Virtual Private Networks
MPLS
IPsec
SSL/TLS

Ahmed Mehaoua
Professeur
Université de Paris 5
mea@math-info.univ-paris5.fr

Plan

- I. les VPN : définitions
- II. Les VPN MPLS
- III. Les VPN IPsec
- IV. Les VPN SSL/TLS

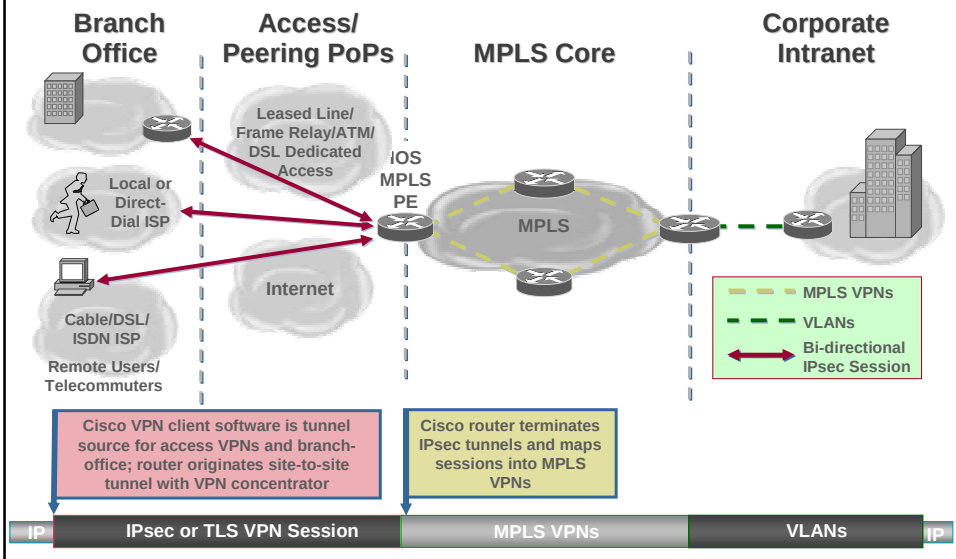
Architectures VPN

Communication layers	Security protocols
Application layer	ssh, S/MIME, PGP, SRTP
Transport layer	SSL, TLS, WTLS
Network layer	IPsec
Data Link layer	PPTP, L2TP MPLS
Physical layer	Scrambling, Hopping, Quantum Communications

➢ VIII. Application de IPsec : les VPN

A "VPN service" is a service which offers secure, reliable connectivity over a shared public network infrastructure such as the Internet. Because the infrastructure is "shared", connectivity can be provided at lower cost than existing dedicated private networks

Extranet VPN : IPsec to MPLS to VLAN

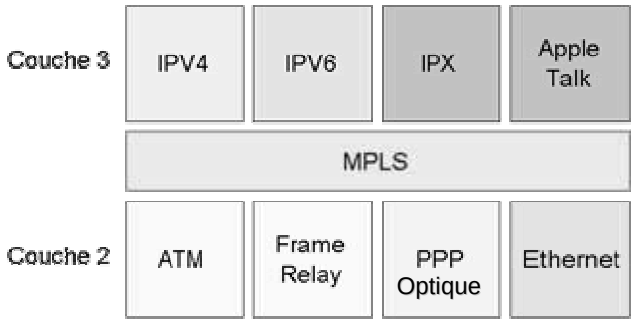


Typologie des technologies VPN

- VPN de niveau 2
- VPN-IP avec MPLS
- VPN-IP avec IPSEC
- VPN-SSL

MPLS : multiprotocole ?

Multi-Protocole Label Switching



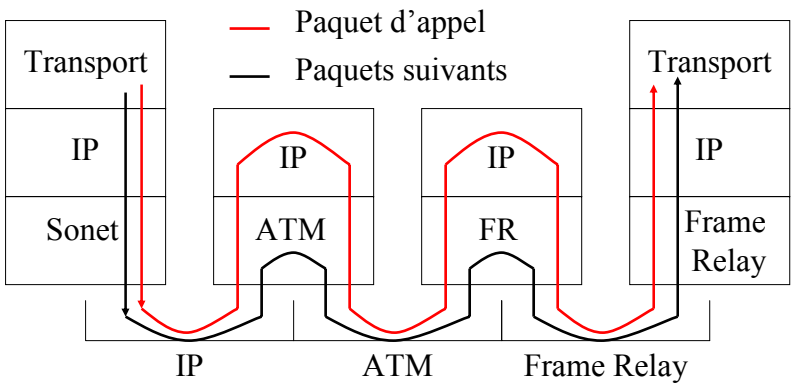
VPN MPLS

- Avantages: Une bonne partie des avantages de MPLS reposent sur son support du routage explicite :

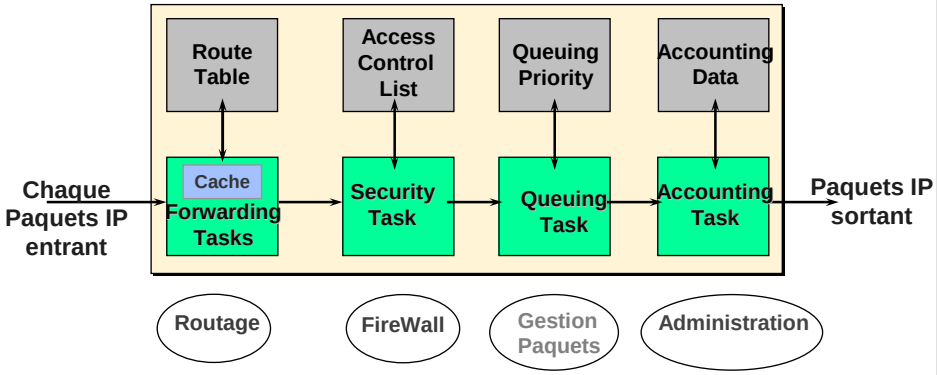
1. Ingénierie de trafic (répartition du trafic/charge entre plusieurs chemins)
2. Qualité de service (pré-dimensionnement des chemins)
3. Intégration de différentes infrastructures de commutation de bout en bout via les labels MPLS
4. Indépendant des protocoles de routage IP (inter et intra AS) (transparence pour les clients)
5. Externernalisation du réseau privé virtuel (VPN) : recentrage des entreprises sur leur cœur de marché et de compétence; et bénéficier des nouvelles technologies réseaux;

- Applications : Les VPN IP sont considérés comme l'application de choix pour MPLS. Grâce à ce dernier, les opérateurs peuvent garantir une qualité de service de bout en bout sur leurs réseaux IP. Le trafic émanant d'un RPV pourra être étiqueté et injecté dans les réseaux des opérateurs et des fournisseurs d'accès à Internet (FAI), et se voir attribuer une qualité de service indépendante des autres flux transitant sur ces réseaux.

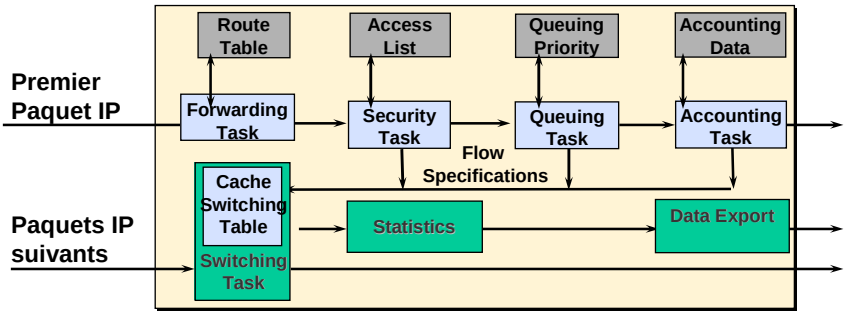
Evolution du Routage IP
- Vers la Commutation IP -



