Les réseaux privés virtuels sécurisés (VPNs)

Filière SR2I – UE SR2I205

Rida Khatoun

rida.khatoun@telecom-paristech.fr

Plan

- Définition d'un VPN
- Types de VPN
- Protocoles IPsec
  - Protocoles de sécurisation : AH, ESP
  - Protocoles d'échange de clés : IKE
  - Négociation des paramètres de sécurité dans IPsec
- Exemples d'architectures VPN
- Mise en œuvre (démonstration sur Fortigate)
- VPN SSL/TLS

Définition d'un VPN

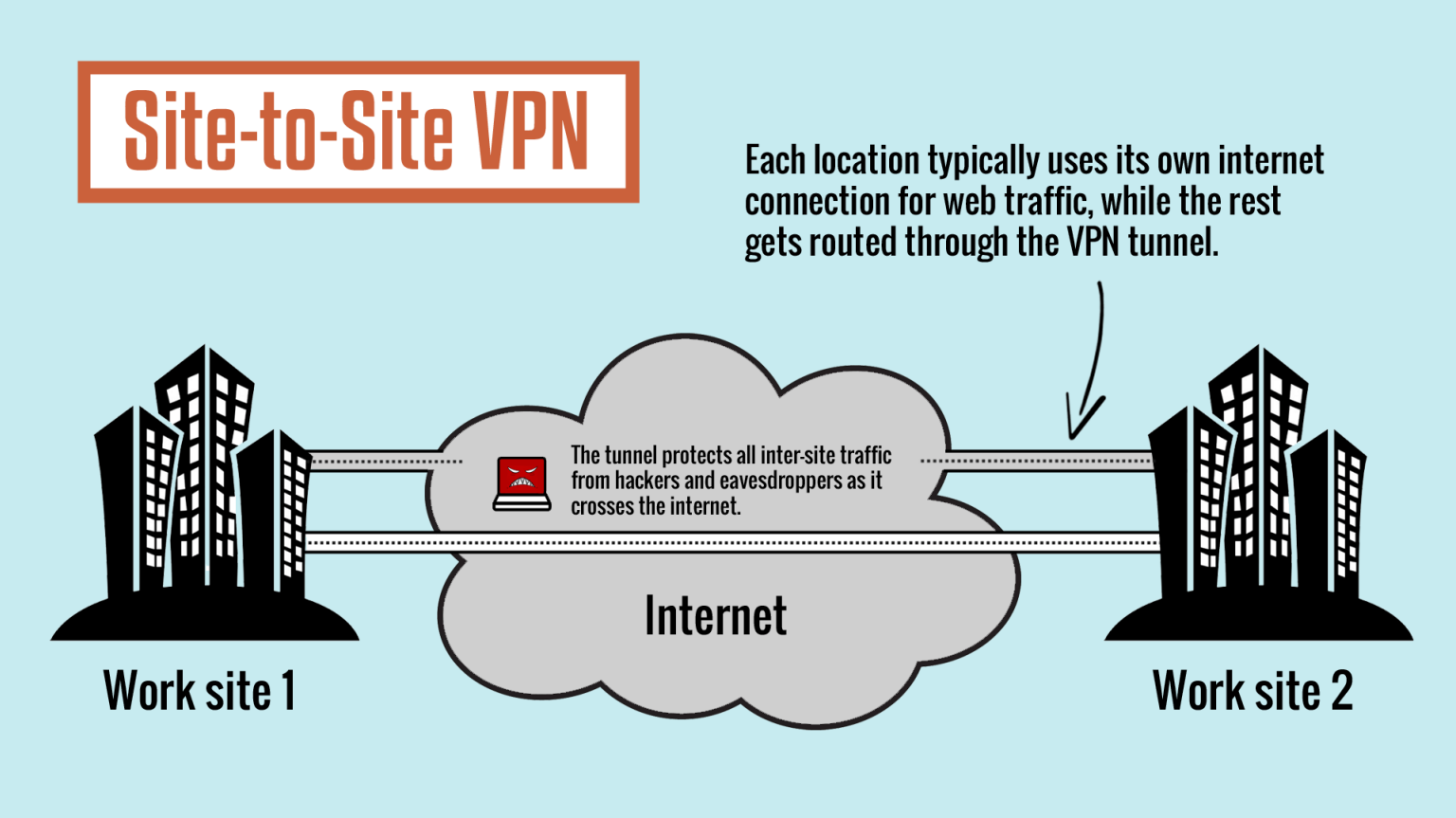
- VPN = Virtual Private Network
- Technologie réseau permettant de construire un réseau privé à l'intérieur d'une infrastructure publique
  - Privé = les échanges transitant par ce réseau sont confidentiels pour les autres utilisateurs du réseau public
  - Virtuel = le réseau privé ainsi crée n'est pas matérialisé par des liens physiques

