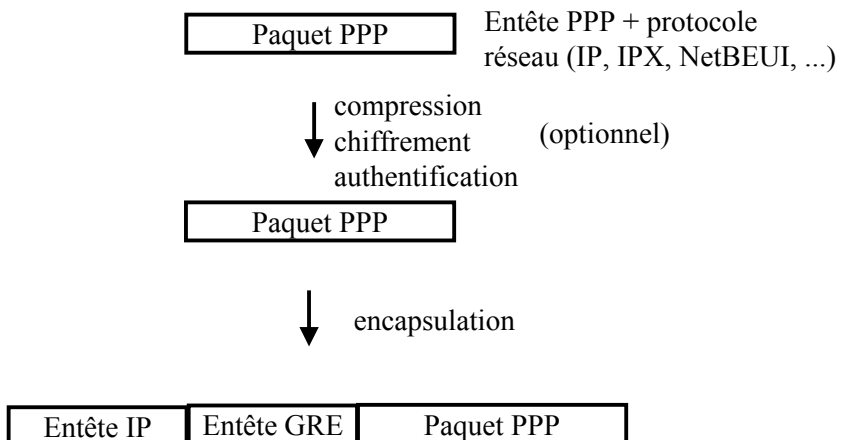
Il est possible de compresser et de chiffrer les données.

Il y a 2 composantes :

- la connexion de contrôle entre client et serveur (tcp/1723) ;
- l'encapsulation de PPP dans IP via GRE (protocole : 47).

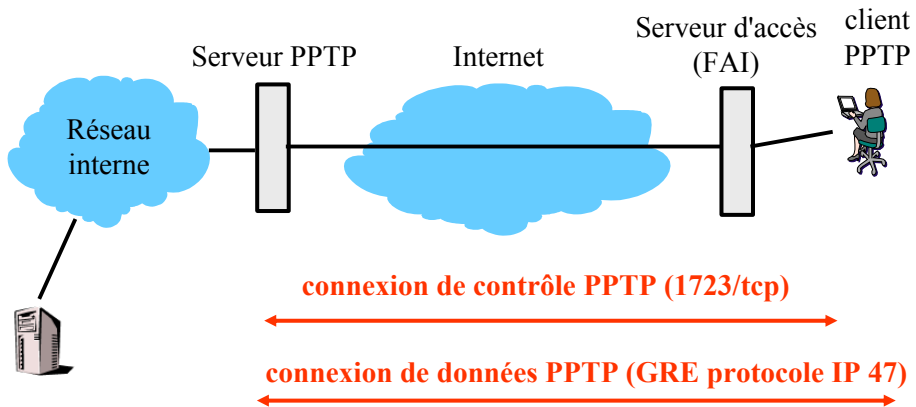
227

PPTP



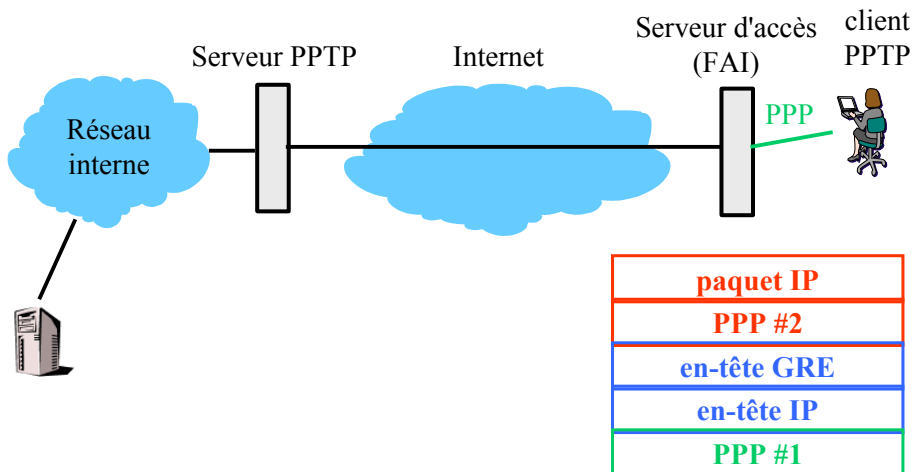
228

PPTP



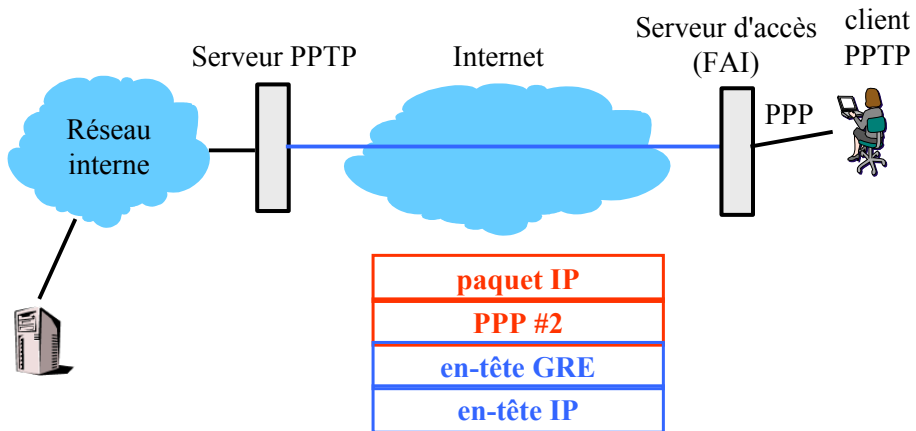
229

PPTP



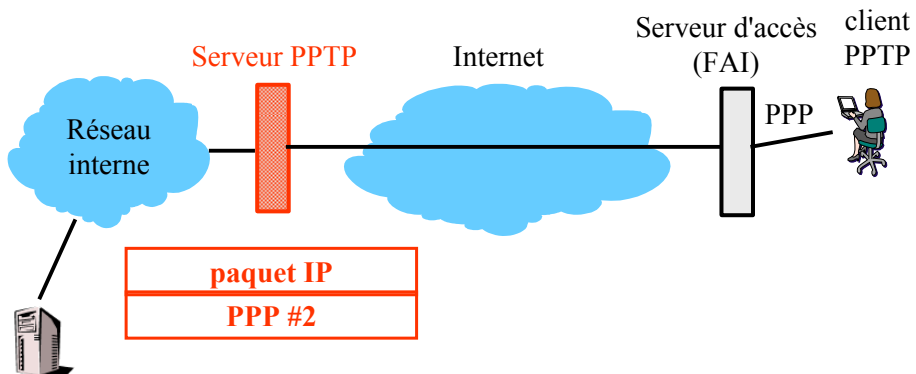
230

PPTP



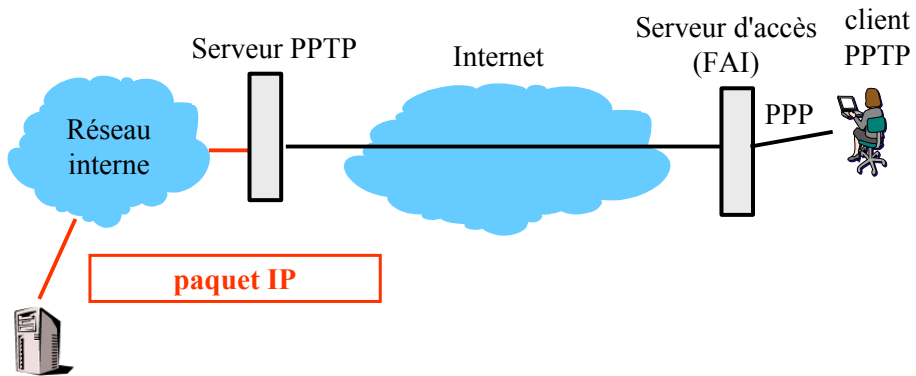
231

PPTP



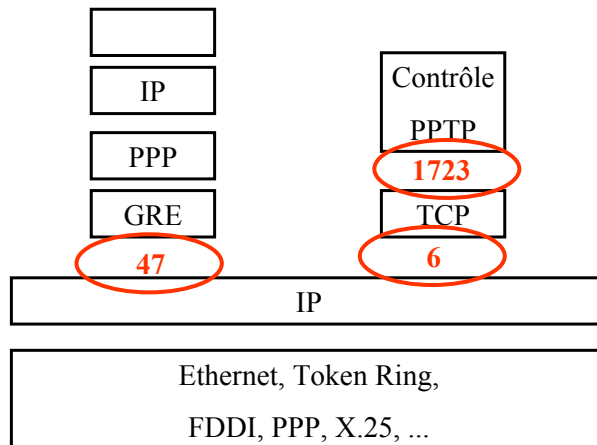
232

PPTP



233

PPTP



234

PPTP



Disponibilité :

- serveur : Windows NT Server, Linux, FreeBSD, OpenBSD, ...
- client : Windows NT, Windows 9x, Windows 2000, Linux, FreeBSD, OpenBSD, ...

Sécurité :

Microsoft a développé sa propre version du protocole : Microsoft PPTP.

On y trouve une extension d'authentification (MS-CHAPv2) et une extension de chiffrement (MPPE).

Performances :

Attention à la montée en charge...

235

PPTP



Résumé :

- le tunnel est initié par le client ;
- le tunnel est terminé par un serveur ;
- il existe une connexion de contrôle entre le client et le serveur ;
- les paquets PPP sont encapsulés dans le protocole IP.

236

Rappel du plan



1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Eléments de comparaison des VPN
6. Exemples d'implémentation

237

L2F



L2F (Layer Two Forwarding) est un protocole développé par Cisco.

RFC 2341

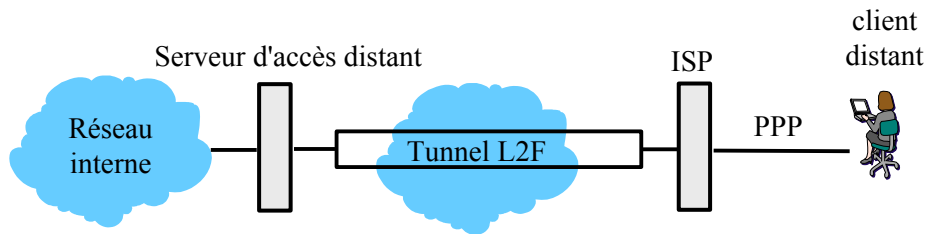
Proche de PPTP, il permet d'acheminer une connexion PPP sur une machine distincte de celle où se trouve l'interface physique.

Il y a 2 composantes :

- une connexion PPP entre le client et le FAI ;
- un tunnel L2F entre un serveur distant et le FAI pour “faire suivre” la connexion PPP (encapsulation avec entête et checksum).

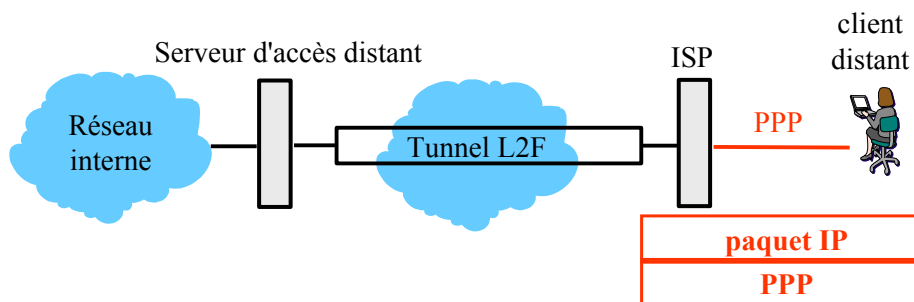
238

L2F



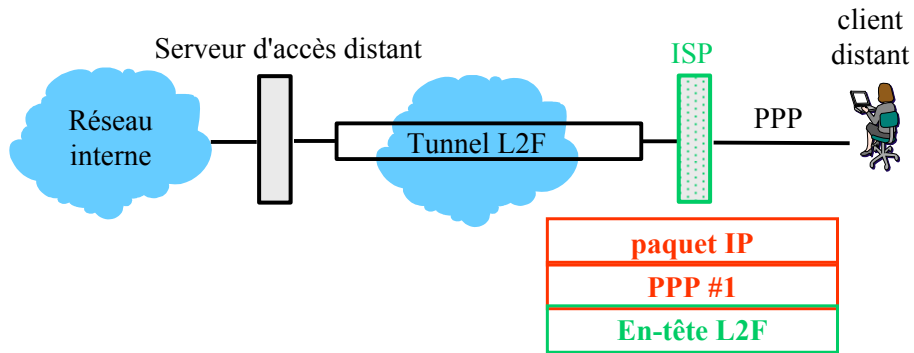
239

L2F



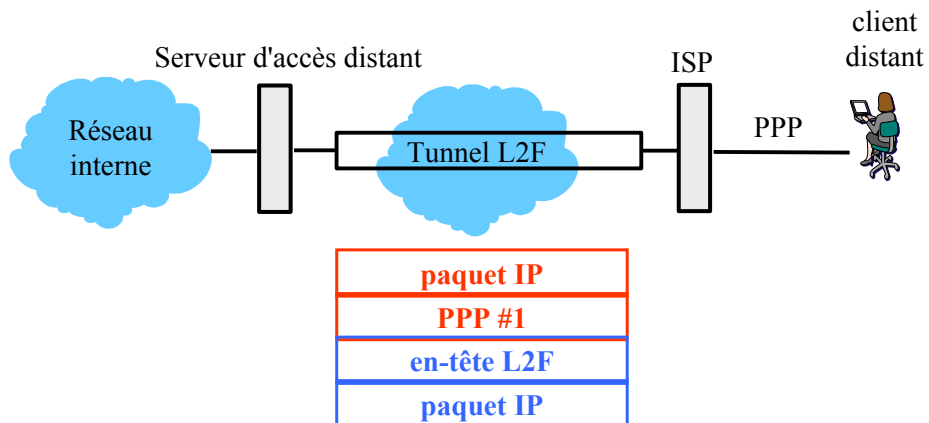
240

L2F



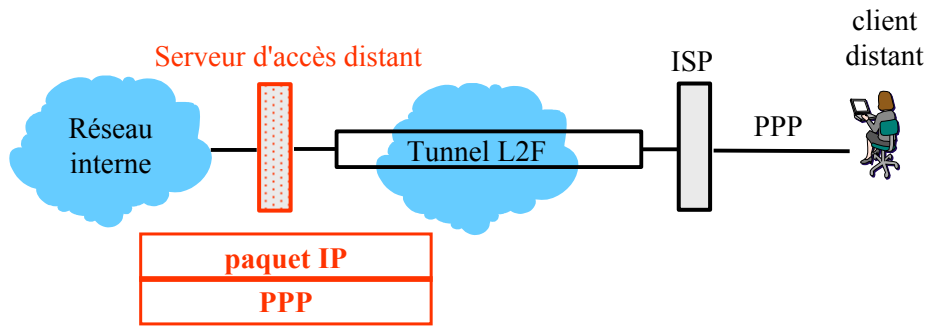
241

L2F



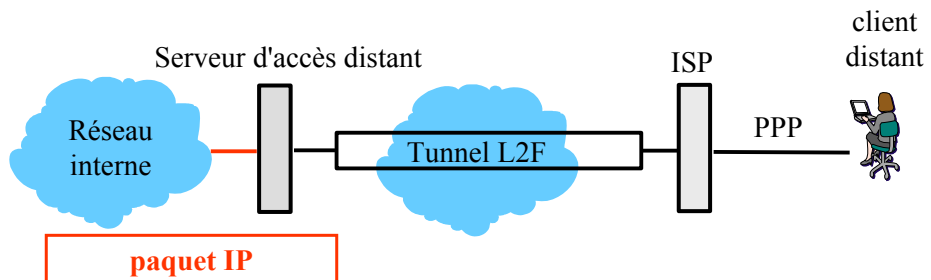
242

L2F



243

L2F



244

Résumé :

- les clients se connectent en PPP ;
- les tunnels sont initiés par l'ISP ou l'opérateur ;
- les tunnels sont terminés par un serveur ;
- il n'y a pas de chiffrement ;
- authentification possible par l'ISP et/ou la passerelle de l'entreprise.