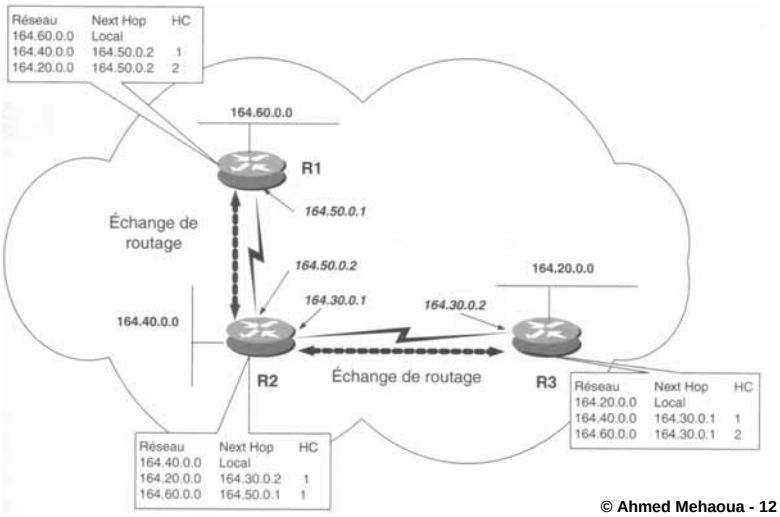
Fonctions d'un Routeur IP



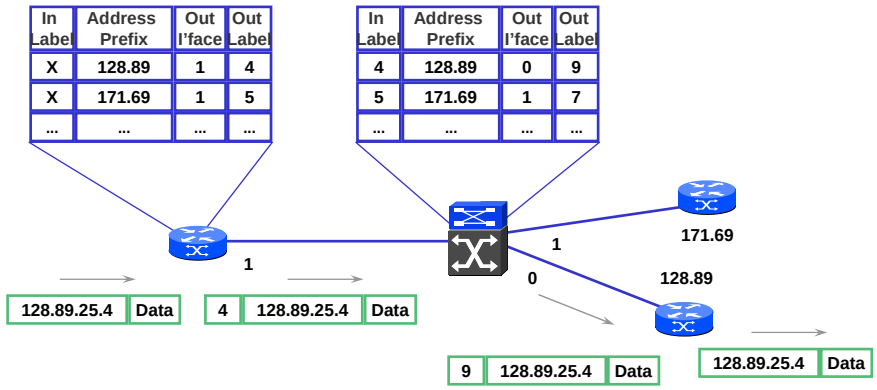
Evolution du Routage IP
- Vers la Commutation IP -



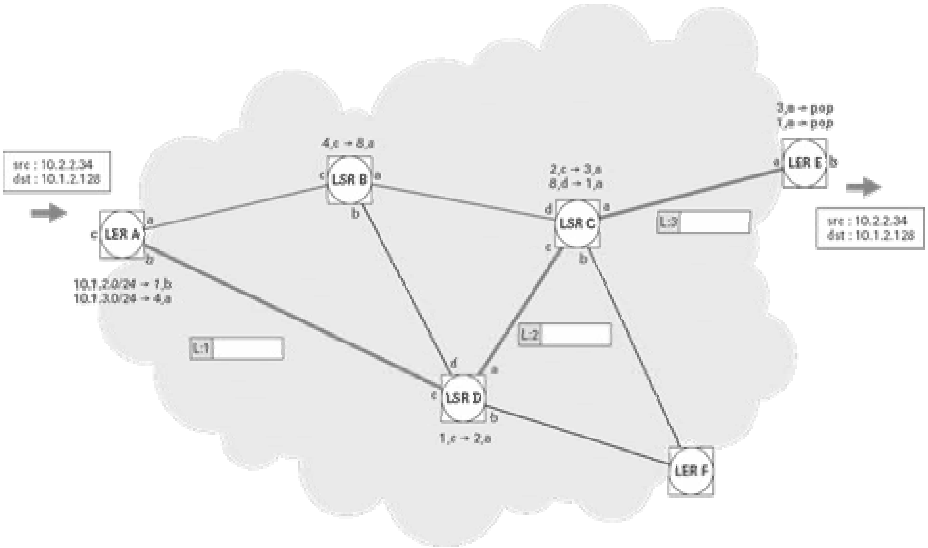
Routage IP intra-AS
Protocole RIP



LABEL SWITCHING



LABEL SWITCHING

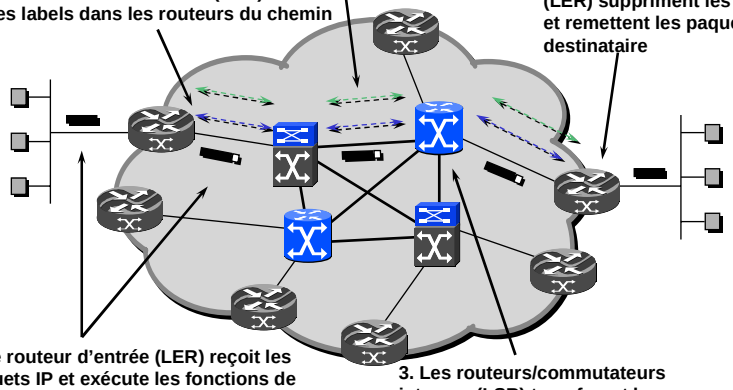


MPLS : Opérations

1a. Protocole de routage existant (e.g. OSPF, IGRP) détermine la meilleure route dans le domaine MPLS

1b. Le Label Distribution Protocol (LDP) Initialise les labels dans les routeurs du chemin

4. Les routeurs de sortie (LER) suppriment les labels et remettent les paquets au destinataire



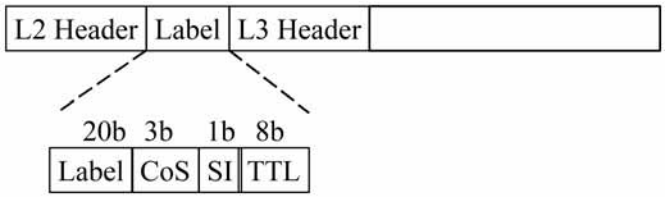
2. Le routeur d'entrée (LER) reçoit les paquets IP et exécute les fonctions de niveau 3 puis insère les labels

3. Les routeurs/commutateurs internes (LSR) transferent les paquets labelisés en utilisant la commutation de label

MPLS : Terminologie

- **LDP:** Label Distribution Protocol
- **LSP:** Label Switched Path
- **FEC:** Forwarding Equivalence Class
- **LSR:** Label Switching Router
- **LER:** Label Edge Router
- **MPLS-shim** : en-tête MPLS

Codification des Labels
FEC : Forward Equivalent Class

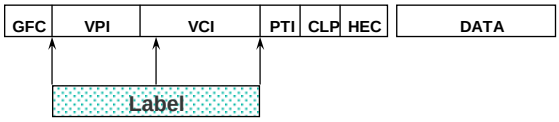


Label de taille fixe 4 octets (32 bits) :

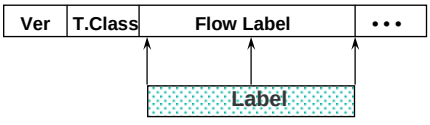
1. **Label (20 bits) : valeur du label**
2. **CoS (3 bits): classe de service du paquet (exp: champs ToS dans IP)**
3. **S (Stack Indicator) : indique le bas de la pile. Il est mit à un 1 pour le dernier label dans la pile et a 0 pour tous les autres labels.**
4. **TTL (8 bits): durée de vie du paquet (pour éviter les boucles).**