Définition d'un VPN

- Pour que les données demeurent lisibles aux deux extrémités du tunnel, il faut utiliser le même protocole de tunneling dans tous les composants du VPN.
- Il existe plusieurs protocoles avec différents niveaux de sécurité, dont
  - PPTP (Point-to-Point Tunneling Protocol)
    - Niveau 2, rapide et peu sécurisé
  - L2F (Layer Two Forwarding),
  - GRE (Generic Routing Encapsulation)
  - L2TP (Layer Two Tunneling Protocol)
    - Niveau2, bien sécurisé, très utilisé, soucis avec les firewalls
  - l'IPSec :
    - Niveau 3, protection élevé, très utilisé

Types de VPN

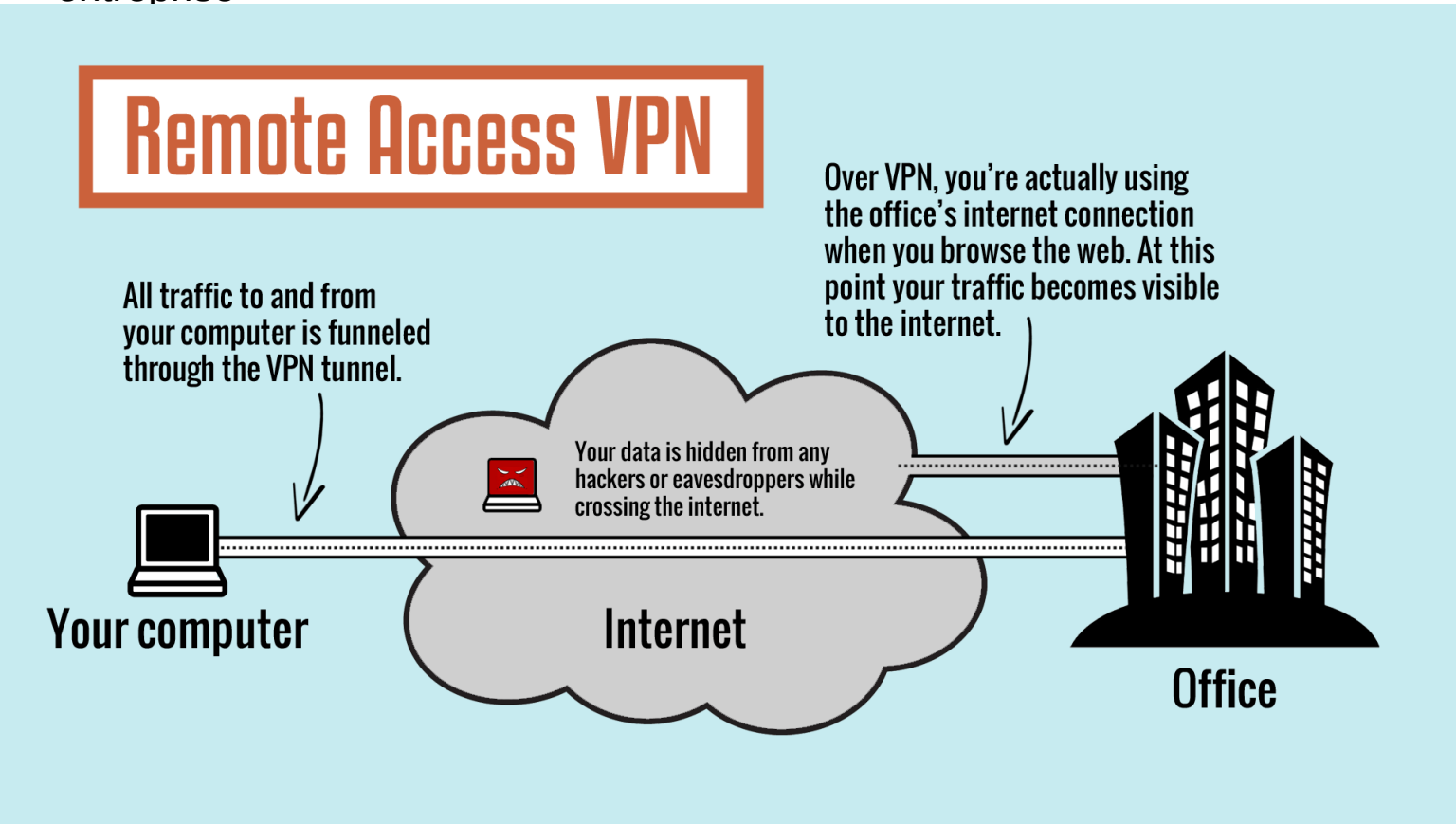
- VPN site-to-site
  - Permet de relier deux réseaux