Rappel du plan

1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Éléments de comparaison des VPN
6. Exemples d'implémentation

L2TP



L2TP (Layer Two Tunneling Protocol) a été développé par Cisco et Microsoft pour reprendre les avantages de PPTP et de L2F.

RFC 2661

Ce protocole permet :

- d'établir dynamiquement, de maintenir et de terminer des connexions PPP ;
- d'encapsuler des trames PPP dans divers protocoles (UDP, ATM, ...).

Il y a deux composantes dans l'architecture d'un tunnel L2TP :

- le LAC (L2TP Access Concentrator) ;
- le LNS (L2TP Network Server).

247

L2TP



Le LAC est l'une des deux extrémités du tunnel L2TP. C'est la terminaison physique de la connexion PPP venant du système distant.

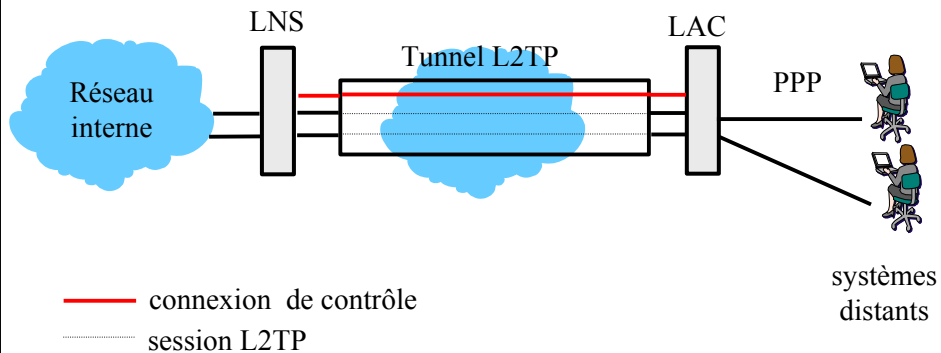
Le LNS est l'autre extrémité de ce tunnel. Il s'agit de la terminaison logique de la connexion PPP.

La connexion de contrôle : connexion dans la bande qui sert à transmettre les messages de contrôle pour l'établissement, la suppression et la maintenance des sessions et du tunnel.

On utilise le protocole UDP.

248

L2TP



249

L2TP

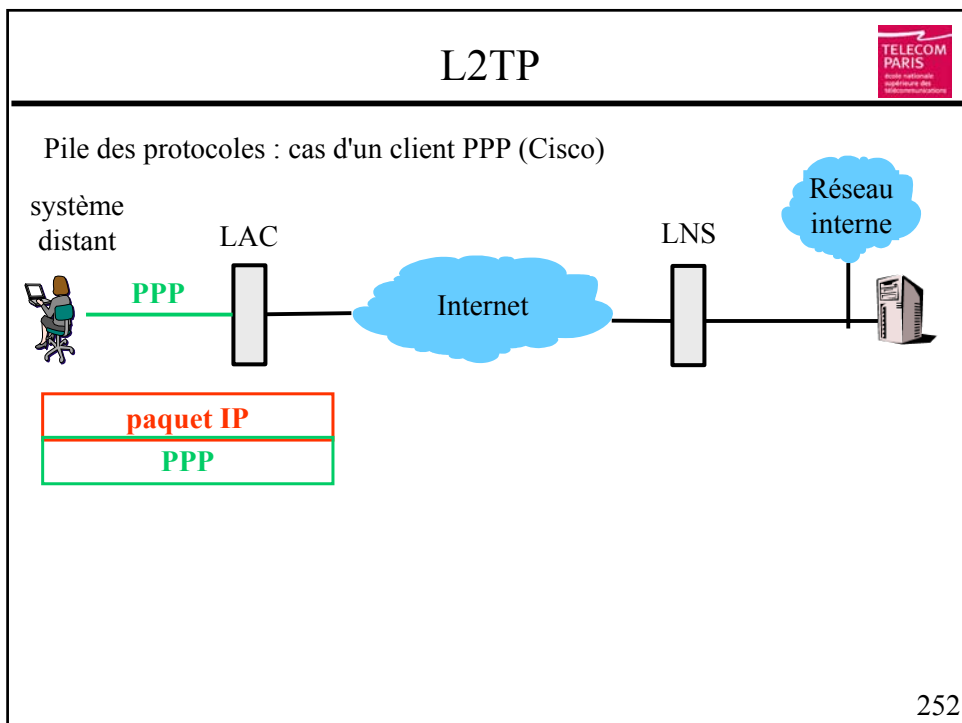
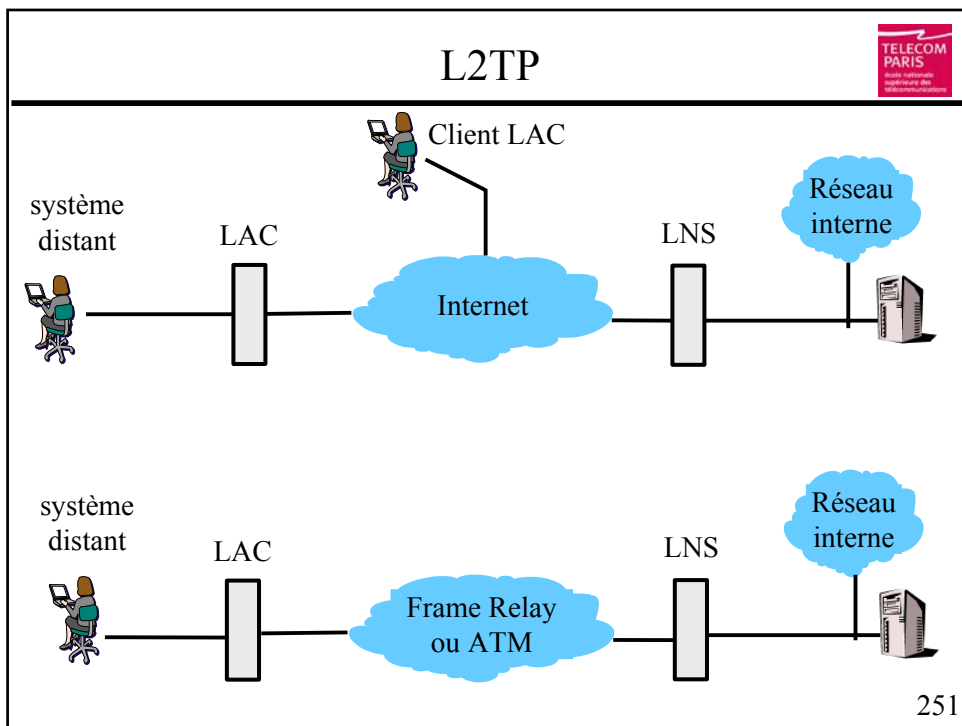
L'établissement du tunnel se déroule en 2 étapes successives :

- l'établissement d'une connexion de contrôle ;
- l'établissement d'une session L2TP.

Lors de l'établissement de la connexion de contrôle, il peut y avoir une phase d'authentification du tunnel, basé sur le protocole PPP CHAP (PPP Challenge Handshake Authentication Protocol - RFC 1994).

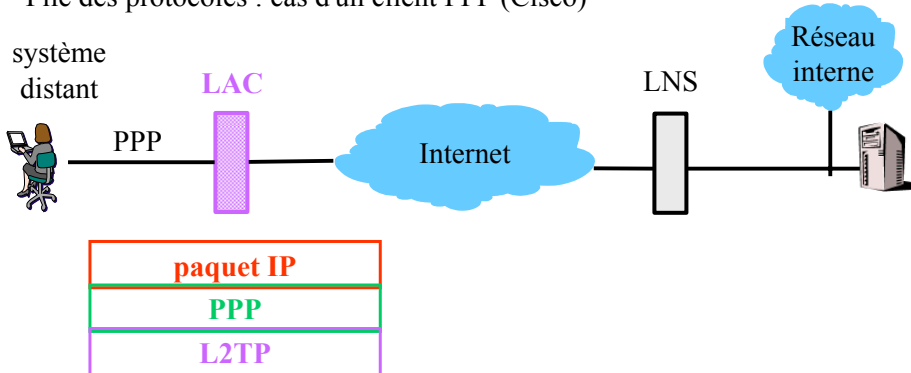
Note : il peut y avoir de multiples sessions L2TP au sein d'un même tunnel, et de multiples tunnels peuvent exister entre une même paire LAC/LNS.

250



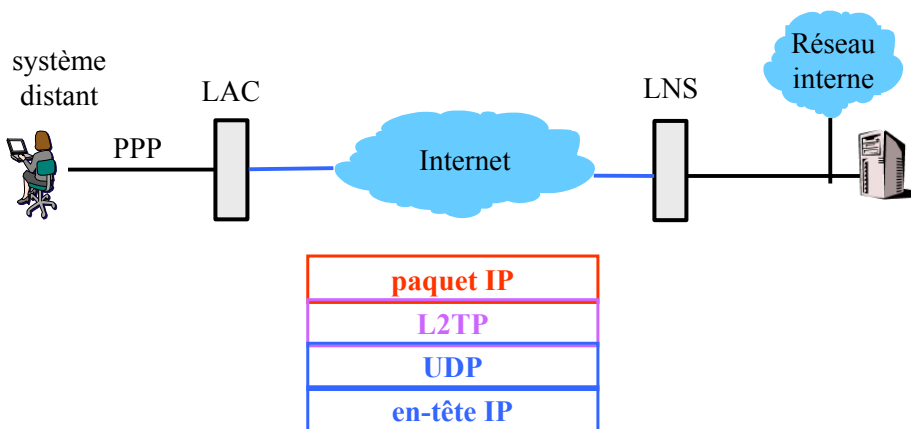
L2TP

Pile des protocoles : cas d'un client PPP (Cisco)



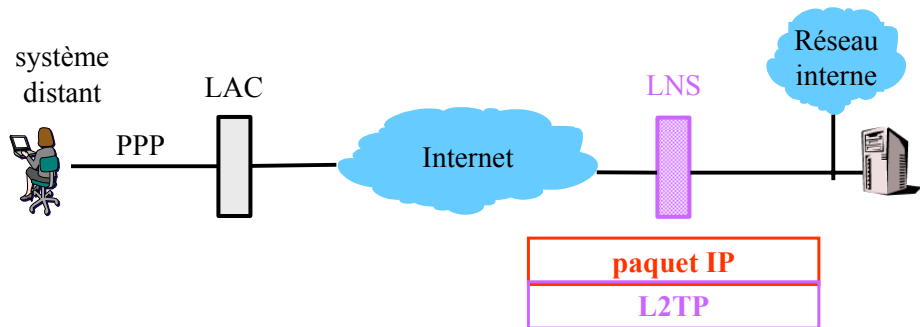
253

L2TP



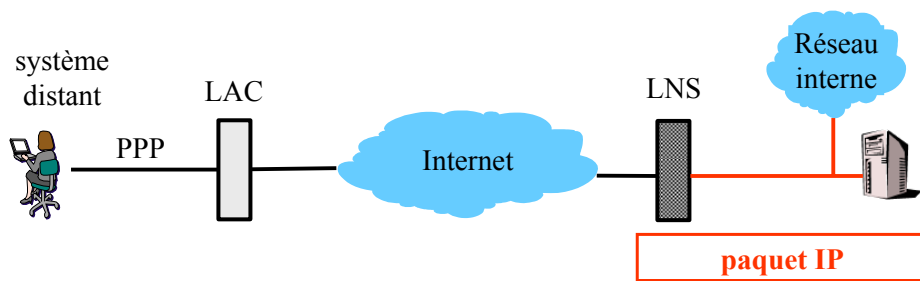
254

L2TP



255

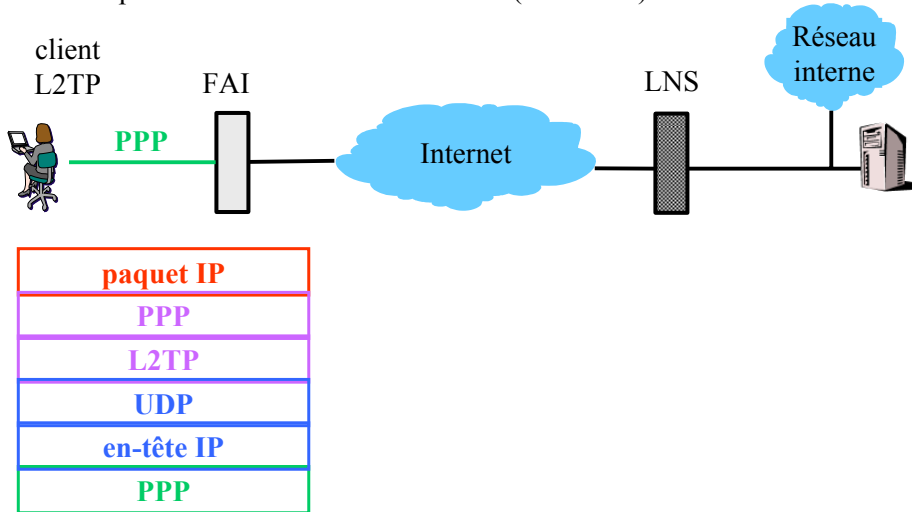
L2TP



256

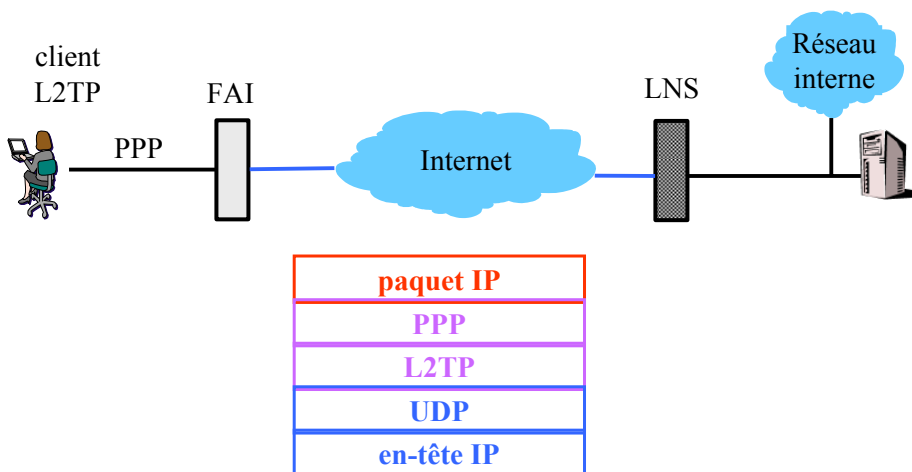
L2TP

Pile des protocoles : cas d'un client L2TP (Microsoft)