LABEL SWITCHING

ATM Cell Header



IPv6 Flow Label Field



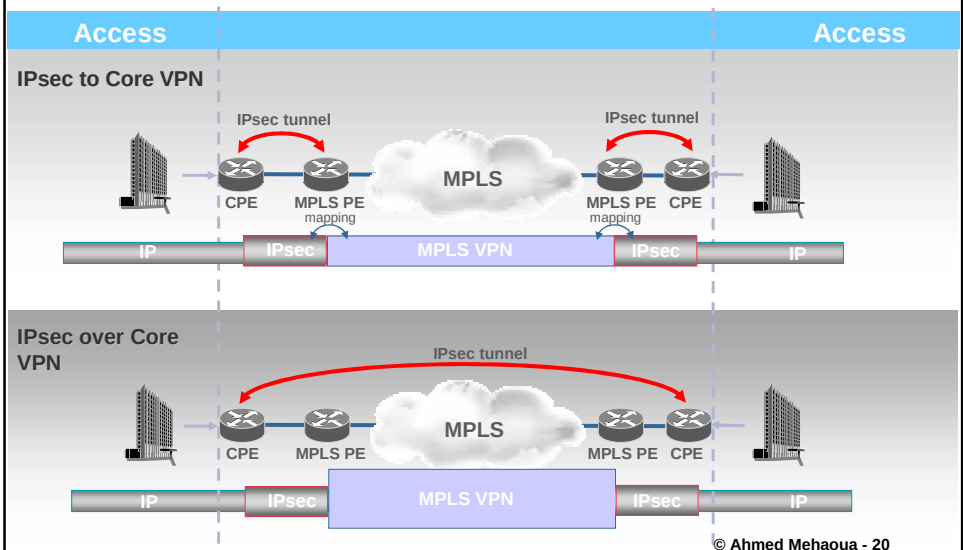
PPP Header : Optic

LAN MAC Tag header : Ethernet

Typologie des technologies VPN

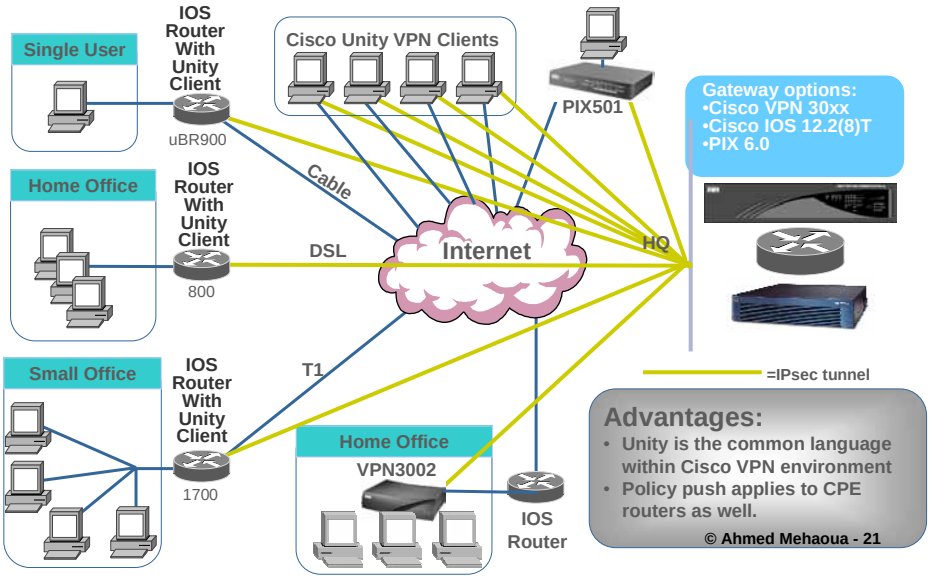
- VPN de niveau 2
- VPN-IP avec MPLS
- VPN-IP avec IPSEC
- VPN-SSL

➢ VIII. Application de IPsec : les VPN



Easy VPN Remote/Server

HUB site with static known IP address, client have dynamic addresses



Les offres commerciales de VPN pour l'entreprise

Les grands acteurs en France et en Europe aujourd'hui:

- Level 3
- France Telecom (Oleane VPN)
- Easynet
- Qwest
- Global Crossing

© Ahmed Mehaoua - 22

VPN IP- Offre en France -

Il existent 10 offrent de réseaux privés virtuel IP en France (01réseaux/Sept. 2002)

- Cable & Wireless (IPsec, aucune classe, 50 PoP nat, 49 PoP int.)
- Cegetel/Infonet (MPLS, 3 classes: Std, Data, Tps réel, 160, 55 pays)
- Colt (IPsec, 4, 13, via offre IP Corporate)
- FT/Global One (MPLS, 3, 60, 140)
- KPNQwest (IPSec, 5 classes, 30, 300)
- Maiaah (MPLS, 5, 7, via réseaux tiers)
- QoS Networks (IPSec, 5 classes, 1, 9)
- LDcom/Siris (MPLS, 4 classes, 77, 0)
- Worldcom (MPLS)

Critère des choix d'un fournisseur de VPN

- La bande passante proposée
- La qualité de service négociée (SLA)
- Gamme d'options d'accès en boucles locales
- Assistance

Sécurisation des communications IP

- Pour sécuriser les échanges ayant lieu sur un réseau TCP/IP, il existe plusieurs approches :
  - niveau applicatif (PGP)
  - niveau transport (protocoles TLS/SSL, SSH)
  - niveau physique (boîtiers chiffrant).
- IPsec vise à sécuriser les échanges au niveau de la couche réseau.
- IPsec veut dire IP Security Protocols : ensemble de mécanismes de sécurité commun à IPv4 et IPv6.

IPSec : le standard

- Norme prévue pour IPv6
- Adaptée à IPv4, vu la lenteur de déploiement IPv6 et les besoins forts des entreprises
- Série de RFC : 2401, 2402, 2406, 2408
- Très nombreuses pages !!

Les services IPsec

- **Services de sécurités offerts par IPsec :**
  - **Authentification des extrémités**
  - **Confidentialité des données échangées**
  - **Authenticité des données**
  - **Intégrité des données échangées**
  - **Protection contre les écoutes et analyses de trafic**
  - **Protection contre le rejeu**

- **2 modes d'exploitation d'IPsec :**
  - **Transport : Protège juste les données transportées (LAN)**
  - **Tunnel : Protège en plus l'en-tête IP (VPN)**

- **IPsec permet :**
  - **La mise en place de VPN**
  - **Sécuriser les accès distants (Utilisation nomade)**
  - **Protection d'un serveur sensible**

Composants d'IPsec

- Protocoles de sécurité :
  - Authentication Header (AH)
  - Encapsulation Security Payload (ESP)
- Protocole d'échange de clefs :
  - Internet Key Exchange (IKE)
- Bases de données internes :
  - Security Policy Database (SPD)
  - Security Association Database (SAD)

I. Normalisation d'IPsec par l'IETF

RFC 2401	Security Architecture for the Internet Protocol	S. Kent, R. Atkinson	Novembre 1998	STANDARD
RFC 2402	IP Authentication Header			
RFC 2406	IP Encapsulating Security Payload (ESP)			

Quel protocole pour quel service de sécurité ?

Table 16.1 IPSec Services

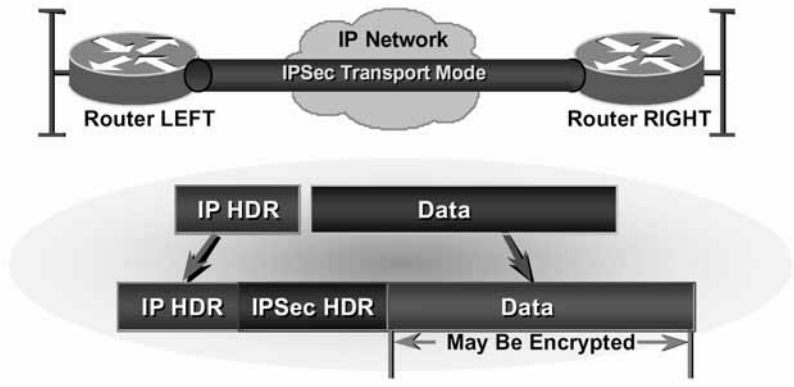
	AH	ESP (encryption only)	ESP (encryption plus authentication)
Access control	✔	✔	✔
Connectionless integrity	✔		✔
Data origin authentication	✔		✔
Rejection of replayed packets	✔	✔	✔
Confidentiality		✔	✔
Limited traffic flow confidentiality		✔	✔

II. Modes d'IPsec

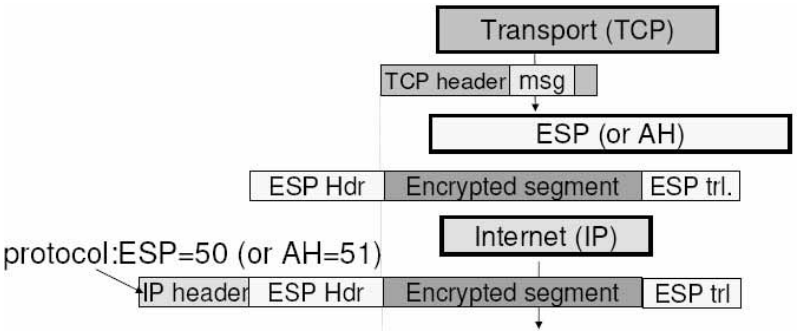
• mode transport
- insertion transparente entre TCP et IP
- pas de masquage d'adresse
- facilité de mise en œuvre
⇨ sécurise de bout en bout les échanges entre deux utilisateurs.