Source: tiptopsecurity

Types de VPN

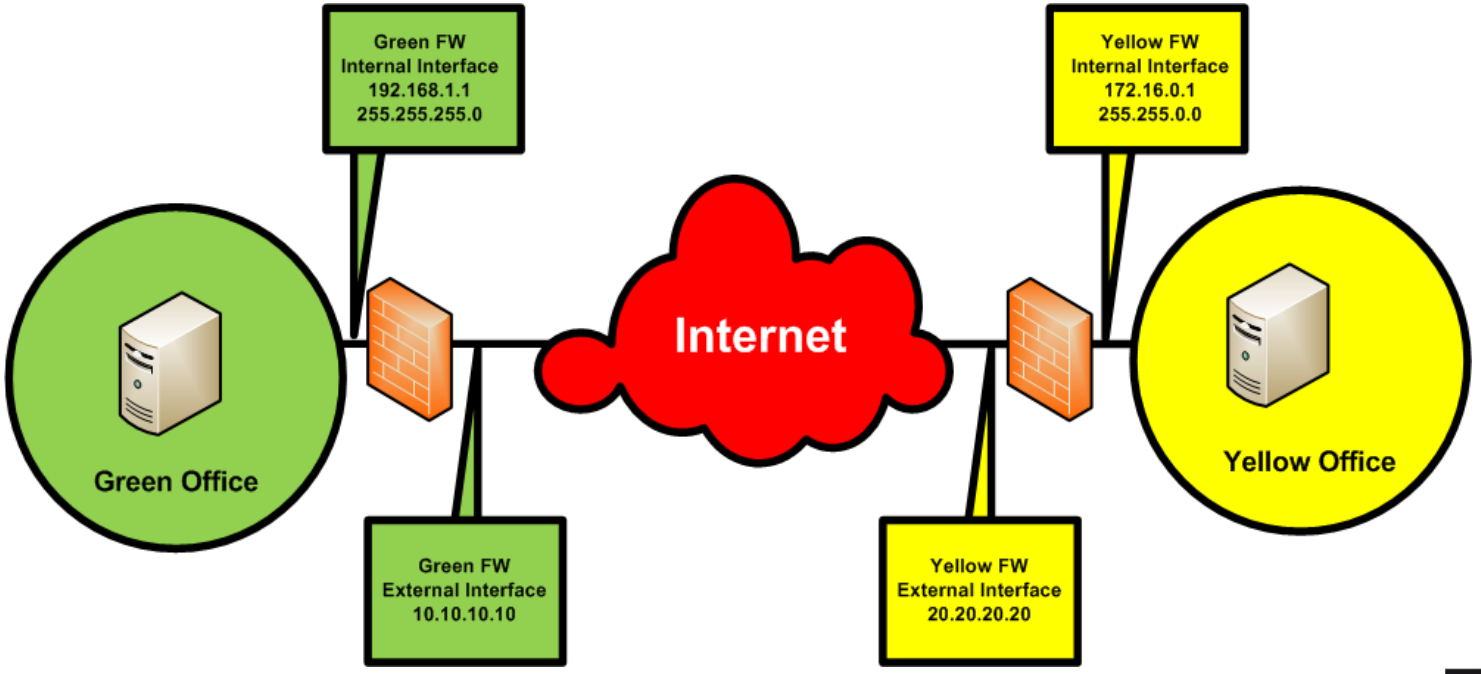
- VPN d'accès
  - Permet à des utilisateurs nomades d'accèder à distance au réseau de leur entreprise