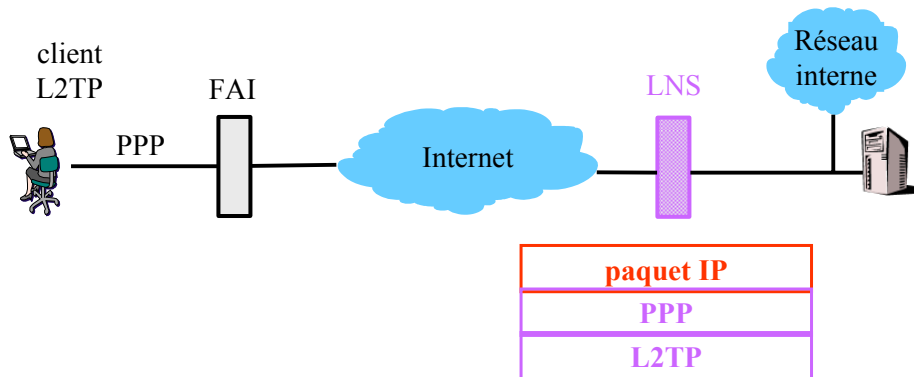
257

L2TP



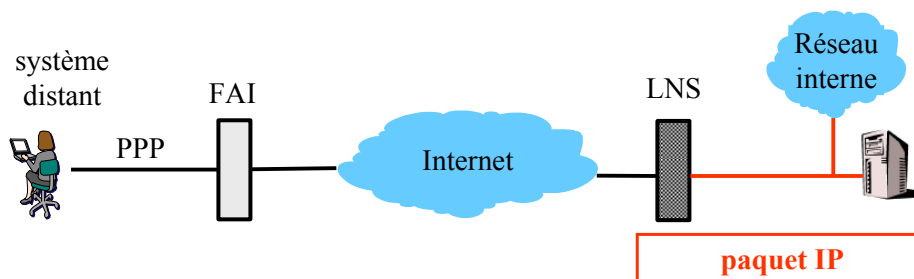
258

L2TP



259

L2TP



260

Encapsulation dans le protocole IP

Lors d'une encapsulation des trames PPP dans le protocole IP, la solution par défaut est l'utilisation du protocole transport UDP, sur le port 1701.

Ce port est également utilisé par L2F. Le numéro de version (1 pour L2F, 2 pour L2TP) sert à différencier les deux types de paquets.

Rappel du plan

1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Eléments de comparaison des VPN
6. Exemples d'implémentation

Rappel du plan

1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Eléments de comparaison des VPN
6. Exemples d'implémentation

263

IPSec

Le protocole utilisé pour déployer des VPN au niveau réseau est IPSec.

Avec IPSec, chaque paquet IP est chiffré et/ou authentifié.

Ce protocole est inclus dans la pile TCP/IP (obligatoire dans IPv6, optionnel dans IPv4), et peut donc être mis en oeuvre sur tout équipement du réseau : routeur, serveur, station de travail...

Il existe deux modes :

- transport : en-tête non modifié
- tunnel : encapsulation dans un nouveau paquet IP

264

Les deux mécanismes d'IPSec servant à protéger les paquets transférés sont AH et ESP.

Les paramètres relatifs à ces mécanismes sont stockés dans les SA (associations de sécurité).

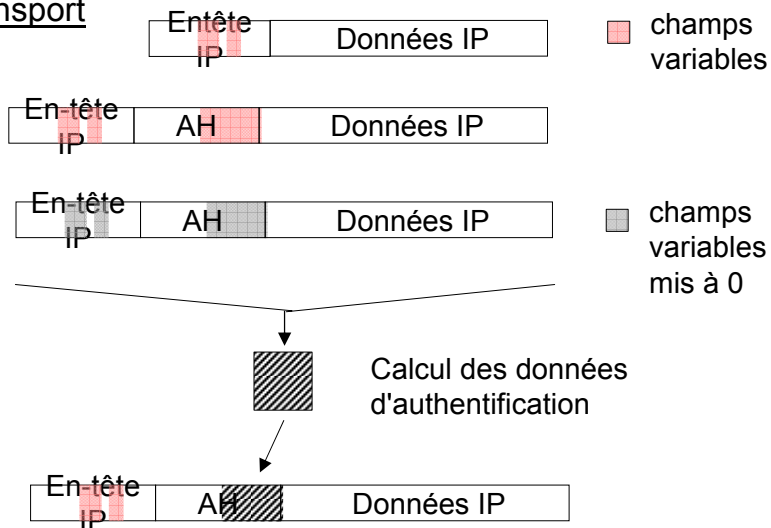
Pour gérer les associations de sécurité, le protocole IKE (Internet Key Exchange) est utilisé.

Mode transport : entre 2 correspondants finaux

Mode tunnel : entre 2 passerelles ou entre 2 correspondants finaux

IPSec - AH

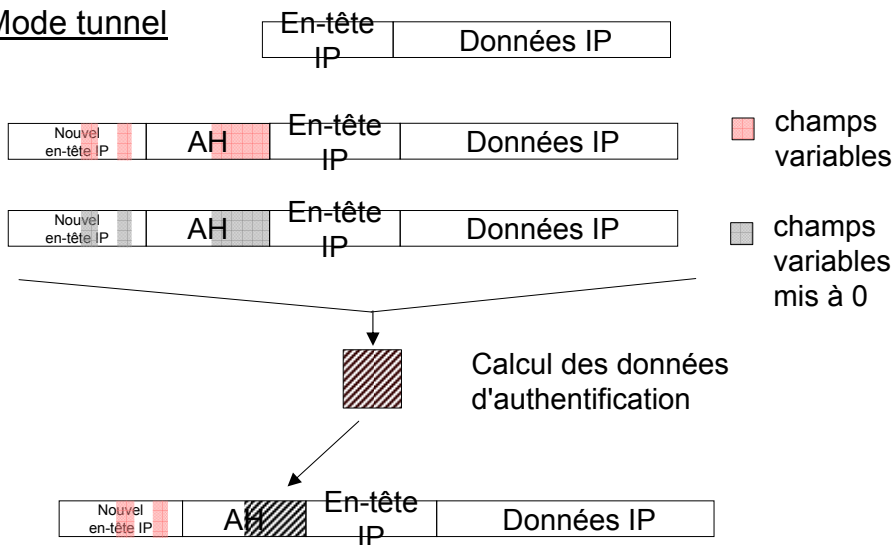
Mode transport



267

IPSec - AH

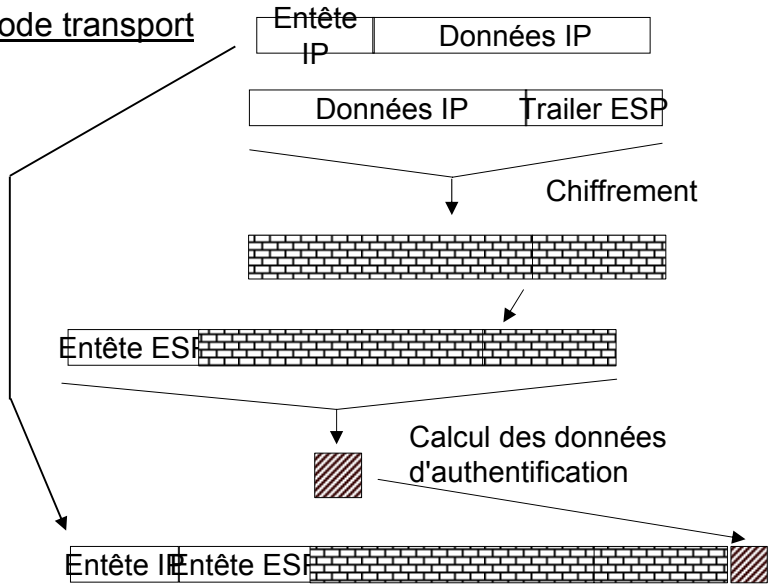
Mode tunnel



268

IPSec - ESP

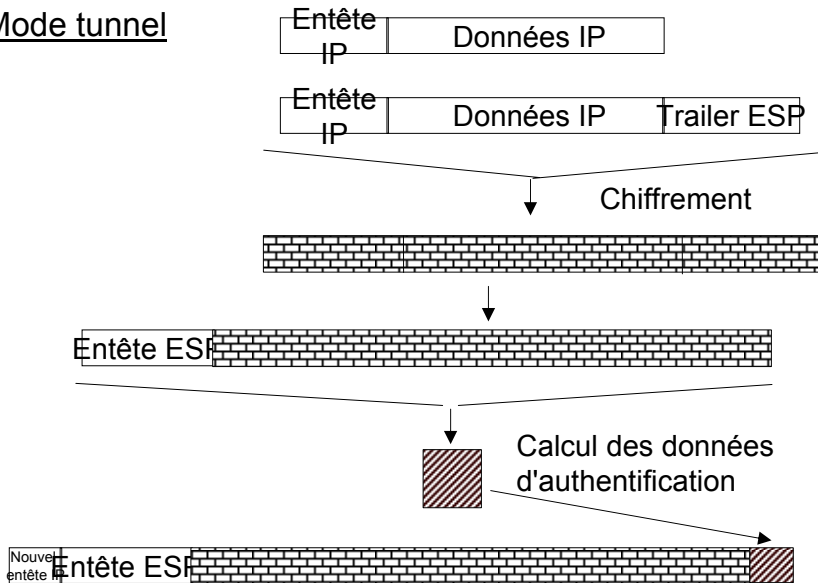
Mode transport



269

IPSec - ESP

Mode tunnel



270

Gestion des performances :

- RSVP (Ressource Reservation Protocol) : on réserve dynamiquement de la bande-passante pour une application ou un utilisateur spécifique
- MPLS (Multi Protocol Label Switching)

Tests d'interopérabilité

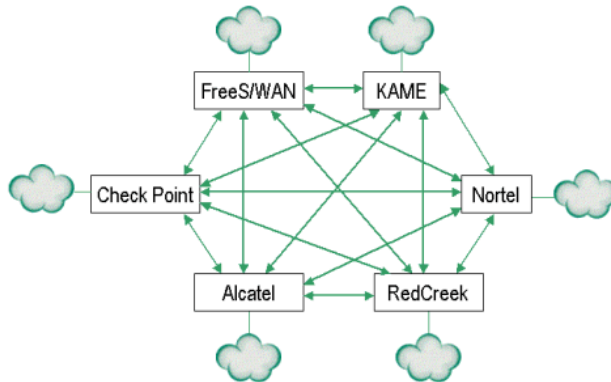
Effectués par le cabinet HSC
dans le cadre de la conférence IPsec2000
(24-27 octobre 2000)
et
IPsec 2001
(23-26 octobre 2001)

<http://www.hsc.fr/ipsec/ipsec2001>

Tests d'interopérabilité

Tests avec secret partagé (2000)

Results: PSK



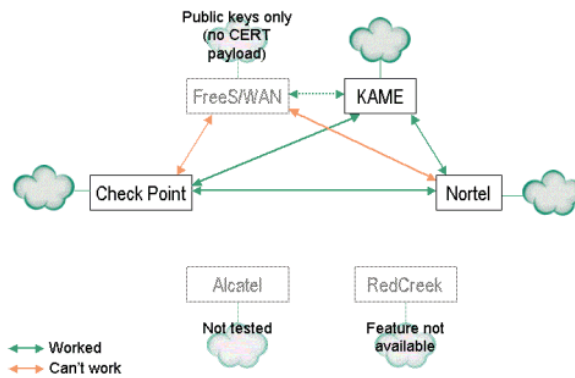
IPsec 2000 Interoperability demo — © Labouret — © 2000, Hervé Schauer Consultants (www.hsc.fr)

273

Tests d'interopérabilité

Tests avec signature RSA (2000)

Results: RSA-Sig



IPsec 2000 Interoperability demo — © Labouret — © 2000, Hervé Schauer Consultants (www.hsc.fr)

274

Tests d'interopérabilité

Tests avec signature RSA (2001)

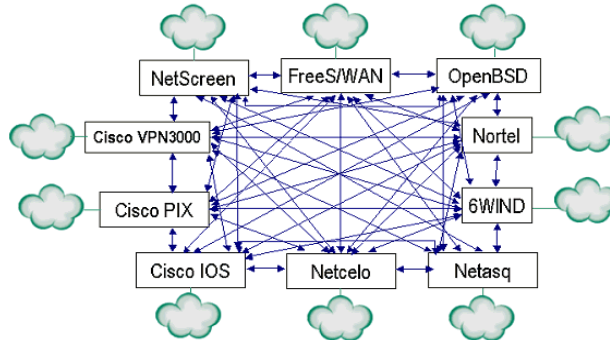


IPsec 2001 Interoperability Demonstration



Test Setup = Fully Meshed VPN

- $10 \times 9 / 2 = 45$ tunnels, 90 tests



© Labouret

© 2001, Hervé Schauer Consultants

6

1 seul échec (invitation : Nortel, réponse : OpenBsd)
5 semi-succès (principalement OpenBSD), 84 succès

275

Tests d'interopérabilité

Conclusion

Interopérabilité IPsec :
Peu de problème depuis un moment

Interopérabilité IKE :
Beaucoup de progrès ont été faits

276

Les accès distants avec IPSec

Le principal problème avec IPSec est d'obtenir une adresse interne.

De plus, il y a une restriction des modes IKE utilisables avec un adressage dynamique.

Il existe un projet : IPSRA (IPSec Remote Access).

277

Rappel du plan

1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Eléments de comparaison des VPN
6. Exemples d'implémentation

278

CIPE

Protocole développé en 1996 par Olaf Titz.

Principe : on encapsule les paquets IP dans des datagrammes UDP.

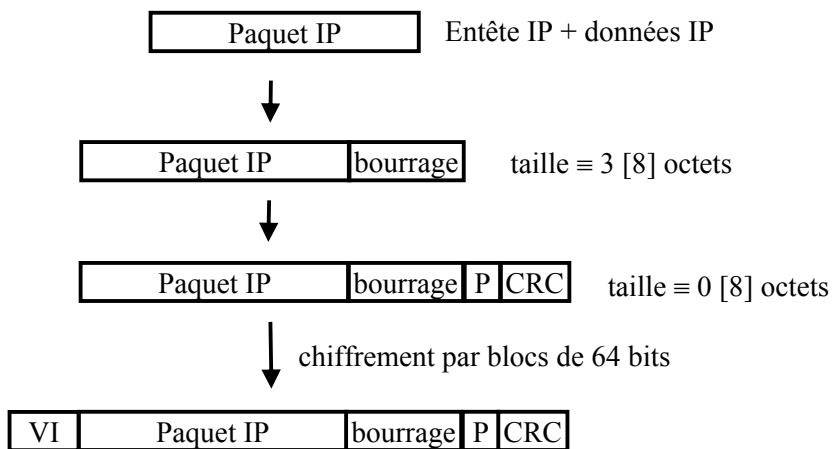
Il y a deux parties dans le protocole :

- le chiffrement et le calcul d'une "checksum" ;
- l'échange dynamique de clés.