• mode tunnel
- insertion après IP
- encapsulation des datagrammes IP dans d'autres datagrammes IP
- masquage d'adresse
⇨ sécurise lien par lien les segments de réseau.
⇨ utilisé quand au moins une des deux extrémités d'IPsec se comporte comme une passerelle

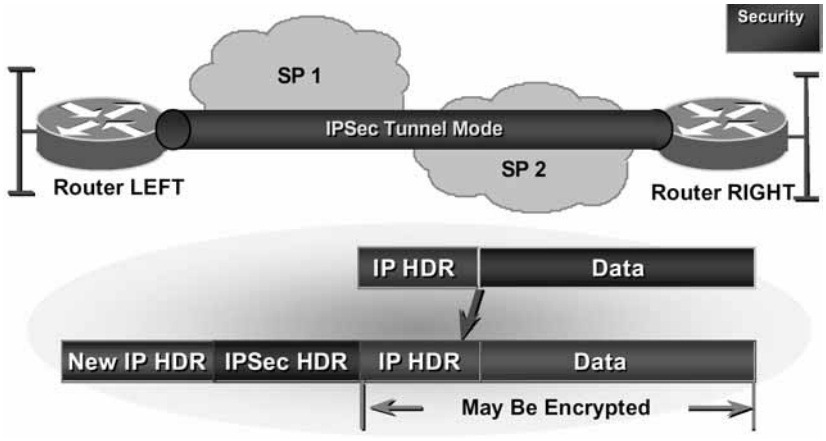
IPsec : Mode Transport



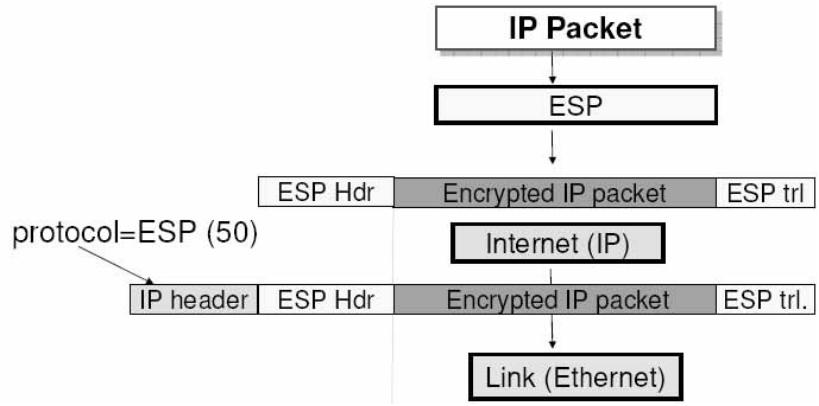
Transport Mode Encapsulation



IPsec : Mode Tunnel



Tunnel Mode Encapsulation



III. Protocole de sécurité AH

- AH = Authentication Header : 1er protocole et aussi le plus simple
- définit dans le RFC 2402
- Garantit :

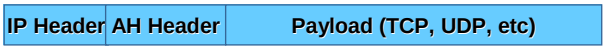
- l'authentification.
- l'unicité (anti-rejeu)
- l'intégrité

! Pas de confidentialité !
=> les données sont seulement signées mais pas chiffrées
support des algorithmes MD5 (128 bits) et SHA-1

IPSec protocols – AH protocol

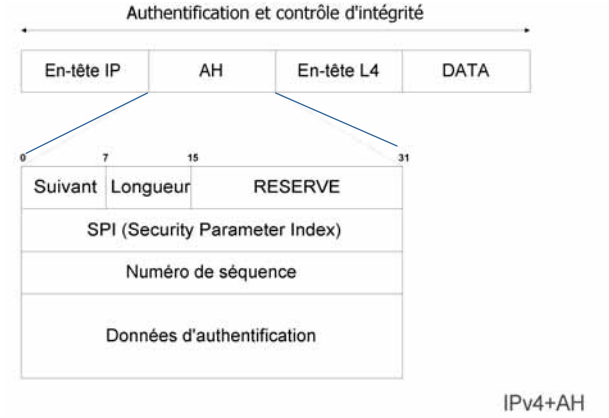
- **AH - Authentication Header**
  - **Defined in RFC 1826**
  - **Integrity: Yes, including IP header**
  - **Authentication: Yes**
  - **Non-repudiation: Depends on cryptography algorithm.**
  - **Encryption: No**
  - **Replay Protection: Yes**

Transport Packet layout



Tunnel Packet layout

III. Protocole de sécurité AH



IPv4+AH

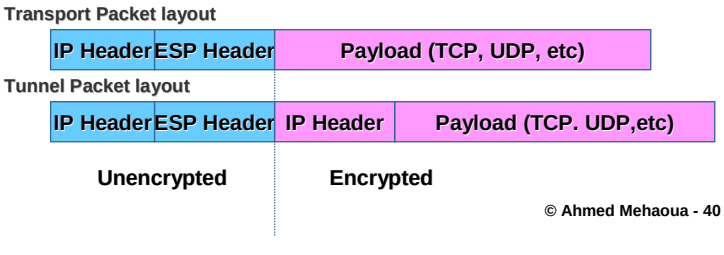
IV. Protocole de sécurité ESP

- ESP = Encapsulating Security Payload
- définit dans le RFC 2406
- Seules les données sont protégées (pas de protection en-tête)
- Garantit:

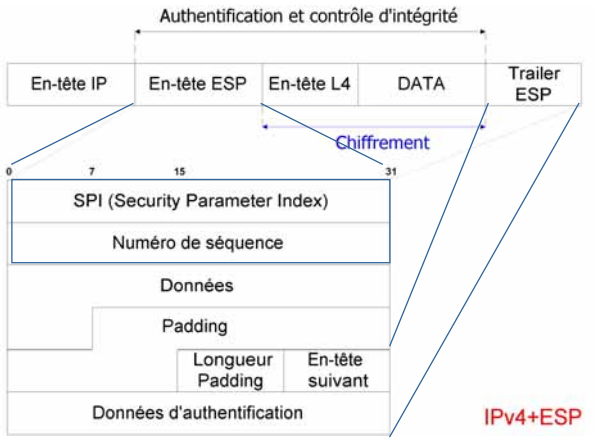
- l'authentification.
- l'unicité (anti-rejeu)
- l'intégrité
- la confidentialité

IPSec protocols – ESP protocol

- **ESP – Encapsulating Security Payload**
  - **Defined in RFC 1827**
  - **Integrity: Yes**
  - **Authentication: Depends on cryptography algorithm.**
  - **Non-repudiation: No**
  - **Encryption: Yes**
  - **Replay Protection: Yes**



IV. Protocole de sécurité ESP



V. Fonctionnement SA, SAD, SPD

❶ SA : 'Security Association'

- Les Associations de sécurité (SA) définissent les paramètres des divers mécanismes utilisés pour la sécurisation des flux sur le réseau privé virtuel
- A chaque SA correspond un bloc de données identifié par un index et contenant les informations correspondantes
- Plus précisément, chaque association est identifiée de manière unique à l'aide d'un triplet composé de :
  - le SPI (Security Parameter Index) : index de la SA défini par le récepteur
  - l'adresse de destination des paquets
  - l'identifiant du protocole de sécurité (AH ou ESP)

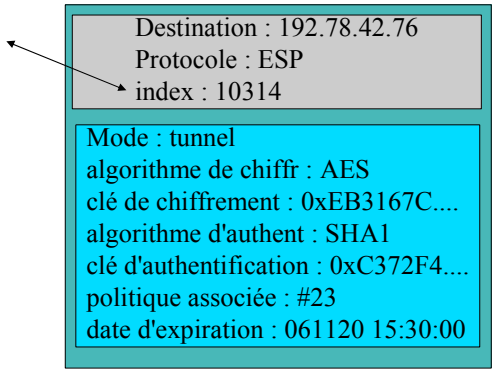
V. Fonctionnement SA, SAD, SPD

❶ SA : 'Security Association' (suite)

Les informations échangées sont :
- index de la SA appelé SPI (pour Security Parameter Index)
- un numéro de séquence, indicateur utilisé pour le service d'anti-rejeu
- une fenêtre d'anti-rejeu : compteur 32 bits
- dépassement de séquence
- paramètres d'authentification (algorithmes et clés)
- paramètres de chiffrement (idem)
- temps de vie de la SA
- mode du protocole IPsec (tunnel ou transport)

Attention : un SA est Unidirectionnelle: protéger les deux sens d'une communication classique requiert deux associations.

SA : exemple



V. Fonctionnement SA, SAD, SPD

❷ **SAD : Security Association Database**

- base de données des SA pour la gestion des associations de sécurité actives

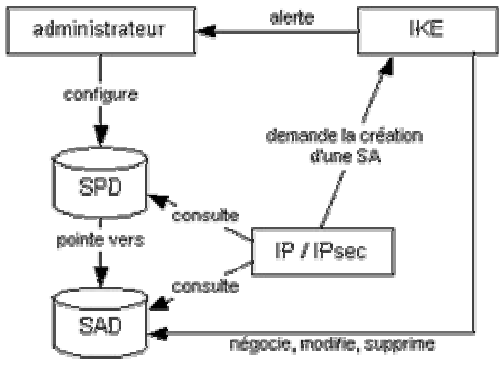
- consultée pour savoir quels mécanismes il faut appliquer à chaque paquet reçu ou à émettre.

❸ **SPD : Security Policy Database**

- base de données des politiques de sécurité (SPD).