Source: tiptopsecurity

Protocole IPsec

- Principes d'IPsec
  - IPsec ensemble de protocoles définis par IETF
  - Premier RFC en 1995 sans gestion de clés
  - 2’eme version en 1998 avec la gestion des clés (IKE)
  - Haut niveau de sécurité dans l'échange des paquets IP (niveau de la couche réseau)



Protocole IPsec

- Le protocole IPSec est basé sur :
  - IP Authentification Header (AH n°51, RFC 2402)
    - Intégrité, authentification
    - Permet de s'assurer de l'identité des deux extrémités du tunnel et de l'intégrité des données
    - Utilise MD5, SHA-1, SHA-256, HMAC,
    - En-tête AH inséré à la suite de l'en-tête IP
  - Encapsulating Security Payload (ESP,n°50, RFC 2406)
    - Chiffrement des paquets
    - ESP assure le chiffrement des données (3DES, AES, etc)
    - Paquet IP est chiffré et réencapsulé dans un autre paquet IP

Protocole IPsec

- AH et ESP dans les deux modes

	Mode Transport	Mode Tunnel
AH	Authentifie l'information utile IP + certains champs de l'en-tête IP	Authentifie le paquet IP entier + certains champs de l'en-tête externe
ESP	Chiffre l'information utilse IP	Chiffre tout le paquet IP
ESP (avec authentification)	Chiffre l'information utile IP et authentifie l'information utile IP	Chiffre et authentifie le paquet tout entier

Table: Fonctionnalité des modes tunnel et transport.

Protocole IPsec

- AH dans les deux modes

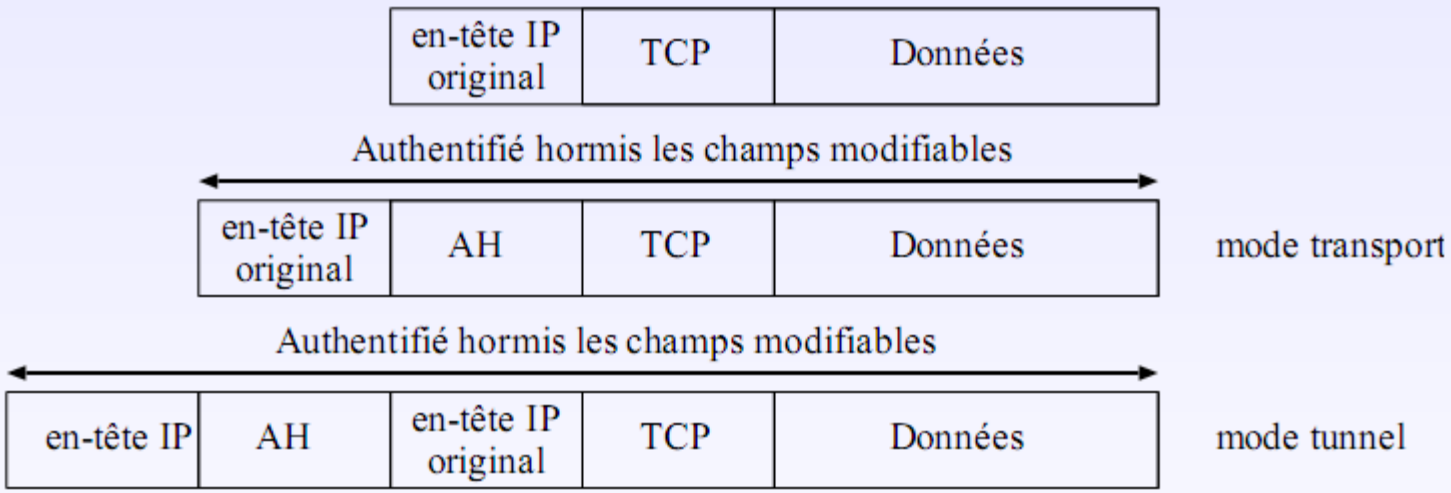


Figure: Portée de l'authentification AH.