279

CIPE

Le chiffrement



VI = Vecteur d'Initialisation

280

L'échange dynamique des clés

Il y a trois types de messages :

- *NK_REQ* pour demander une négociation des clés ;
- *NK_IND* pour envoyer une clé et son CRC ;
- *NK_ACK* pour accuser réception d'un clé (contient son CRC).

Chaque interface possède trois clés :

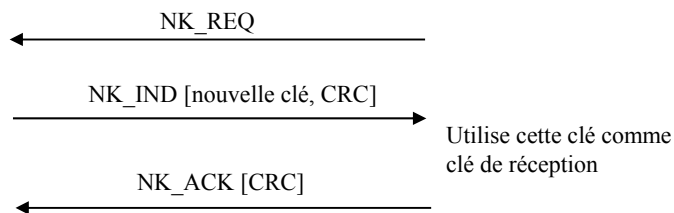
- une clé statique ;
- une clé dynamique d'émission ;
- une clé dynamique de réception.

Au départ, seule la clé statique de chaque interface est activée.

Négociation de clé :

Interface émettrice

Interface réceptrice



Utilise cette clé comme
clé d'émission

Utilise cette clé comme
clé de réception

Les deux derniers messages sont chiffrés par les clés statiques.

Algorithmes utilisés : IDEA, Blowfish (128 bits).

Recommandations : une clé dynamique ne doit pas servir plus de 15 minutes, et ne doit pas servir à chiffrer plus de 2^{32} paquets.

Si ces limites sont dépassées, il faut régénérer une nouvelle clé.

Le protocole est implémenté pour des systèmes Unix et Windows.

C'est un protocole très simple pour protéger les paquets IP.

Il n'est pas compatible avec IPSec.

1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Éléments de comparaison des VPN
6. Exemples d'implémentation

Introduction : SSL



SSL : Secure Socket Layer

Il s'agit d'un protocole à négociation, développé par Netscape.

<http://wp.netscape.com/eng/ssl3/draft302.txt>

Il s'applique entre la couche TCP et l'application.

Il ne fonctionne pas avec UDP.

SSL permet :

- d'authentifier mutuellement le serveur et le client
- d'assurer la confidentialité des communications
- d'assurer l'intégrité des données.

TLS v1.0 (Transport Layer Security protocol), développé par l'IETF,
correspond à la version 3.1 de SSL

RFC 2246

285

Introduction : SSL



SSL se base sur des sous-protocoles :

- SSL handshake (authentification mutuelle du serveur et du client, négociations des algorithmes, négociations des clés de session)
- SSL Change Cipher Spec
- SSL Alert (envoi de messages d'erreur - *warning* / *fatal*)
- SSL Record (confidentialité et intégrité des données)

286

Introduction : SSL



Il existe une implémentation OpenSource de SSL v2.0, SSL v3.0 et TLS v1.0 : SSLeay, devenue OpenSSL

<http://www.columbia.edu/~ariel/ssleay/>

<http://www.openssl.org>

Attention aux vulnérabilités !

<http://www.certa.ssi.gouv.fr/site/CERTA-2003-AVI-156/index.html.2.html>

30 septembre 2003

-> déni de service

-> exécution de code arbitraire à distance

287

Introduction : SSL



On peut sécuriser toutes les applications qui s'appuient sur la couche SSL / TLS (HTTPS, IMAPS, FTPS, ...).

Attention aux proxys ! => protocole SOCKS

Le protocole SSL n'impose pas une consultation systématique d'une CRL pour valider un certificat.

288

Introduction : SSL



Avantages :

- il n'y a plus de problème de NAT/NAPT
- les logiciels clients supportant ces protocoles sont de plus en plus répandus

Inconvénient :

- il faut que l'application supporte la librairie SSL
- le protocole est souvent implémenté de manière incomplète (pas d'authentification client par certificat pour IMAPS et SMTPS)

289

Introduction : SSL



Evolution : le protocole Stunnel

Principe : encapsuler des communications TCP dans une couche SSL

<http://www.stunnel.org>

290

Introduction : SSL

Protocole SSL classique



291

Introduction : SSL

Protocole Stunnel



292

Protocole Stunnel

Avantages : - protocoles transparents pour les applications
- permet d'accéder par SSL à des applications non SSL

Inconvénients : - nécessite un client Stunnel
- ne supporte l'encapsulation d'UDP nativement

SSH

SSH (Secure Shell)

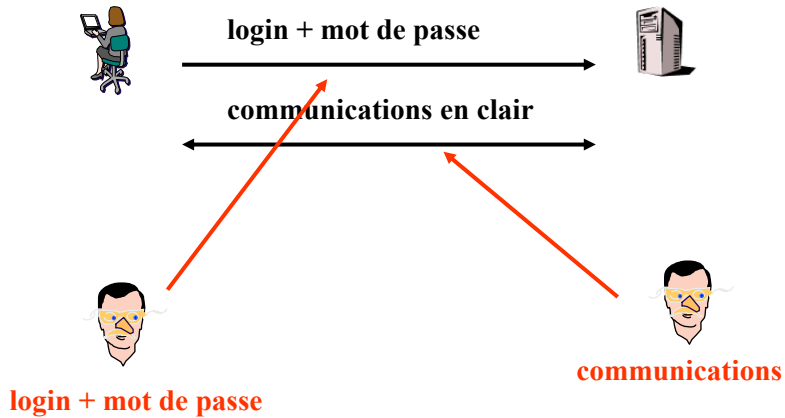
Niveau application

Protocole client / serveur

A l'origine, SSH est utilisé comme un service telnet sécurisé : le mot de passe ne circule pas en clair et les échanges sont chiffrés.

SSH

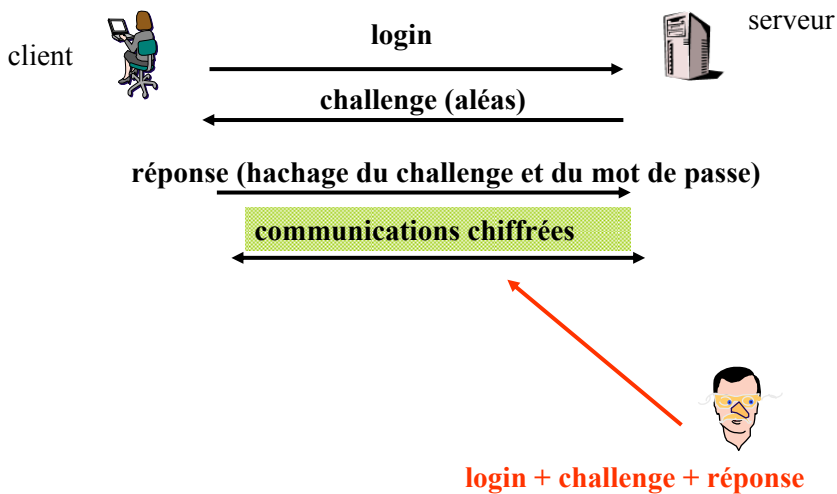
Telnet :



295

SSH

SSH :



296

Avantage de SSH :

- chiffrement de la session et mot de passe unique (*OTP*)
- le client SSH est disponible sur un très grand nombre de plateformes
- la sécurité est assurée depuis le client jusqu'au serveur (*end-to-end security*)

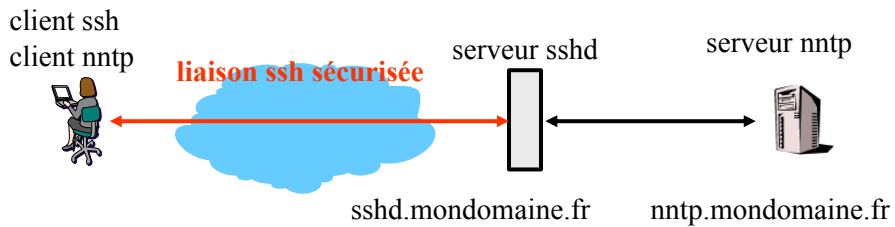
SSH : Port Forwarding

Le *Port Forwarding* (ou transfert de port) permet d'utiliser une liaison SSH pour transporter des protocoles non sécurisés (POP, NNTP, ...).

On peut alors construire un VPN basé sur des liaisons SSH.

SSH

SSH : Port Forwarding



299

SSH

SSH : Port Forwarding

Exemple de configuration sur un poste Linux :

```
$ ssh -L 4444:nntp.mondomaine.fr:119 sshd.mondomaine.fr
```

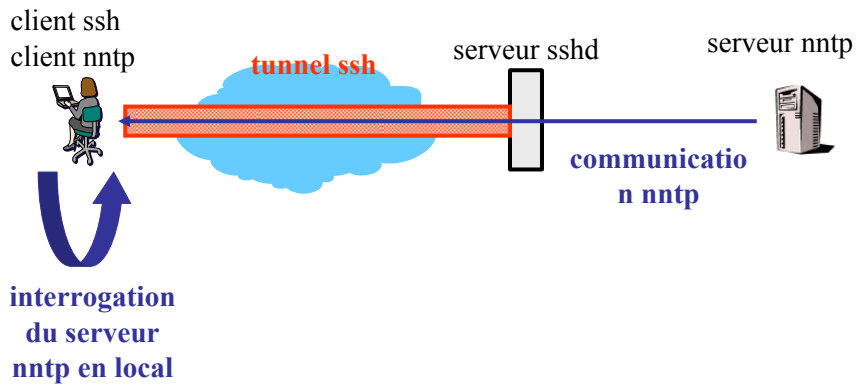
puis on configure le client nntp de la manière suivante :

- serveur : localhost
- port : 4444

300

SSH

SSH : Port Forwarding



301

SSH

SSH : Port Forwarding

Avantages : - on conserve les applications en « standard » (clients et serveurs)

Inconvénients : - gestion manuelle des clés

- le numéro de port utilisé par le protocole doit être fixe.

- tous les protocoles ne sont pas triviaux à implémenter
(ex : ftp/ftp-data)

302

Rappel du plan

1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Éléments de comparaison des VPN
6. Exemples d'implémentation

303

Éléments de comparaison

Les protocoles PPTP et L2TP :

- permettent de transporter tous les protocoles réseau ;
- sont bien adaptés au utilisateur itinérant ;
- sont implémentés par de nombreux éditeurs.

304

Eléments de comparaison



Le protocole IPSec :

- utilise des algorithmes puissants pour l'authentification et le chiffrement ;
- est difficile à implémenter pour des utilisateurs itinérants.

Le protocole CIPE :

- est facile à configurer et à utiliser ;
- est moins riche en fonctionnalités ;
- est incompatible avec IPSec.

305

Eléments de comparaison



Le protocole SSH :

- est très largement répandu sur tous les type de plates-formes ;
- un tunnel est établi pour chaque application ;
- le numéro de port doit être prévisible.

306

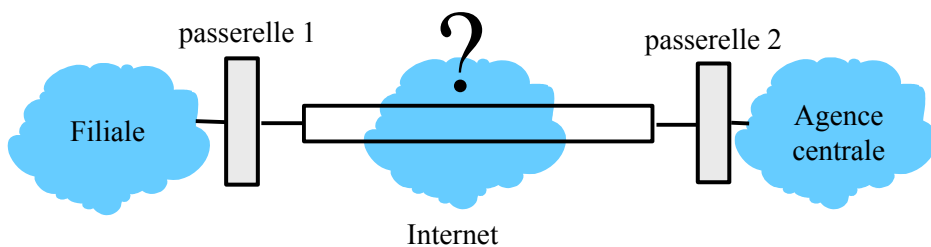
Rappel du plan

1. Présentation des VPN
2. Les VPN au niveau liaison de données
 - 2.1 Rappels
 - 2.2 PPTP
 - 2.3 L2F
 - 2.4 L2TP
3. Les VPN au niveau réseau
 - 3.1 IPSec
 - 3.2 CIPE
4. Les VPN au niveau application
5. Eléments de comparaison des VPN
6. Exemples d'implémentation

307

Exemples

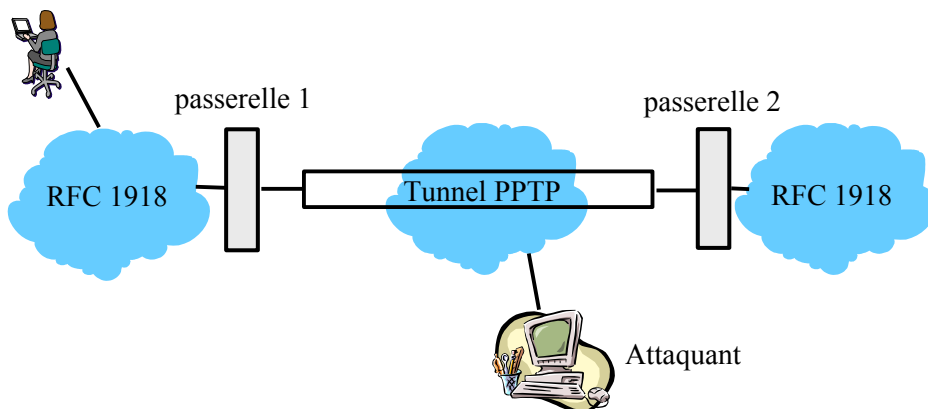
Contexte



308

Exemples

Première implémentation : PPTP



309

Exemples

1ère étape : se faire envoyer un mail de la part de la victime.

En-tête de mail :

```
Received:
from V1 (user123.branch12.company.com [192.168.1.2]) by mail.company.com
(8.9.3+Sun/8.9.3) with SMTP id QAA9544; Tue, 24 Oct 2001 16:33:30 + 0200
(MEST)
....
X-Mailer: KMail [version 1.0.29.2]
```

310

Exemples



2ème étape : récupérer la configuration du routeur de la victime.

Par exemple :

```
interface Tunnel0
ip address 10.1.2.3 255.255.255.252
tunnel source Serial0
tunnel destination 2.1.1.1
```

où Serial0 est l'interface WAN, avec comme adresse IP 2.1.1.2

311

Exemples

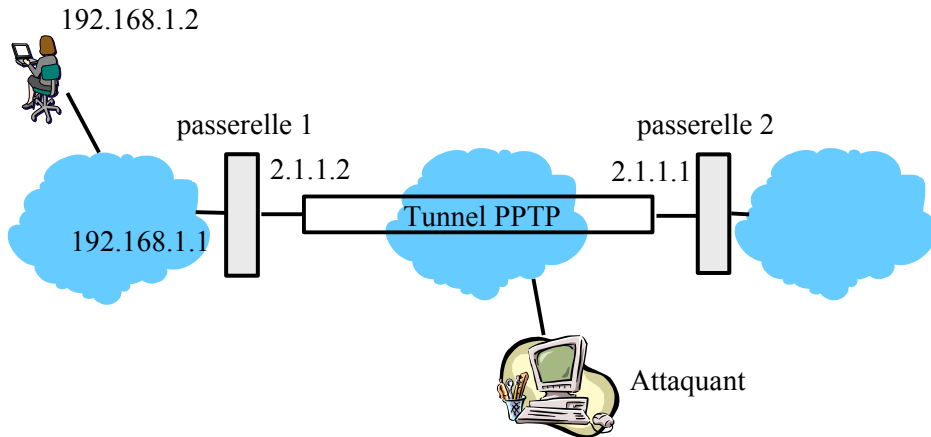


Que connaissons-nous à présent ?