- Pour savoir comment appliquer les mécanismes sur les paquets

V. Fonctionnement SA, SAD, SPD



V. Fonctionnement SA, SAD, SPD

Exemple 1 : trafic sortant

1) IPsec reçoit des données à envoyer
2) Consultation de la base de données SPD (quel traitement pour ces données ?)
3) Mécanismes de sécurité pour ce trafic ?
4) Oui ⇨ récupération des caractéristiques requises pour la SA et consultation de la base SAD
5) SA existe ?
6) Oui ⇨ utilisée pour traiter le trafic en question
   Non ⇨ appel à IKE pour établir une nouvelle SA

V. Fonctionnement SA, SAD, SPD

Exemple 2 : trafic entrant

1) Réception paquet
2) Examen l'en-tête: services IPsec appliqués ?
3) Oui ⇨ références de la SA ? ⇨ consultation de la SAD
4) Déchiffrement
5) Consultation de la SPD : « SA du paquet correspond bien à celle requise par les politiques de sécurité ? »

VI. Gestion, distribution des clefs – IKE / ISAKMP

- Problèmatique : afin d'échanger des données de façon sécurisée, il est nécessaire de se mettre d'accord sur les paramètres à utiliser, et notamment d'échanger les clefs de session
- Il faut 2 paires de clés (AH et ESP) : soit 2 par direction.

- 1er solution : configuration manuelle des équipements
  - **(unique méthode proposée dans la première version d'IPSec…)**

- 2eme solution : gestion dynamique des paramètres au moyen d'un protocole sécurisé adapté
  - système automatique pour la creation à la demande des clés pour les SA
  - Plusieurs solutions : SKIP, Prothuris, Oakley, SKEME and ISAKMP ➔ IKE
  - IKE est un protocole orienté connexion utilisé par les équipements IPsec pour échanger et gérer les associations de sécurité à travers l'Internet :
    - Echange de clefs à l'aide de protocoles cryptographiques
    - Fournit une authentification des entités
    - Permet un établissement de SA à travers un réseau non sécurisé

VI. Gestion, distribution des clefs – IKE / ISAKMP

❶ IKE : Internet Key Exchange : Qu'est ce que c'est ?

➢ IKE est un ensemble de protocoles et de mécanismes assurant une gestion sécurisée et dynamique des paramètres de sécurité utilisés dans IPSec

➢IKE = échange de clés d'authentification + gestion des SA

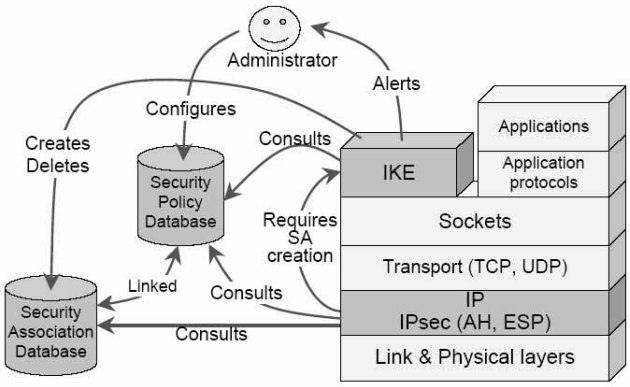
➢ RFC 2409

➢ Deux manières d'échanger des clefs :

- clefs pré-partagées
- certificats X.509

VI. Gestion, distribution des clefs – IKE / ISAKMP

❶ IKE : Internet Key Exchange



VII. Faiblesses d'IPsec

➢Limitations dues à la gestion manuelle des clefs

- AH et ESP s'appuient sur des numéros de séquence initialisés à 0 lors de la création d'une SA et incrémentés lors de l'envoi de chaque datagramme.
- Numéros de séquence sont stockés dans un entier de 32 bits ≈ 4 milliards
- Passé cette limite, nécessité de créer une nouvelle SA et donc une nouvelle clef.
- Possibilité de désactivation des numéros après la limite.

➢Broadcast et multicast

Problème de performance et impossibilité de résolution par l'augmentation de la puissance.

VII. Faiblesses d'IPsec (2)

➢Firewalls

Le filtrage de datagrammes IPsec est délicat pour deux raisons :

- les RFCs ne précisent pas si, sur un système remplissant simultanément les fonctions de passerelle de sécurité et de firewall, le décodage de l'IPsec doit avoir lieu avant ou après l'application des règles de firewalling ;
-il n'est pas possible au code de firewalling de lire certaines données, par exemple des numéros de port, dans des données chiffrées, ou transmises dans un format qu'il ne connaît pas.

➢NATs

Théoriquement, aucune translation d'adresse ne devrait affecter un datagramme IPsec, car ce type d'opération modifie le contenu des datagrammes, ce qui est incompatible avec les mécanismes de protection de l'intégrité des données d'IPsec.

VII. Faiblesses d'IPsec (suite 3)

➢ Non support de protocoles réseaux autres qu'IP

IPsec est un protocole qui ne prévoit que le convoyage sécurisé de datagrammes IP

Ceci n'est pas suffisant, car d'autres standards comme IPX et NetBIOS sont utilisés sur un grand nombre de réseaux. Il existe cependant une solution à ce problème : encapsuler les données à protéger dans du PPP, lui-même transporté par IPsec. Le rôle de PPP est en effet de permettre la transmission de différents protocoles au-dessus d'un lien existant.

Conclusion: IPSec-IKE-SA

- IPsec ensemble de protocoles et mécanismes pour le chiffrement de paquets IP
- SA->traitement à mettre en oeuvre sur un paquet IP quand il faut lui appliquer IPSec
- IKE->gestionnaire de SA
- IPSec->utilisateur de SA

Typologie des technologies VPN

- VPN de niveau 2
- VPN-IP avec MPLS
- VPN-IP avec IPSEC
- VPN-SSL

Technologies VPN

Communication layers	Security protocols
Application layer	ssh, S/MIME, PGP
Transport layer	SSL, TLS, WTLS
Network layer	IPsec
Data Link layer	PPTP, L2TP
Physical layer	Scrambling, Hopping, Quantum Communications

MPLS

Agenda

- Introduction
  - Motivation, evolution, standardization
  - Applications
- SSL Protocol
  - SSL phases and services
  - Sessions and connections
  - SSL protocols and layers
    - SSL Handshake protocol
    - SSL Record protocol / layer
- SSL solutions and products
- Conclusion

Sécurisation des échanges

- Pour sécuriser les échanges ayant lieu sur le réseau Internet, il existe plusieurs approches :
  - niveau applicatif (PGP)
  - niveau réseau (protocole IPsec)
  - niveau physique (boîtiers chiffrant).
- TLS/SSL vise à sécuriser les échanges au niveau de la couche Transport.
- Application typique : sécurisation des transactions Web

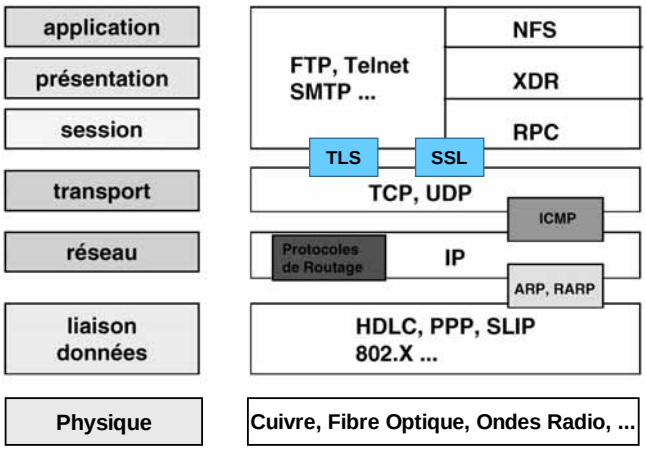
Transport Layer Security

- Advantages
  - Does not require enhancement to each application
  - NAT friendly
  - Firewall Friendly
- Disadvantages
  - Embedded in the application stack (some mis-implementation)
  - Protocol specific --> need to duplicated for each transport protocol
  - Need to maintain context for connection (not currently implemented for UDP)
  - Doesn't protect IP adresses & headers

Security-Sensitive Web Applications

- Online banking
- Online purchases, auctions, payments
- Restricted website access
- Software download
- Web-based Email
- Requirements
  - Authentication: Of server, of client, or (usually) of both
  - Integrity: Of requests, of responses, etc.
  - Confidentiality: Of data transfers
  - Availability: No Denial of Service
- Some minor applications : SSL VPN (end-to-end)
- Main tool: SSL / TLS protocol