Protocole IPsec

- ESP dans les deux modes

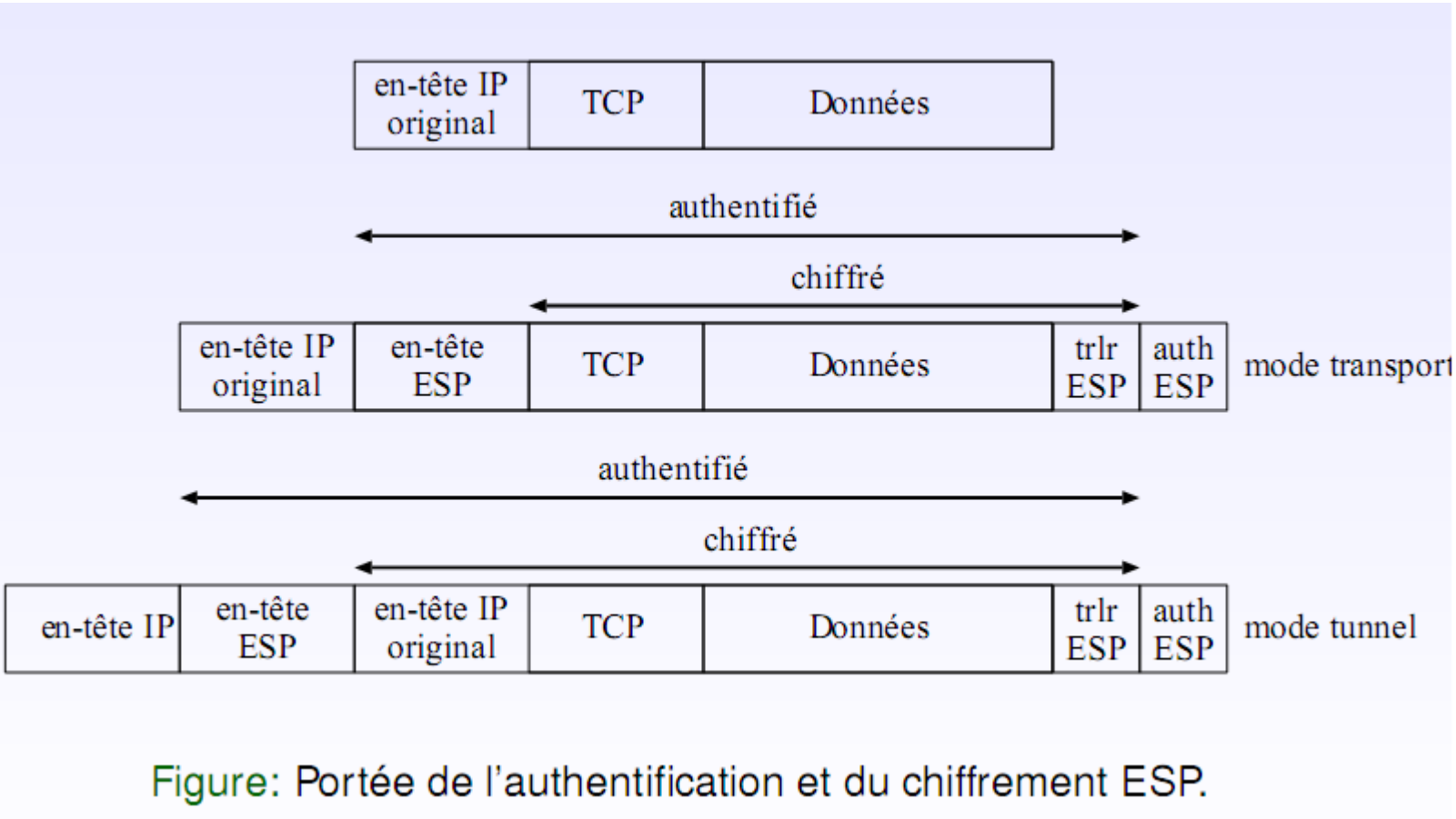