- l'adresse IP de la victime : 192.168.1.2
 - les adresses IP source (filiale) et destination du tunnel (agence centrale)
- : 2.1.1.1 et 2.1.1.2

312

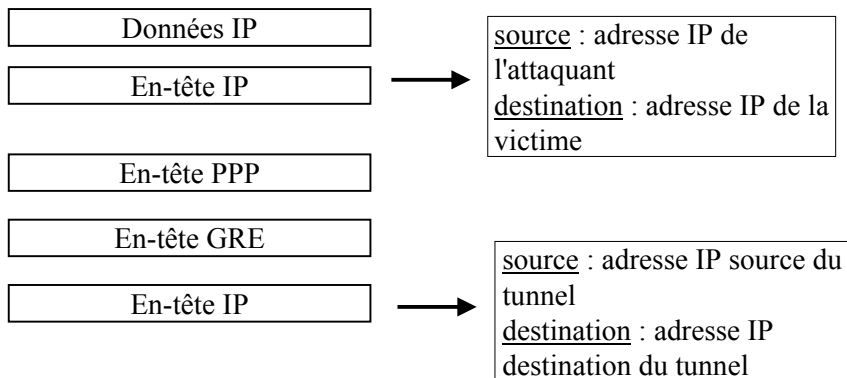
Exemples



313

Exemples

Pour initier l'attaque, on commence par forger un paquet PPTP grâce à GRE :



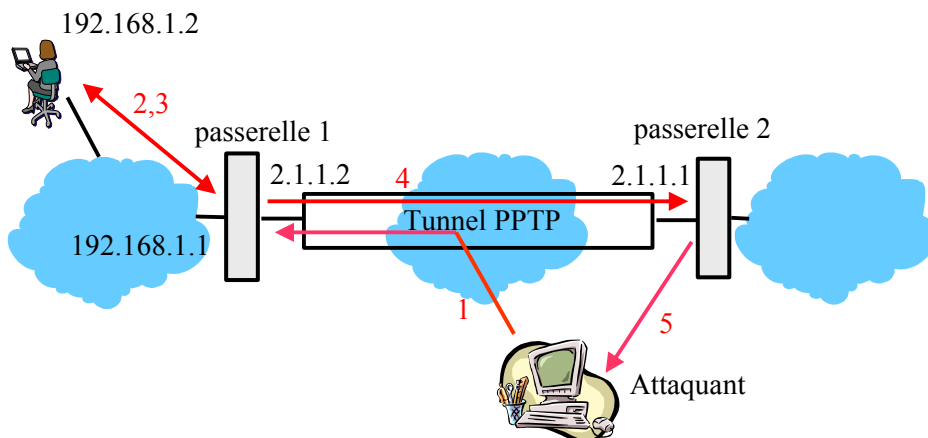
314

Exemples

- Lorsque le routeur de la victime reçoit le paquet, il vérifie sa provenance, et décapsule le paquet IP.
- Le paquet IP est envoyé à la victime à travers le réseau interne.
- La victime renvoie le paquet d'après sa table de routage (ici, la passerelle par défaut est le routeur du tunnel).
- Le routeur de la filiale n'a qu'une route par défaut, et encapsule le paquet dans une trame PPTP, et l'envoie par le tunnel.
- Le routeur de l'agence centrale possède un accès à l'Internet, et peut donc faire suivre le paquet en le décapsulant.

315

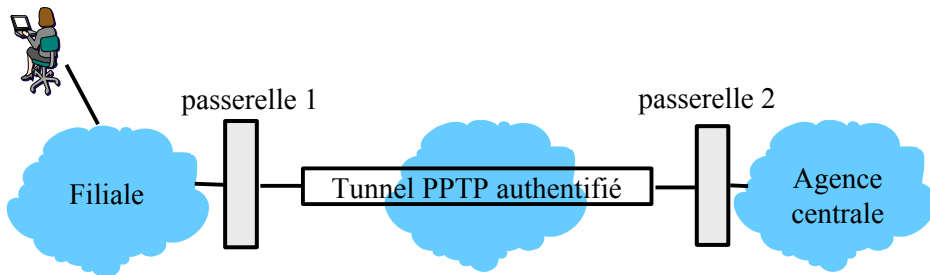
Exemples



316

Exemples

Deuxième implémentation : PPTP avec authentification



317

Exemples

Authentification dans PPTP :

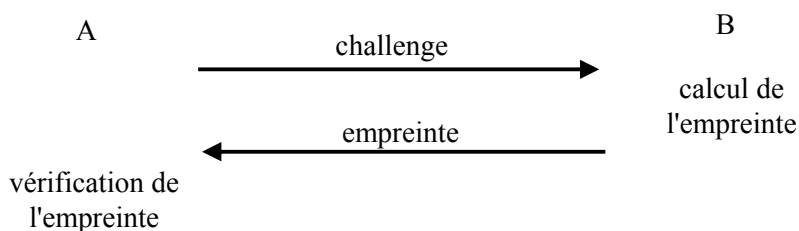
- génération d'une clé de 32 bits, incluse dans l'en-tête
- utilisation de l'algorithme CHAP (Challenge Handshake Authentication Protocol)

318

Exemples

Protocole CHAP :

- génération et envoi d'un “challenge” (ou épreuve)
- calcul et envoi d'une empreinte par une fonction de hachage
- vérification du calcul de l'empreinte



319

Exemples

Implémentation de Microsoft MS-CHAP :

- le serveur envoie une épreuve de 8 octets aléatoires au client.
- le client utilise le hachage LAN Manager de son mot de passe pour en dériver 3 clés DES. Chacune de ces clés sert à chiffrer l'épreuve. Les trois blocs sont concaténés dans une réponse de 24 octets. Le client crée alors, avec la même procédure, une seconde réponse de 24 octets en utilisant le hachage Windows NT.
- le serveur utilise les hachages du mot de passe client pour déchiffrer ses réponses.

320

Exemples

Deuxième version du protocole : MS-CHAPv2

Cette version supprime la faiblesse principale de MS-CHAPv1 : la transmission de deux hachages différents d'un même mot de passe.

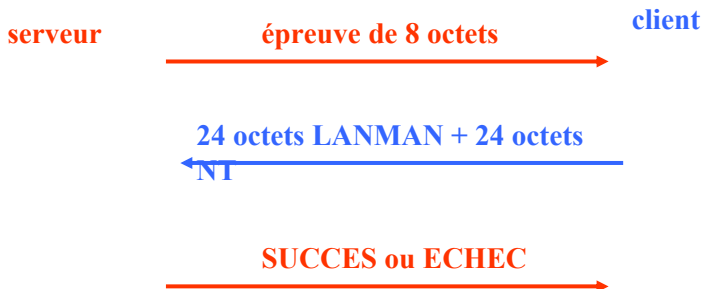
De plus, certaines améliorations ont été faites :

- système d'authentification du serveur ;
- plus robuste contre les attaques en déni de service ;
- ne laisse plus filtrer d'information au sujet des sessions VPN actives.

321

Exemples

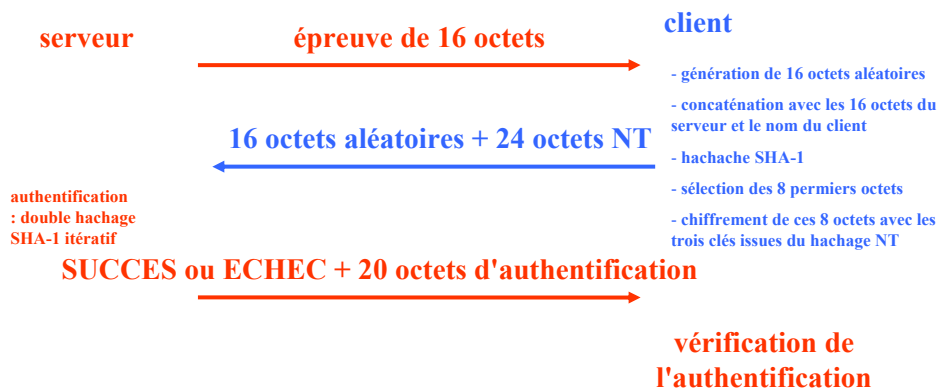
Version 1 du protocole (algorithme 0x80) :



322

Exemples

Version 2 du protocole (algorithme 0x81) :



323

Exemples

Malgré cela, la sécurité de ce protocole reste basée sur le mot de passe utilisé, et peut être contournée par des attaques par dictionnaire.

Exemple de méthodes utilisées par les dictionnaires :

- mots avec des chiffres ;
- casse modifiée ;
- mots inversés ;
- acronymes ;
- mots avec ponctuation.

324

Exemples

Et le chiffrement ?

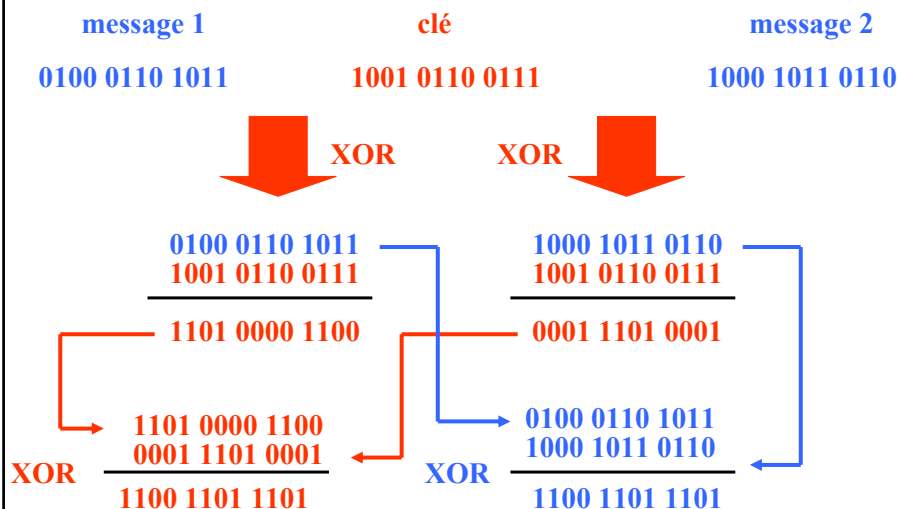
L'implémentation de Microsoft des protocoles PPTP propose le protocole MPPE (Microsoft Point-to-Point Encryption) pour chiffrer le tunnel.

Dans la première version, la même clé était utilisée pour les deux sens de la communication.

Or l'algorithme utilisé est le chiffrement par flux RC4 (XOR)

325

Exemples



326

Exemples

Version 2 :

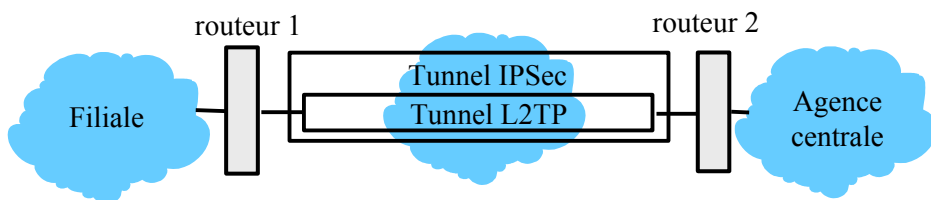
Dans le protocole MS-CHAPv2, deux clés différentes sont utilisées, une pour chaque direction.

Mais ces clés sont également basées sur les mots de passe des utilisateurs, qui ont, en général, une entropie trop faible pour garantir un niveau de sécurité satisfaisant.

327

Exemples

Troisième implémentation : L2TP avec protection IPSec



328

Exemples



Les trames L2TP circulant entre les deux routeurs sont protégées par un canal IPSec.

L2TP est un protocole répandu, développé par Microsoft et Cisco, mais ne propose pas de mécanisme de protection.

En revanche, IPSec propose des mécanismes fiables d'authentification (AH) et de chiffrement (ESP).

329

Exemples



Attaques sur le protocole L2TP :

- découverte de l'identité des utilisateurs en écoutant le trafic
- modification des paquets (données et contrôle)
- interception de sessions PPP ou L2TP (*session hijacking*)
- déni de service (terminaison forcée des connexions PPP ou des tunnels L2TP)

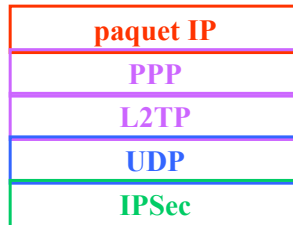
330

Exemples

On utilise IPSec pour sécuriser le trafic L2TP

RFC 2888 : *Secure Remote Access with L2TP*

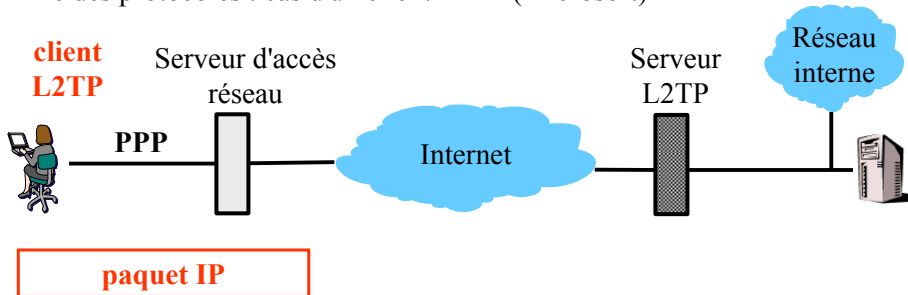
RFC 3193 : *Securing L2TP using IPSec* (Nov. 2001)



331

Exemples

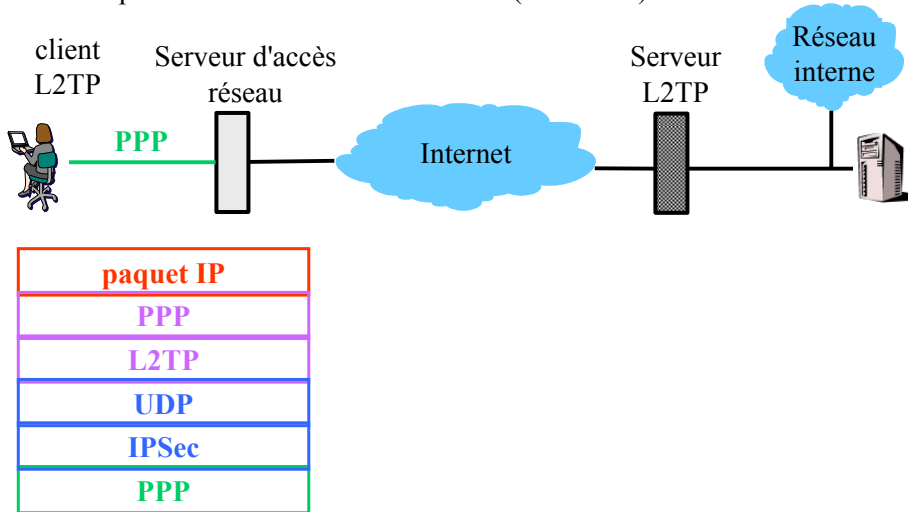
Pile des protocoles : cas d'un client L2TP (Microsoft)