IPsec et l'architecture TCP/IP

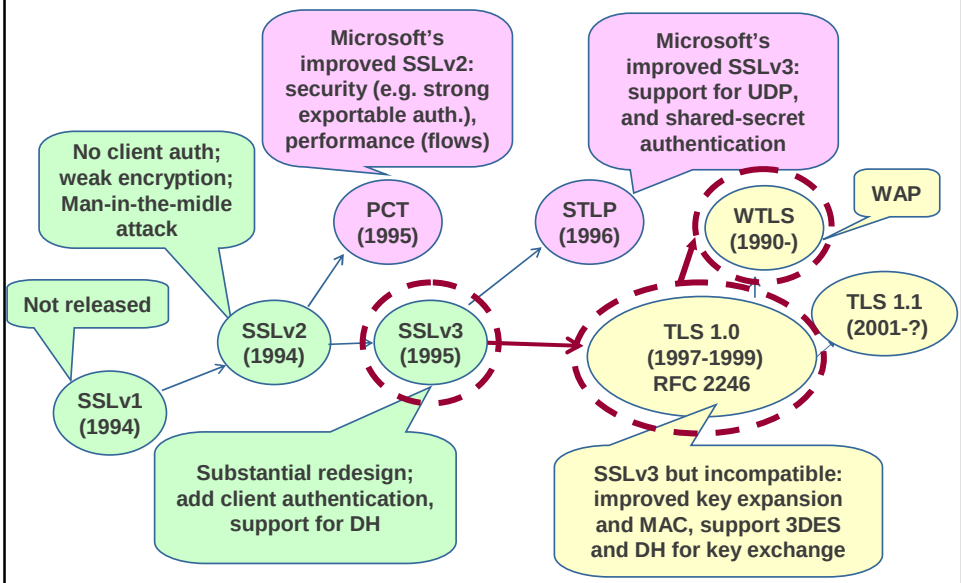


Transport Layer Security Protocols

- Connectionless and connection-oriented transport layer service:
  - Security Protocol 4 (SP4) – NSA, NIST
  - Transport Layer Security (TLSP) – ISO
- Connection-oriented transport layer service:
  - Encrypted Session Manager (ESM) – AT&T Bell Labs.
  - Secure Socket Layer (SSL) – Netscape Communications
  - Transport Layer Security (TLS) – IETF TLS WG

Most popular transport layer security protocols

SSL/TLS Evolution



DSS : Digital Signature Standard (Public-key)
DH : Diffie-Hellman

SSL/TLS in a Nutshell

- SSL & TLS provide a `secure TCP tunnel from client to server`:
  - Message Confidentiality
  - Message and connection integrity
  - Authentication of server, optionally also of client
- Implemented in almost all web clients, servers
- Many implementations, libraries, e.g. Open-SSL
- SSL: Secure Socket Layer
  - Version 3 designed by Netscape Corp.
  - Original goal and main use: secure credit card number
  - SSL (& TLS) operate on top of `standard` Sockets API
- TLS: Transport Layer Security
  - Version 1.0 – RFC 2246
  - IETF standard version of SSL
  - We usually say just SSL but refer to both

SSL/TLS : Applications and ports

Protocol	Normal Port	Secured Protocol	Encrypted Port	RFC
HTTP	80	**HTTPS**	443	2818
NTP	119		563	
LDAP	389	**SLDAP**	636	2830
FTP-DATA	20	**SFTP**	989	draft
FTP-control	21	**SFTP**	990	draft
TELNET	23	**SSH**	22	4251
IMAP	143	**SIMAP**	993	2595
IRC	194		994	2813
POP3	110	**SPOP3**	995	2595
SMTP	25	**SMTP-TLS**	465 (25)	2487

SSL-based current solutions

- OpenSSL : www.openssl.org
  - a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols
- Win32 SSL API
- JavaSSL : www.bpsinfo.com/javassl/
  - A Java package which uses JNI to provide an SSLSocket class
- Apache-SSL : http://www.apache-ssl.org/
- OpenSSH : www.openssh.com/
  A port of the OpenSSL-based SSH package
- Mocana Security Suite : www.mocana.com
  - Software suite including (SSL, SSH, IPsec, IKE, Radius, …)
- sNFS : www.quick.com.au/ftp/pub/sjg/help/sNFS.html
  - An SSL-based NFS variant
- OpenVPN : www.openvpn.org

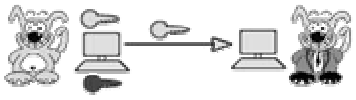
SSL/TLS principles

- Client / Server
- Provide 4 security services :
  - Authentication of server
  - Confidentiality of exchanged data
  - Integrity of exchanged data
  - Optional : authentication of client (if client has a certificate)
- Combining various security mechanisms :
  - Asymmetric Ciphering : authentication (RSA)
  - Certificate : to validate public key of the server
  - Symmetric Ciphering : Confidentiality of data transmission
  - Hach function : integrity of data (MD-5, SHA-1)

Principe de fonctionnement de SSL: Exemple du paiement en ligne

Sécurisation des transactions par SSL

1 Le client, connecté au site marchand sécurisé par SSL, clique sur un lien hypertexte déclenchant une requête de formulaire sécurisé et la création d'une clé privée que le client va conserver, et d'une clé publique qui sera expédiée au serveur marchand

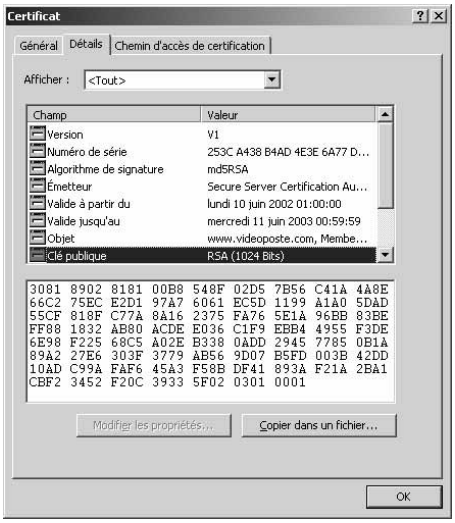


2 Le serveur marchand crée une clef de session en cryptant un message aléatoire à l'aide de la clef publique, puis l'envoie au client

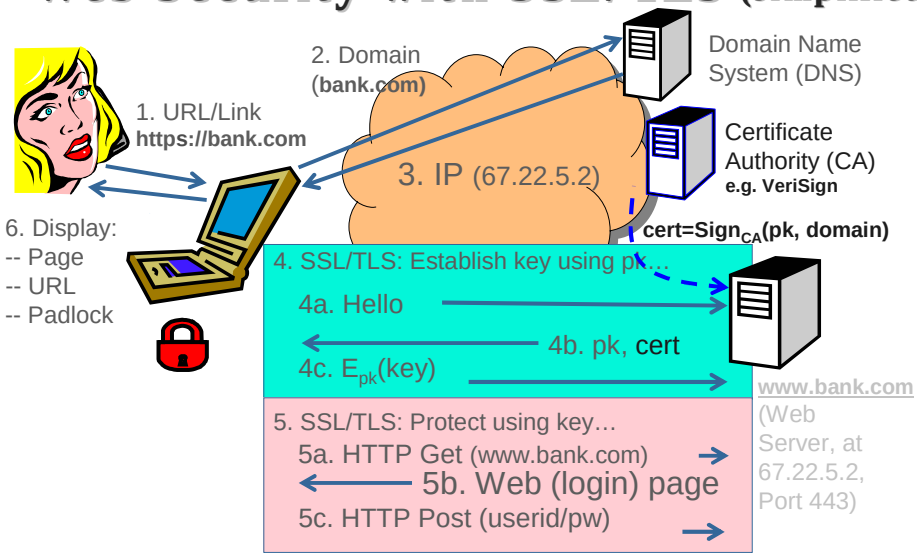
3 A réception, le client crypte la clef de session à l'aide de la clef privée, puis l'envoie au serveur marchand, qui va la décrypter à l'aide de la clé publique afin de vérifier l'authenticité du message, donc de l'acheteur

4 Le reste des transactions peut alors se faire à l'aide de la clef de session, connue des deux côtés et inconnue des autres entités du réseau

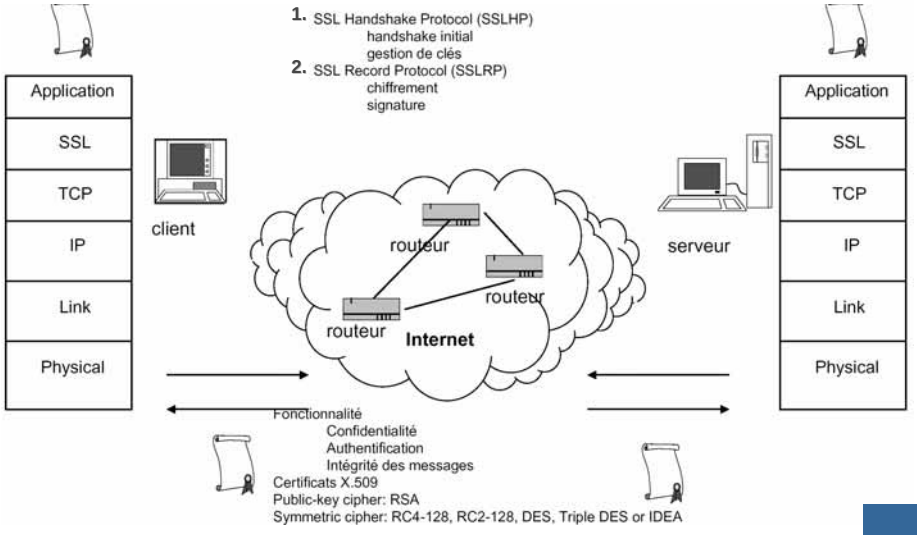
Exemple de Certificat X509



Web Security with SSL/TLS (simplified)

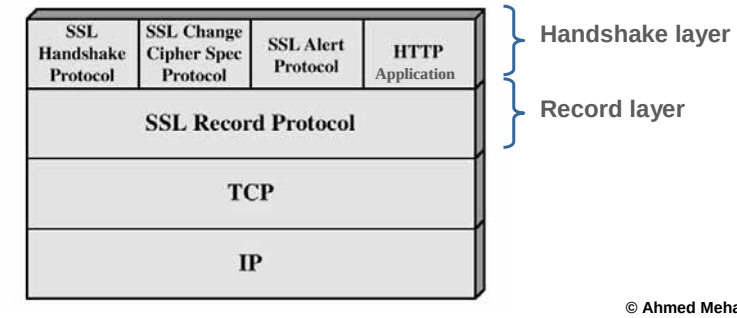


SSL Architecture



SSL Architecture (suite)

- SSL is built in two layers:
  - **SSL Handshake Layer** – used for managing SSL exchanges (cipher suite negotiation, session key generation, etc.)
  - **SSL Record Layer** – used to secure communication between client and server with the established session keys



SSL Main Protocols

- SSL Record Protocol
  - Layered on top of a connection-oriented and reliable transport layer service
  - Provides message origin authentication, data confidentiality, and data integrity
- Handshake Protocol
  - Used to mutually authenticate client and server and exchange session key

SSL Sub-Protocols

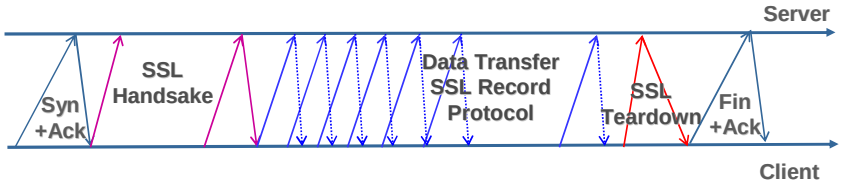
- Layered on top of the SSL Record Protocol
- Provides support for SSL session and connection establishment

- Alert Protocol
  - Used to transmit alerts via SSL Record Protocol
  - Alert message: (alert level, alert description)
- ChangeCipherSpec Protocol
  - Used to change cipher specifications
  - Can be changed at the end of the handshake or later
- Application Protocol
  - Used to directly pass application data to the SSL Record Protocol

SSL Operation Phases

Client uses SSL API to open connection

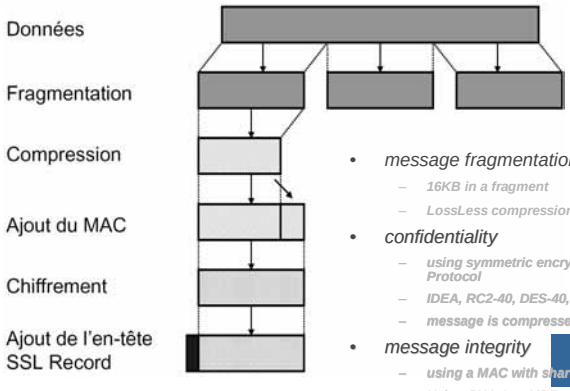
- TCP Connection Phase
- Handshake Phase (SSL Handshake Protocol)
  - Negotiate (agree on) algorithms, methods
  - Authenticate server and optionally client
  - Establish keys
  - Establish connection (keys and optionally Initialization Vector)
- Data transfer Phase (SSL Record Protocol)
- SSL Secure Teardown Phase



SSL Record Protocol

- Ce protocole fournit 2 services à une connexion SSL :
  - Confidentialité : définit une clé secrète pour le chiffrement
  - Intégrité du message : définit une clé secrète pour le calcul de l'empreinte
- SSL Record Protocol : opérations
  - Fragmentation :
    - Le message est fragmenté en blocs de taille maximum 2^{14} octets
  - Compression :
    - Cette opération est prévue dans les spécifications mais non implémentée
  - Calcul du MAC :
    - Utilise la clé secrète
    - Utilise l'algorithme SHA-1 ou MD5
  - Chiffrement :
    - Le message + MAC sont chiffrés avec un chiffrement symétrique
  - Ajout de l'en-tête :
    - 5 octets, composée de longueur du message, version, etc.

SSL Record Protocol (suite)



- *message fragmentation and compression*
  - *16KB in a fragment*
  - *LossLess compression (1 KB is OK)*
- *confidentiality*
  - *using symmetric encryption with a shared secret key defined by Handshake Protocol*
  - *IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128, AES*
  - *message is compressed before encryption (optional)*
- *message integrity*
  - *using a MAC with shared secret key*
  - *Using SHA-1 or MD5*
- *Non replay*
  - *Use counter (64 bits) to prevent replay in SSL session*

SSL Alert Protocol

- SSL Alert Protocol signals state changes and indicates errors
- SSL Alert is invoked by:
  - Handshake protocol – in case of problem
  - Record protocol – e.g. if MAC is not valid
  - Application – to close connection (close_notify)
  - Connections should be closed with close_notify to allow detection of truncation attacks (dropping last messages)
  - **Notice**: close_notify is normal, not a failure alert!
- The alerts are carried in an “Alert Record”

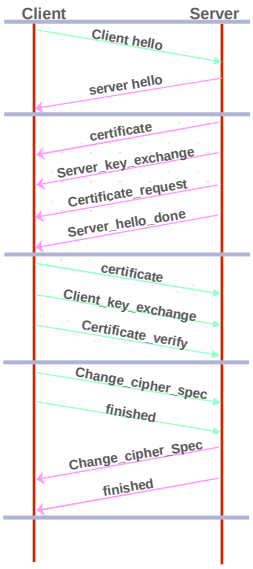
SSL Alert Protocol (2)

- Use two-byte message to convey SSL-related alerts to peer entity
  - First byte is severity level
    - warning(1) or fatal(2)
  - Second byte is specific alert
    - Always fatal: unexpected_message, bad_record_mac, decompression_failure, handshake_failure, illegal_parameter
    - Other alerts: close_notify, no_certificate, bad_certificate, unsupported_certificate, certificate_revoked, certificate_expired, certificate_unknown
- Compressed and encrypted like all SSL data

SSL Handshake Protocol

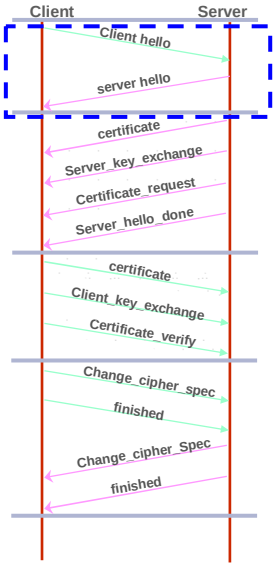
- Allow server and client to agree on cipher suite (algorithms and options):
  - authenticate server (madatory)
  - Authenticate client (option)
  - negotiate encryption algorithms (symmetric or asymmetric)
  - negotiate Signature & MAC algorithms
  - negotiate cryptographic keys to be used
  - Send certificate(s)
  - SSL Message compression
- Comprise a series of messages in 4 phases
  - Establish Security Capabilities
  - Server Authentication and Key Exchange
  - Client Authentication and Key Exchange
  - Finish