332

Exemples

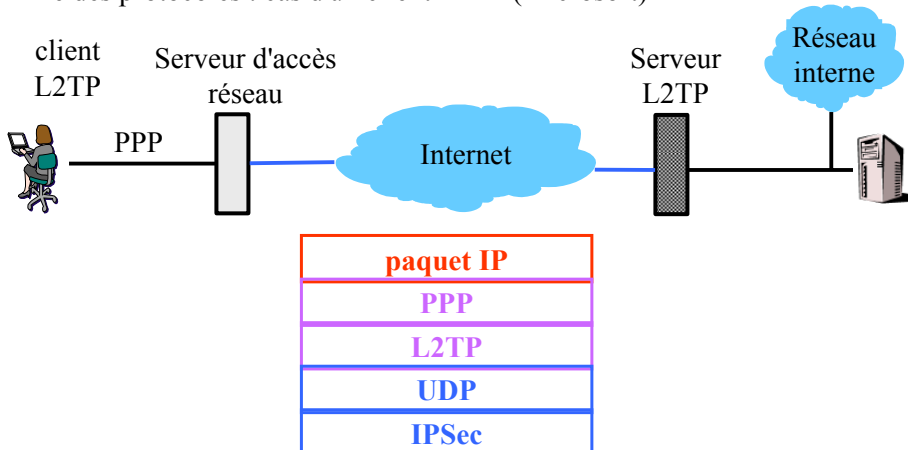
Pile des protocoles : cas d'un client L2TP (Microsoft)



333

Exemples

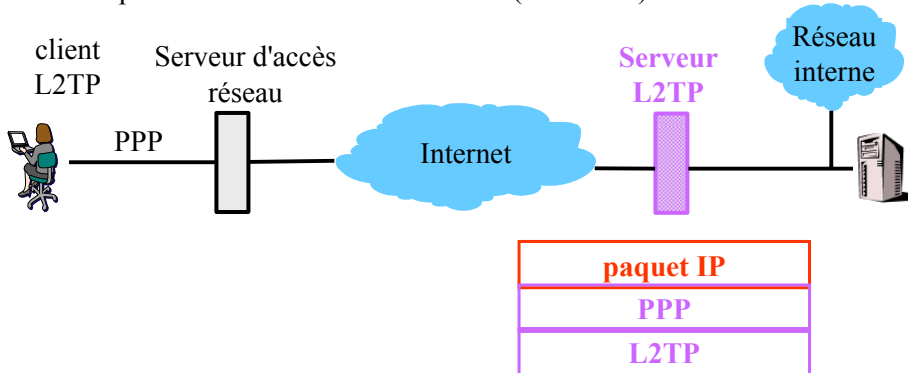
Pile des protocoles : cas d'un client L2TP (Microsoft)



334

Exemples

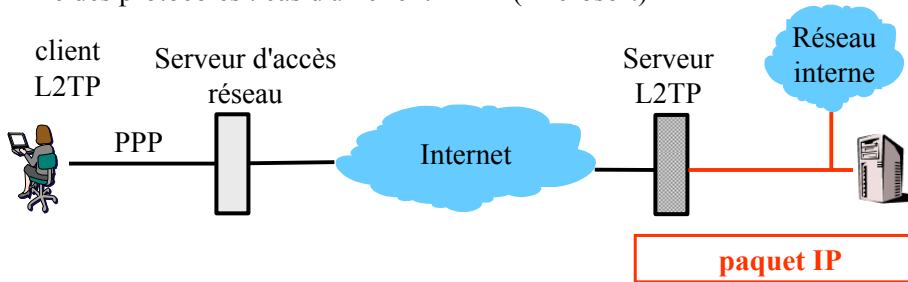
Pile des protocoles : cas d'un client L2TP (Microsoft)



335

Exemples

Pile des protocoles : cas d'un client L2TP (Microsoft)



336

Conclusion



La technologie des VPN est une technologie mature.

Des solutions existent pour assurer la sécurité des données.

La qualité de service est le prochain enjeu des VPN, mais se heurte encore à de nombreux problèmes (nature du réseau sous-jacent, multiples ISP, ...).

337

Bibliographie



Toutes les RFC sont consultables sur le site de l'IETF :

<http://www.ietf.org>

- RFC 1331 : The Point-to-point Protocol (*W. Simpson*)
- RFC 1332 : The PPP Internet Protocol Control Protocol (*G. McGregor*)
- RFC 2784 : Generic Routing Encapsulation (*D. Farinacci, T. Li, S. Hanks, D. Meyer & P. Traina*)
- RFC 2637 : Point-to-Point Tunneling Protocol (*K. hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little & G. Zorn*)
- RFC 2341 : Cisco Layer Two Forwarding Protocol (*A. Valencia, M. Littlewood & T. Kolar*)

338

Bibliographie



- RFC 2661 : Layer Two Tunneling Protocol (*W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn & B. Palter*)
- RFC 2888 : Secure Remote Access with L2TP (*P. Srisureh*)
- RFC 3193 : Securing L2TP using IPSec (*B. Patel, B. Aboda, W. Dixon, G. Zorn & S. Booth*)
- Documentation Microsoft : “PPTP Frequently asked Questions”
<http://www.microsoft.com/NTServer/ProductInfo/faqs/PPTPfaq.asp>
- Cryptanalyse de Microsoft PPTP (*B. Schneier & Mudge*)
<http://www.counterpane.com/pptpv2-paper.html>

339

Bibliographie



- CIPE (*Olaf Titz*)
<http://sites.inka.de/sites/bigred/devel/cipe.html>
- Test d'interopérabilité (*Ghislaine Labouret - HSC*)
<http://www.hsc.fr/ipsec/ipsec2001/>

340

KUROSE STUFF AFTER

Plan

- 1 Qu'est-ce que la sécurité ?
- 2 Principes de cryptographie
- 3 Authentification
- 4 Intégrité
- 5 Distribution de clés et certification
- 6 contrôle d'accès : firewalls
- 7 Attaques and parades
- 8 Sécurité dans plusieurs couches

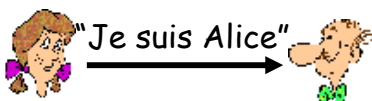
Plan

- 1 Qu'est-ce que la sécurité ?
- 2 Principes de cryptographie
- 3 Authentification
- 4 Intégrité
- 5 Distribution de clés et certification
- 6 contrôle d'accès : firewalls
- 7 Attaques and parades
- 8 Sécurité dans plusieurs couches

Authentification

But : Bob veut qu'Alice lui “prouve” son identité

Protocole ap1.0: Alice dit “Je suis Alice”



Scénario d'échec ?



Authentification

But : Bob veut qu'Alice lui "prouve" son identité

Protocole ap1.0: Alice dit "Je suis Alice"

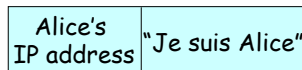


"Je suis Alice"

Dans un réseau,
Bob ne peut pas "voir"
Alice, alors Trudy dit
"Je suis Alice" simplement qu'elle est
Alice

Authentification : autre tentative

Protocole ap2.0 : Alice dit "Je suis Alice" dans un paquet IP
contenant son adresse IP source

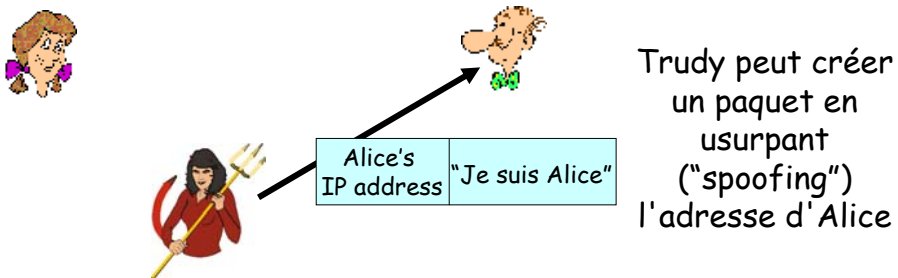


Scénario d'échec ?



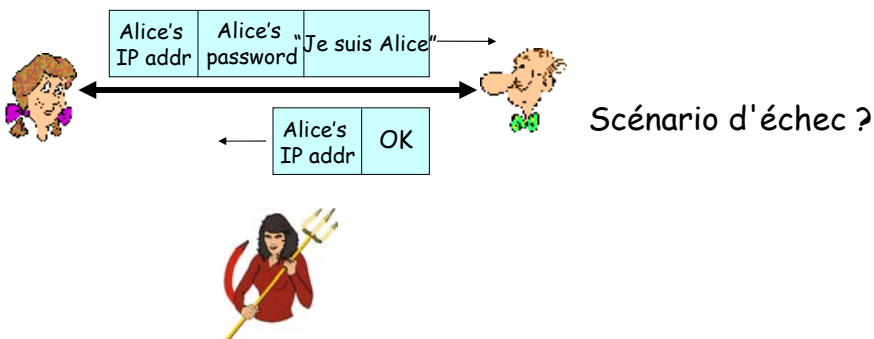
Authentification : autre tentative

Protocole ap2.0 : Alice dit "Je suis Alice" dans un paquet IP contenant son adresse IP source



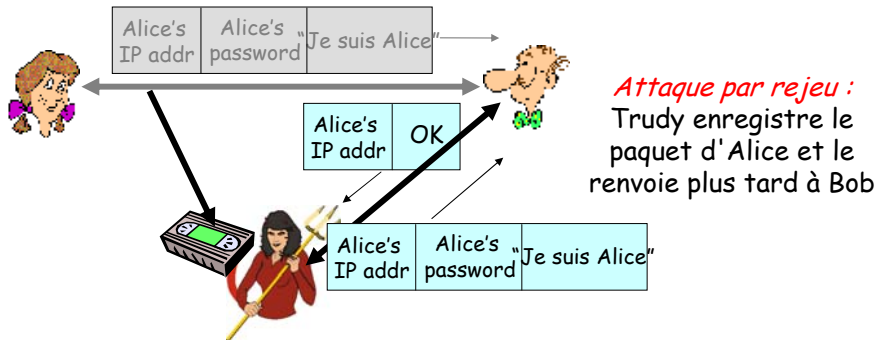
Authentification : autre tentative

Protocole ap3.0 : Alice dit "Je suis Alice" et envoie son mot de passe secret pour le prouver.



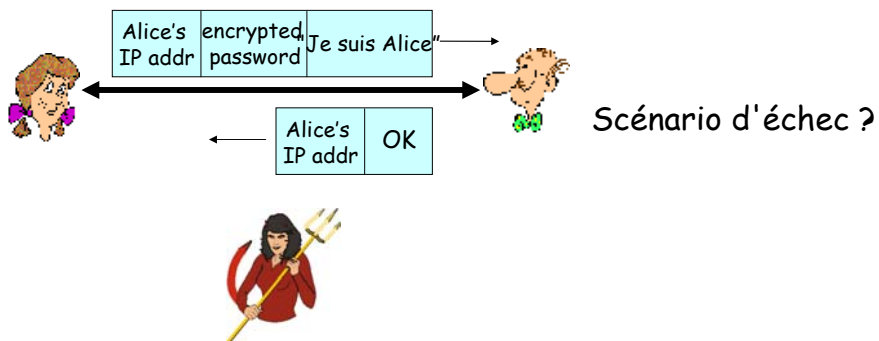
Authentication: autre tentative

Protocole ap3.0 : Alice dit "Je suis Alice" et envoie son mot de passe secret pour le prouver.



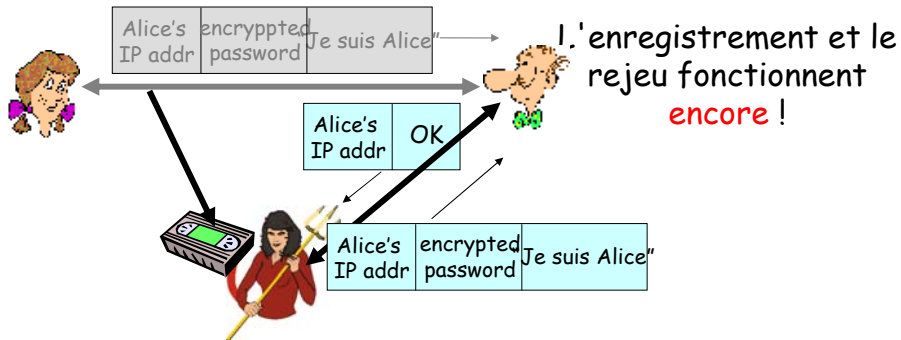
Authentication : encore un autre essai

Protocole ap3.1 : Alice dit "Je suis Alice" et envoie son mot de passe secret *chiffré* pour le prouver.



Authentification : encore un autre essai

Protocole ap3.1 : Alice dit "Je suis Alice" et envoie son mot de passe secret *chiffré* pour le prouver.

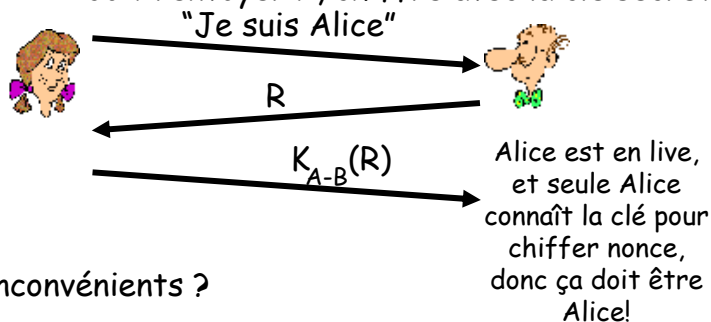


Authentification : encore un autre essai

But : éviter l'attaque du rejeu

Nonce: nombre (R) utilisé *seulement-une-fois*

ap4.0: pour prouver qu' Alice est "en live", Bob envoie à Alice *nonce*, R. Alice doit renvoyer R, chiffré avec la clé secrète "Je suis Alice"



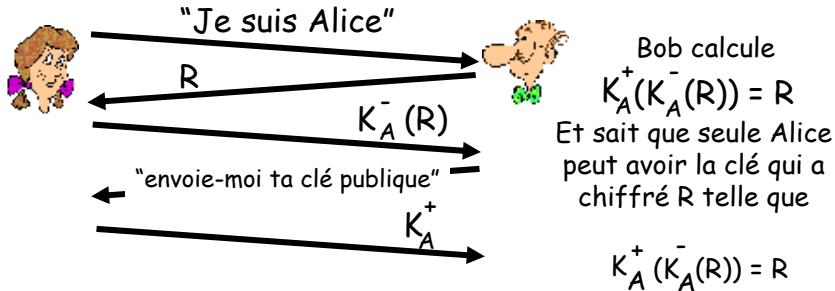
Échecs, inconvénients ?