SSL Handshake Protocol Phases



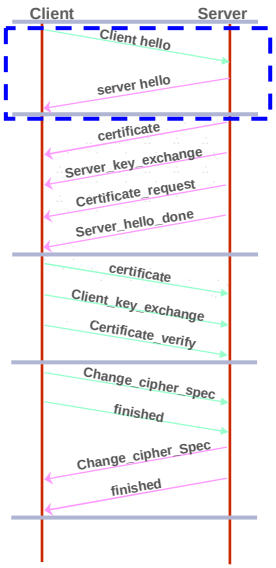
- **Phase 1 – Establish parameters**
  Negotiation of the session ID, key exchange algorithm, MAC algorithm, encryption algorithm, and exchange of initial random numbers
- **Phase 2 – Server authentication (optional: server key-exchange)**
  Server may send its certificate and key exchange message, and it may request the client to send a certificate. Server signals end of hello phase.
- **Phase 3 – Client key-exchange (optional: client authentication)**
  Client sends certificate if requested and may send an explicit certificate verification message. Client always sends its key exchange message.
- **Phase 4 – Finish: validation and begin using exchanged keys**
  Change cipher spec and finish handshake

Phase 1 – Establish Parameters



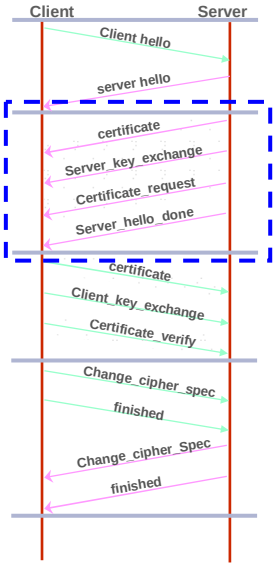
- `Client_hello` message
  - Version (highest available)
  - A timestamp and random string called **`clientHello.random`**
  - Session identifier (used for existing connections)
  - Cipher-suite – a list of supported cryptographic algorithms
  - Compression method
  - **`Client_random, server_random`**
  - From TLS 1.1: Extensions
- `server_hello` : same fields

Phase 1 – Establish Parameters (suite)



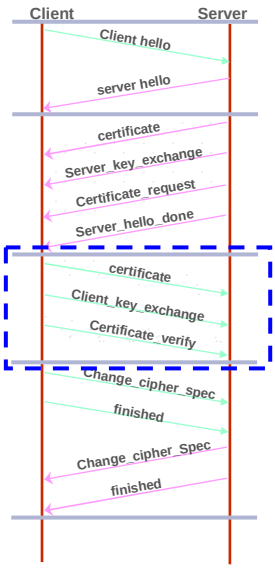
- Cipher-suite elements:
  - Key-exchange method (e.g. RSA)
  - Encryption algorithm (e.g. DES or RC4)
  - MAC (message authentication code) algorithm (e.g. SHA-1 or MD5)
- Server chooses one cipher-suite from those sent by client

Phase 2 – Server Authentication



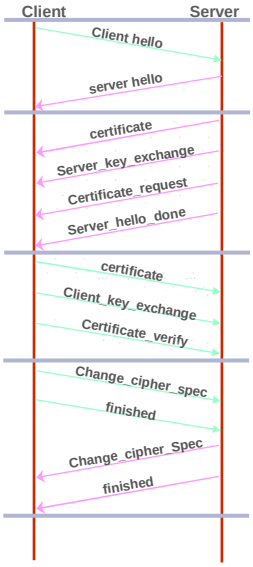
- Phase 2 – Server Authentication and (optional) Server Key-Exchange
- Server sends its `certificate`
  - It sends one or a chain of X.509 certificates
- Optional `Server_key_exchange` message
  - Used in Diffie-Hellman key exchange
  - Not used in RSA key exchange
- Optional `certificate_request` message: for client authentication
- `server_hello_done`

Phase 3 – Client Key-Exchange



- Phase 3 – Client Key-Exchange and (optional) Client Authentication
- The client verifies the server's certificate and sends its side of the key exchange
  - In Diffie-Hellman: the D-H key share
  - In RSA: encryption of random string
- If client authentication used (rarely): client sends certificate (but most clients don't have certificates)

Phase 4 – Finish



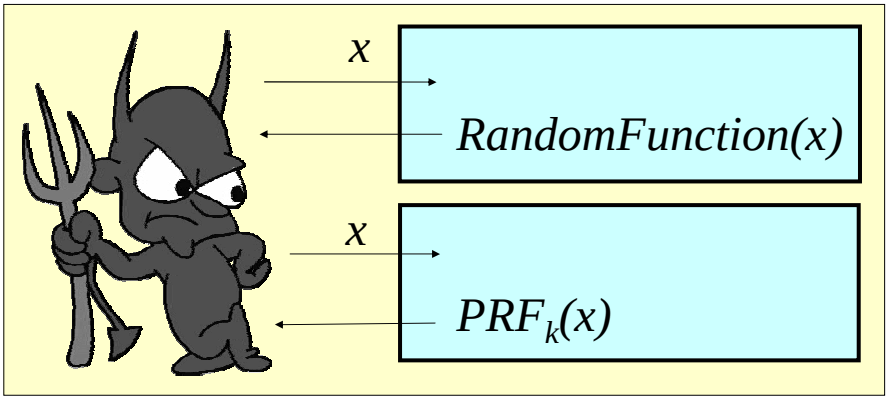
- Client and server send **`change_cipher_spec`** messages
  - Results in use of new cipher-suite (as negotiated in the hello phase)
- Client and server send **`finished`** messages
  - Messages are HMAC on all the handshake messages using **`master_secret`** as the key
  - The MAC is computed twice – once with MD5 and once with SHA1

SSL Connection State Variables

- Master Secret (shared key)
  - Unique to each connection
- Server and client sequence numbers
- *Server_random, client_random:* 32 bytes
  - Unique to each connection, selected by server and client
- Cryptographic keys, Initialization Vectors (IV)
  - Derived from Master Secret using a Pseudo-Random Function (PRF)
  - What's a PRF and how we use it?

A Pseudo-Random Function

- An efficient function using secret key
  - TLS's PRF is based on MD5 and SHA-1 (later…)
- That cannot be distinguished from random



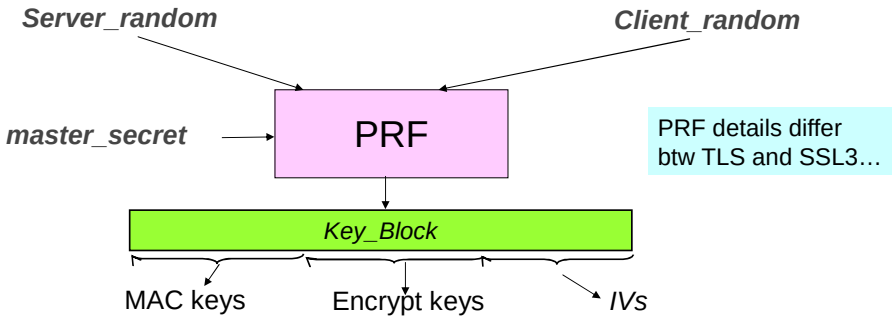
SSL Key Derivation

- The master key is used to derive the following six keys and values:
  - Client MAC key
  - Server MAC key
  - Client encryption key
  - Server encryption key
  - Client Init Vector (for CBC encryption)
  - Server Init Vector (for CBC encryption)
- Separate client and server keys are used
  - So successful attack against server does not compromise client, and vice versa

Deriving Connection Keys, IVs

Key_Block = PRF $_{master\_secret}$ *("key expansion"|| Server_random || Client_random)*

Split *Key_Block* to *ClientMACKey, serverMACKey, ClientEncryptKey,…*(using fixed order)



Summary: Trust & Security with SSL

- Confidentiality & authenticity of messages — Good in SSL/TLS
- Server (site) authentication:
  - Customer needs to identify site (bank, etc.)
- Client authentication:
  - Bank needs to identify account holder
  - Company needs to identify employee
  - Content provider needs to identify subscriber

  — Done, but…
- Non-repudiation:
  - Proof of making/receiving order/transaction
  - Prevent/resolve dispute, identify corruption
- Denial of service

  — Not in SSL

Conclusion

- SSL / TLS is the most widely deployed security protocol, standard
  - Easy to implement, deploy and use; widely available
  - Flexible, supports many scenarios and policies
  - Mature cryptographic design
- But SSL is not always the best tool…
  - Use IP-Sec e.g. for anti-clogging, broader protection, multicast
  - Use application security, e.g. s/mime, for non-repudiation, store-and-forward communication (not online)
- Beware of spoofing
  - Many browsers allow hard-to-detect spoofing
  - Many users will not detect simple spoofing (similar URL)