Authentication : ap5.0

ap4.0 nécessite le partage d'une clé symétrique

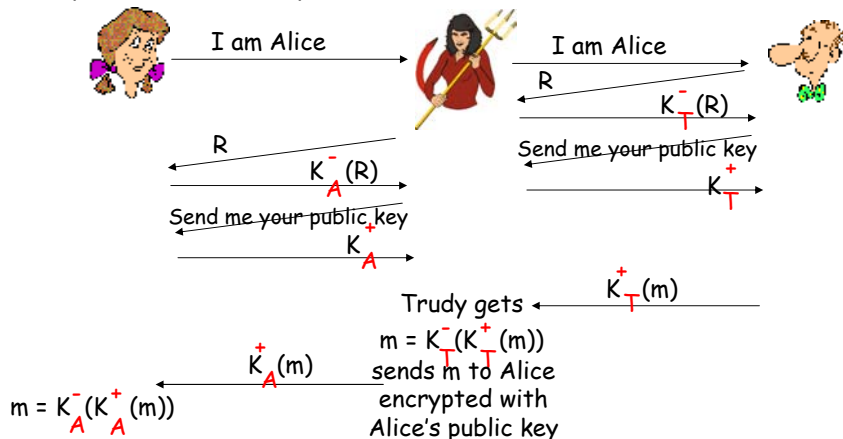
- Peut-on authentifier en utilisant des techniques de clé publique ?

ap5.0: utiliser nonce, crypto à clé publique



ap5.0 : trou de sécurité

Attaque d'une personne au milieu : Trudy se fait passer pour Alice (vis-à-vis de Bob) et pour Bob (vis-à-vis d'Alice)



ap5.0 : trou de sécurité

Attaque d'une personne au milieu : Trudy se fait passer pour Alice (vis-à-vis de Bob) et pour Bob (vis-à-vis d'Alice)



Difficile à détecter :

- ☐ Bob reçoit tout ce qu'Alice envoie et vice versa.
- ☐ le problème est que Trudy reçoit également tous les messages !

Plan

- 1 Qu'est-ce que la sécurité ?
- 2 Principes de cryptographie
- 3 Authentification
- 4 **Intégrité**
- 5 Distribution de clés et certification
- 6 contrôle d'accès : firewalls
- 7 Attaques and parades
- 8 Sécurité dans plusieurs couches

Signatures numériques

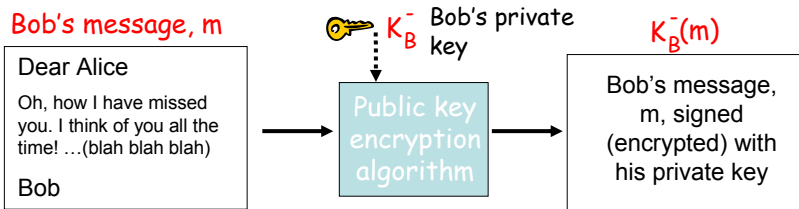
Technique cryptographique analogue aux signatures manuscrites.

- L'émetteur (Bob) signe le document de manière numérique et établit qu'il est le créateur/propriétaire du document.
- **Vérifiable, non falsifiable**: le récepteur (Alice) peut prouver à quelqu'un que Bob et personne d'autre (y compris Alice) a signé ce document

Signatures numériques

Signature numérique simple pour le message m :

- Bob signe m en chiffrant avec sa clé privée K_B , créant le message "signé" $K_B^-(m)$



Signatures numériques (suite)

- Supposons qu'Alice reçoit le msg m et la signature numérique $K_B(m)$
- Alice vérifie que m a été signé par Bob en appliquant la clé publique de Bob K_B à $K_B(m)$ et vérifie que $K_B(K_B(m)) = m$.
- Si $K_B(K_B(m)) = m$, la personne qui a signé m a forcément utilisé la clé privée de Bob.

Alice vérifie ainsi que :

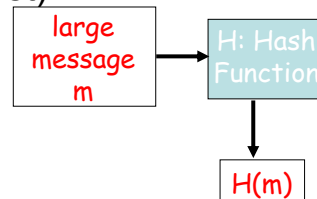
- ➔ Bob a signé m .
- ➔ Personne d'autre n'a signé m .
- ➔ Bob a signé m et pas m' .

Non-répudiation:

- ✓ Alice peut emporter m et la signature $K_B(m)$ à un procès et prouver que Bob a signé m .

Condensats (message digest)

Le chiffrement par clé publique de longs messages est très onéreux "computationnellement"



But : "empreinte digitale" de longueur fixe et facile à calculer

- Appliquer une fonction de hachage H à m , recevoir un message condensé de longueur fixe, $H(m)$.

Propriétés de la fonction de hachage:

- many-to-1
- Produit des messages condensés de taille fixe ("empreinte digitale")
- Étant donné un message condensé x , il est computationnellement impossible de trouver m tel que $x = H(m)$

Internet checksum : fonction cryptographique de hachage "pauvre"

Le Internet checksum possède des propriétés de fonction de hachage :

- Produit un message condensé de longueur fixe du message (somme sur 16-bits)
- many-to-one

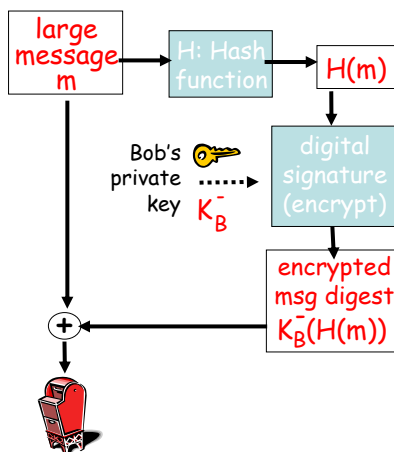
Mais avec un message avec une valeur de hachage donnée, il est facile de trouver un autre message avec la même valeur de hachage :

message	ASCII format	message	ASCII format
I O U 1	49 4F 55 31	I O U <u>9</u>	49 4F 55 <u>39</u>
0 0 . 9	30 30 2E 39	0 0 . <u>1</u>	30 30 2E <u>31</u>
9 B O B	39 42 D2 42	9 B O B	39 42 D2 42
B2 C1 D2 AC		B2 C1 D2 AC	

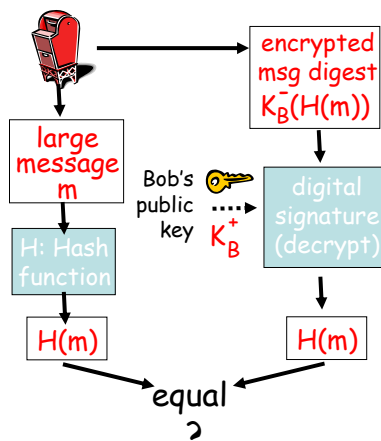
différents messages
Mais checksums identiques!

Signature numérique = Condensat signé

Bob envoie un message signé numériquement :



Alice vérifie la signature et l'intégrité du message signé numériquement :



Algorithmes de fonctions de hachage

- **Fonction de hachage MD5 largement utilisée (RFC 1321)**
 - Calcule un condensat de 128 bits en 4 étapes.
 - Il est difficile, à partir d'une chaîne aléatoire de 128 bits, de construire un msg m dont l hash MD5 est égal à x.
- **SHA-1 est également utilisé.**
 - Standard américain [NIST, FIPS PUB 180-1]
 - Message condensé de 160 bits

Plan

- 1 Qu'est-ce que la sécurité ?
- 2 Principes de cryptographie
- 3 Authentification
- 4 Intégrité
- 5 Distribution de clés et certification
- 6 contrôle d'accès : firewalls
- 7 Attaques and parades
- 8 Sécurité dans plusieurs couches

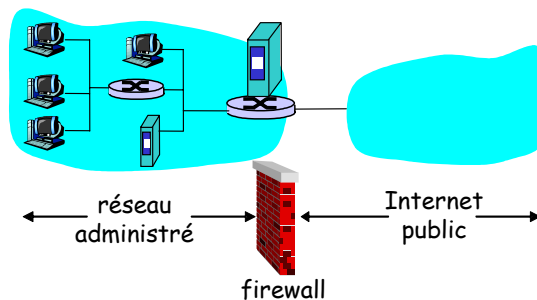
Plan

- 1 Qu'est-ce que la sécurité ?
- 2 Principes de cryptographie
- 3 Authentification
- 4 Intégrité
- 5 Distribution de clés et certification
- 6 **contrôle d'accès : firewalls**
- 7 Attaques and parades
- 8 Sécurité dans plusieurs couches

Firewalls

firewall

Isole le réseau interne d'une organisation de l'Internet, en permettant à certains paquets de passer et en en bloquant d'autres.



Firewalls : pourquoi ?

Évite les attaques par déni de service :

- SYN flooding : l'attaquant établit de nombreuses connexions TCP "bidon", plus de ressources pour les vraies connexions.

Évite la modification / l'accès illégal aux données internes.

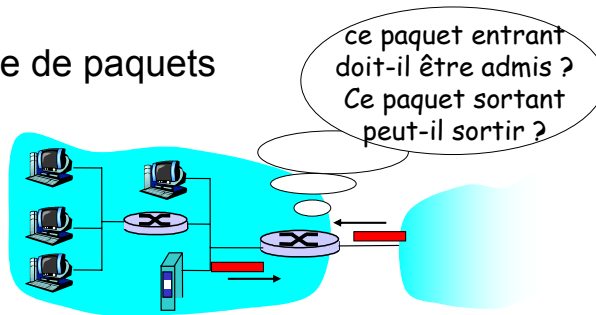
- Ex : l'attaquant remplace la page du CIA par autre chose

Autorise uniquement les accès autorisés à l'intérieur du réseau (ensembles d'utilisateurs / hôtes authentifiés)

2 types de firewalls :

- Niveau applicatif
- Filtrage de paquets

Filtrage de paquets



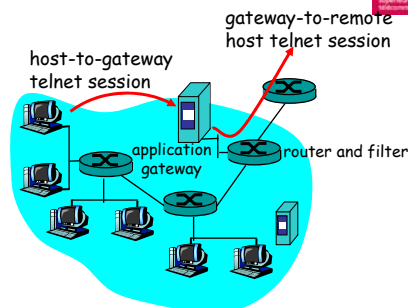
- Réseau interne connecté à l'Internet via **un routeur firewall**
- Le routeur **filtre paquet par paquet**, décision de faire transiter ou de supprimer le paquet selon :
 - L'adresse IP source, l'adresse IP destination
 - Les numéros de ports TCP/UDP source et destination
 - Le type de message ICMP
 - Les bits TCP SYN et ACK

Filtrage de paquets

- Exemple 1 : blocage des datagrammes IP entrants et sortants avec le champ protocole = 17 et avec le port source ou destination = 23.
 - Tous les flux UDP entrants et sortants, ainsi que les connexions telnet, sont bloqués.
- Exemple 2 : Blocage des segments TCP inbound avec ACK=0.
 - Empêche les clients extérieurs de faire des connexions TCP avec des clients internes, mais permet aux clients internes de se connecter à l'extérieur.

Gateway applicative

- Filtre les paquets en fonction des données applicatives aussi bien qu'en fonction des champs IP/TCP/UDP.
- Exemple : permettre à des utilisateurs internes autorisés d'effectuer un telnet à l'extérieur.



1. Nécessite que tous les utilisateurs de telnet passent par la gateway.
2. Pour des utilisateurs autorisés, la gateway établit une connexion telnet à l'hôte de destination. La gateway fait transiter les données entre les 2 connexions.
3. Le filtre du routeur bloque toutes les connexions telnet ne provenant pas de la gateway.

Limites des firewalls et des gateways

- IP spoofing : le routeur ne peut pas savoir si les données proviennent vraiment d'une source autorisée
- Si plusieurs applications ont besoin d'un traitement spécial, chacune a sa propre gateway applicative.
- Le logiciel client doit savoir comment contacter la gateway.
 - Ex : il doit configurer l'adresse IP du proxy dans le browser Web
- Les filtres utilisent souvent une politique tout ou rien pour UDP.
- Compromis : **degré de communication avec le monde extérieur, niveau de sécurité**
- De nombreux sites hautement protégés souffrent toujours d'attaques.

Plan

- 1 Qu'est-ce que la sécurité ?
- 2 Principes de cryptographie
- 3 Authentification
- 4 Intégrité
- 5 Distribution de clés et certification
- 6 contrôle d'accès : firewalls
- 7 **Attaques and parades**
- 8 Sécurité dans plusieurs couches

Menaces de sécurité sur Internet

Mapping :

- Avant d'attaquer : trouver quels services sont implémentés sur le réseau
- Utiliser `ping` pour déterminer quels hôtes ont des adresses sur le réseau
- Scan des ports : essayer d'établir des connexions TCP avec chaque port (regarder ce qui se passe)
- nmap (<http://www.insecure.org/nmap/>) mapper
 - “network exploration and security auditing”

Parades ?

Menaces de sécurité sur Internet

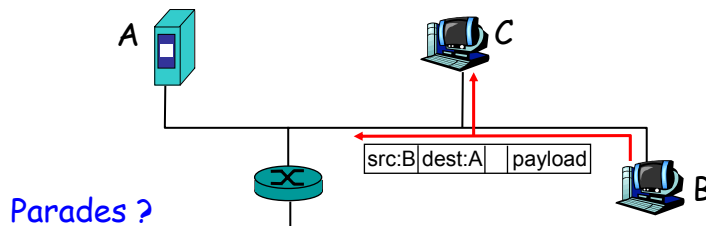
Mapping : parades

- Enregistrer le trafic pénétrant dans le réseau
- Chercher une activité suspecte (adresses IP, ports scannés les uns après les autres)

Menaces de sécurité sur Internet

Reniflement de paquets (packet sniffing) :

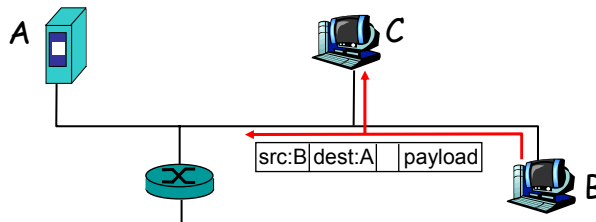
- Médium à diffusion
- Une machine proche lit tous les paquets qui passent
- peut lire toutes les données en clair (par ex les mots de passe)
- Ex : C sniffe les paquets de B



Menaces de sécurité sur Internet

Packet sniffing : parades

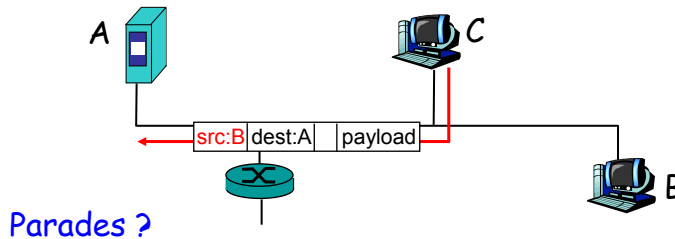
- Tous les hôtes d'une organisation utilisent des logiciels vérifiant périodiquement si l'interface de l'hôte est en mode promiscuous.
- Un hôte par segment du médium à diffusion (switched Ethernet at hub)



Menaces de sécurité sur Internet

IP Spoofing (usurpation d'adresse IP) :

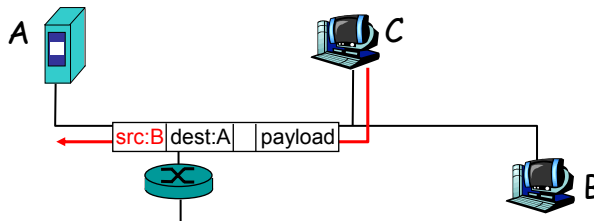
- Peut générer des paquets IP directement à partir d'une application, en mettant n'importe quelle valeur dans le champ d'adresse IP source
- Le récepteur ne peut pas dire si la source est spoofée
- Ex : C se fait passer pour B



Menaces de sécurité sur Internet

IP Spoofing : ingress filtering

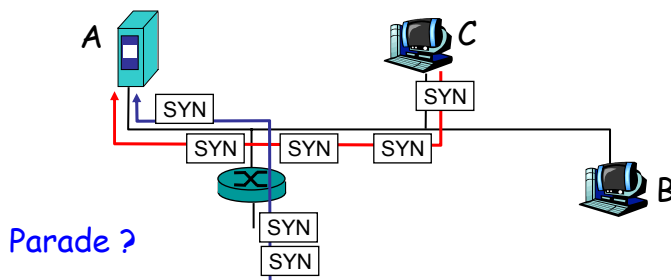
- Les routeurs ne doivent pas transmettre des paquets sortants avec des adresses source invalides (ex : adresse source du datagramme pas dans le réseau du routeur)
- Bien, mais ce filtrage ingress ne peut pas être effectué dans tous les réseaux



Menaces de sécurité sur Internet

Déni de Service (DOS) :

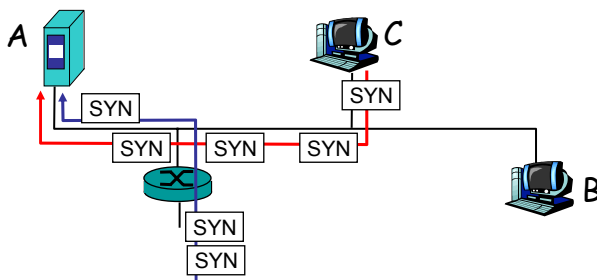
- Flot de paquets malicieux générés pour inonder un récepteur
- DOS distribué (DDOS): plusieurs sources coordonnées pour inonder un récepteur
- Ex : C et un hôte distant font une SYN-attack vers A



Menaces de sécurité sur Internet

Déni de service (DOS) : parades

- **Filtrer** le flot de paquets (ex : SYN) avant qu'ils n'atteignent l'hôte : on jette les bons comme les mauvais
- **Remonter** à la source des flots (probablement une machine innocente, compromise)

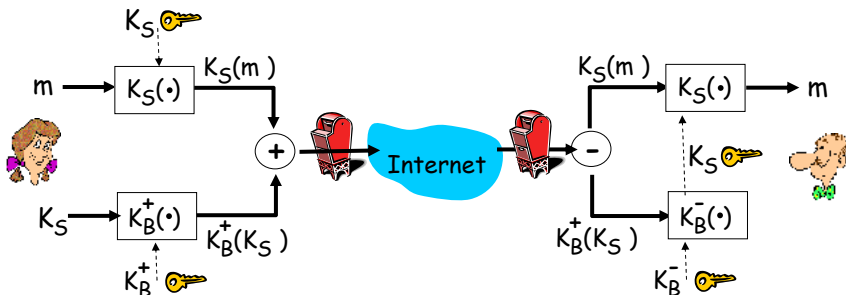


Plan

- 1 Qu'est-ce que la sécurité des réseaux ?
- 2 Principes de cryptographie
- 3 Authentification
- 4 Intégrité
- 5 Distribution de clés et certification
- 6 Contrôle d'accès : firewalls
- 7 attaques et parades
- 8 Sécurité dans plusieurs couches
 - 8.1. Email sécurisé
 - 8.2. Sockets sécurisées
 - 8.3. IPsec
 - 8.4. 802.11 WEP

E-mail sécurisé

- Alice veut envoyer un email sécurisé m à Bob.

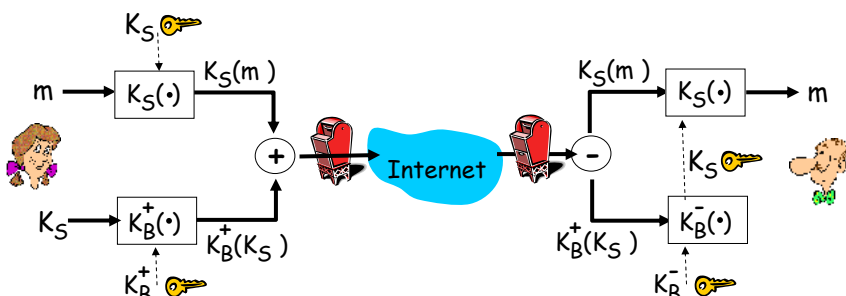


Alice :

- génère une clé privée *symétrique* aléatoire K_S .
- chiffre le message avec K_S (pour l'efficacité)
- chiffre aussi K_S avec la clé publique de Bob.
- envoie à la fois $K_S(m)$ et $K_B(K_S)$ à Bob.

E-mail sécurisé

- Alice veut envoyer un email sécurisé m à Bob.

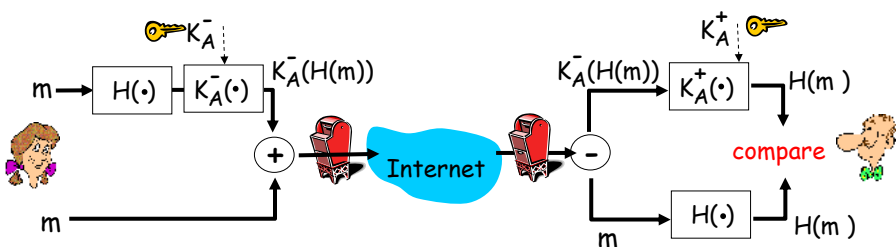


Bob :

- utilise sa clé privée pour déchiffrer et retrouver K_S
- utilise K_S pour déchiffrer $K_S(m)$ pour retrouver m

E-mail sécurisé (suite)

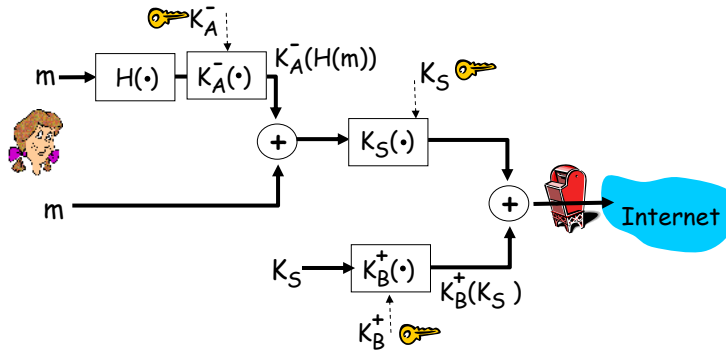
- Alice veut fournir une authentification de l'émetteur et l'intégrité du message.



- Alice signe numériquement le message.
- envoie à la fois le message (en clair) et la signature numérique.

E-mail sécurisé (suite)

- Alice veut fournir le secret, l'authentification de l'émetteur et l'intégrité du message.



Alice utilise 3 clés : sa clé privée, la clé publique de bob et la nouvelle clé symétrique créée

Pretty good privacy (PGP)

- Procédé de chiffrement d'email Internet, standard de-facto.
- Utilise la crypto à clé symétrique, la crypto à clé publique, une fonction de hachage et la signature numérique.
- Garantit le secret, l'authentification de l'émetteur, l'intégrité.
- L'inventeur, Phil Zimmerman, a été la cible de 3 ans d'enquête fédérale

Message signé par PGP :

```
---BEGIN PGP SIGNED MESSAGE---
Hash: SHA1

Bob:My husband is out of town
tonight.Passionately yours,
Alice

---BEGIN PGP SIGNATURE---
Version: PGP 5.0
Charset: noconv
yhHJRhhGJGhgq/12EpJ+lo8gE4vB3mqJ
hFEvZP9t6n7G6m5Gw2
---END PGP SIGNATURE---
```

Secure sockets layer (SSL)

- Sécurité de niveau transport pour toute application basée sur TCP utilisant des services SSL.
- Utilisé entre des browsers Web, des serveurs pour le commerce électronique (shttp).
- Services de sécurité :
 - Authentification du serveur
 - Chiffrement des données
 - authentification du client (optionnel)
- Authentification du serveur :
 - Le browser SSL-capable contient les clés publiques pour les CAs de confiance.
 - Le browser demande le certificat du serveur, fourni par un CA de confiance.
 - Le browser utilise la clé publique du CA pour extraire la clé publique du serveur du certificat.
 - Vérifier le menu de sécurité de votre browser pour voir ses CAs de confiance.

SSL (suite)

Session chiffrée avec SSL :

- Le browser génère une *clé de session symétrique*, la chiffre avec la clé publique du serveur et envoie la clé chiffrée au serveur.
- En utilisant la clé privée, le serveur déchiffre la clé de session.
- Le browser et le serveur connaissent la clé de session
 - Toutes les données envoyées dans la socket TCP (par le client ou par le serveur) sont chiffrées avec la clé de session.
- SSL : base du Transport Layer Security (TLS) de l'IETF.
- SSL peut être utilisé pour des applications non-Web, ex : IMAP.
- L'authentification du client peut être effectuée avec des certificats de client.

IPsec : Sécurité de niveau réseau

- **Secret de niveau réseau Network-layer secrecy:**
 - L'hôte émetteur chiffre les données dans le datagramme IP
 - Segments TCP et UDP; messages ICMP et SNMP.
- **Authentification de niveau réseau**
 - L'hôte de destination peut authentifier l'adresse IP source
- **2 protocoles de principe :**
 - Protocole d'authentification d'en-tête (authentication header AH)
 - Protocole encapsulation security payload (ESP)
- **Pour AH et ESP, handshake de la source et de la destination :**
 - Crée un canal logique de niveau réseau, appelé association de sécurité (SA)
- **Chaque SA est unidirectionnel.**
- **Déterminé de manière unique par :**
 - Un protocole de sécurité (AH or ESP)
 - Adresse IP source
 - Identifiant de connexion sur 32 bits

Protocole Authentication Header (AH)

- Fournit l'authentification de la source, l'intégrité des données mais pas de confidentialité
- En-tête AH inséré entre l'en-tête IP et le champ de données.
- Champ protocole : 51
- Les routeurs intermédiaires traitent les datagrammes comme d'habitude
- **L'en-tête AH inclut**
 - L'identifiant de connexion
 - Données d'authentification : condensat signé par la source et calculé à partir du datagramme IP original.
 - Champ next header : spécifie le type des données (ex : TCP, UDP, ICMP)

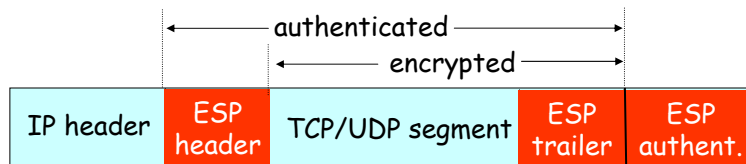
IP header

AH header

data (e.g., TCP, UDP segment)

Protocole ESP

- Fournit le secret, l'authentification de l'hôte, l'intégrité des données.
- Les données et l'ESP trailer sont chiffrés.
- Le champ next header est dans l'ESP trailer.
- Le champ d'authentification d'ESP est similaire au champ d'authentification d'AH
- Protocol = 50



Sécurité dans IEEE 802.11

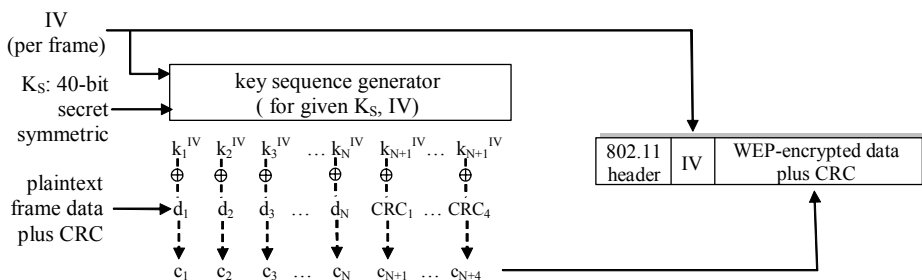
- *Nombreux réseaux IEEE 802.11 disponibles*
 - La plupart n'utilisent pas de chiffrement ni d'authentification
 - Facile de faire du packet-sniffing et autres attaques !
- *Wired Equivalent Privacy (WEP):* authentification comme dans le protocole *ap4.0*
 - L'hôte demande l'authentification au point d'accès
 - Le point d'accès envoie un nonce de 128 bits
 - L'hôte chiffre le nonce en utilisant une clé symétrique partagée
 - Le point d'accès déchiffre le nonce et authentifie l'hôte

Sécurité dans IEEE 802.11

- **Wired Equivalent Privacy (WEP) : chiffrement des données**
 - L'hôte et le point d'accès partagent une clé symétrique de 40 bits (semi-permanente)
 - L'hôte ouvre un vecteur d'initialisation (IV) de 24 bits pour créer une clé de 64 bits
 - La clé de 64 bits est utilisée pour générer un flux de clés, k_i^{IV}
 - k_i^{IV} est utilisée pour chiffrer le i-ème octet, d_i :

$$c_i = d_i \text{ XOR } k_i^{IV}$$
 - IV et les octets chiffrés, c_i sont envoyés dans la trame

802.11 : chiffrement WEP



Sender-side WEP encryption

Casser le chiffrement WEP dans 802.11

Trou de sécurité :

- IV sur 24 bits, un IV par trame, -> l'IV peut être réutilisé
- IV transmis en clair -> la réutilisation d'IV peut être détectée
- **Attaque :**
 - Trudy contraint Alice à chiffrer du texte connu $d_1 d_2 d_3 d_4 \dots$
 - Trudy voit : $c_i = d_i \text{ XOR } k_i^{\text{IV}}$
 - Trudy connaît $c_i d_i$, donc peut calculer k_i^{IV}
 - Trudy connaît la séquence de chiffrement de clés $k_1^{\text{IV}} k_2^{\text{IV}} k_3^{\text{IV}} \dots$
 - La prochaine fois que IV sera utilisé, Trudy pourra déchiffrer !

Sécurité des réseaux (résumé)

Techniques de base.....

- cryptographie (symétrique et publique)
- authentification
- Intégrité des messages
- Distribution des clés

.... utilisées dans de nombreux scénarios de sécurité différents

- Email sécurisé
- transport sécurisé (SSL)
- IP sec
- 802.11 WEP

Ressources

- Livre
- *Computer Networking : A Top Down Approach
Featuring the Internet,*
2nd edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2002.



• J'en suis là