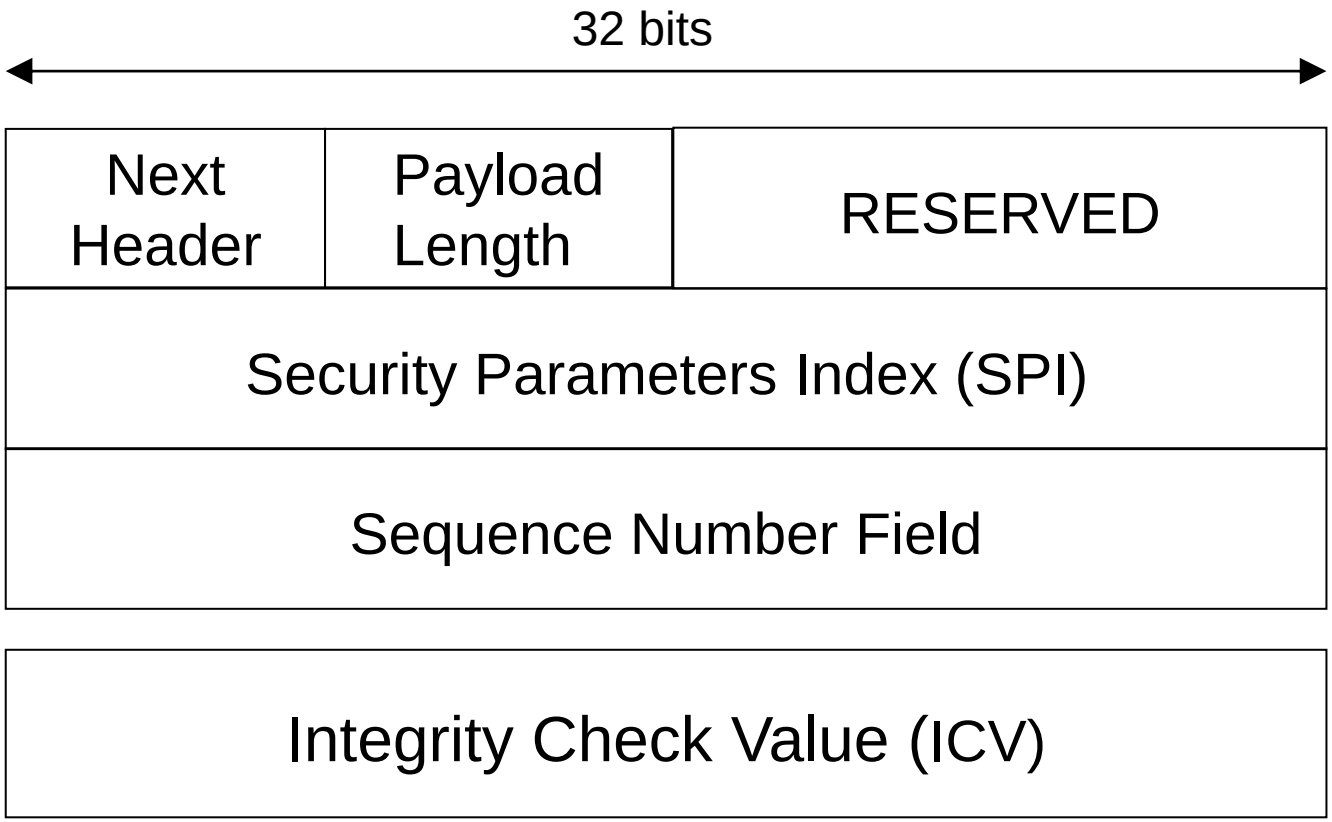


Figure: Portée de l'authentification et du chiffrement ESP.

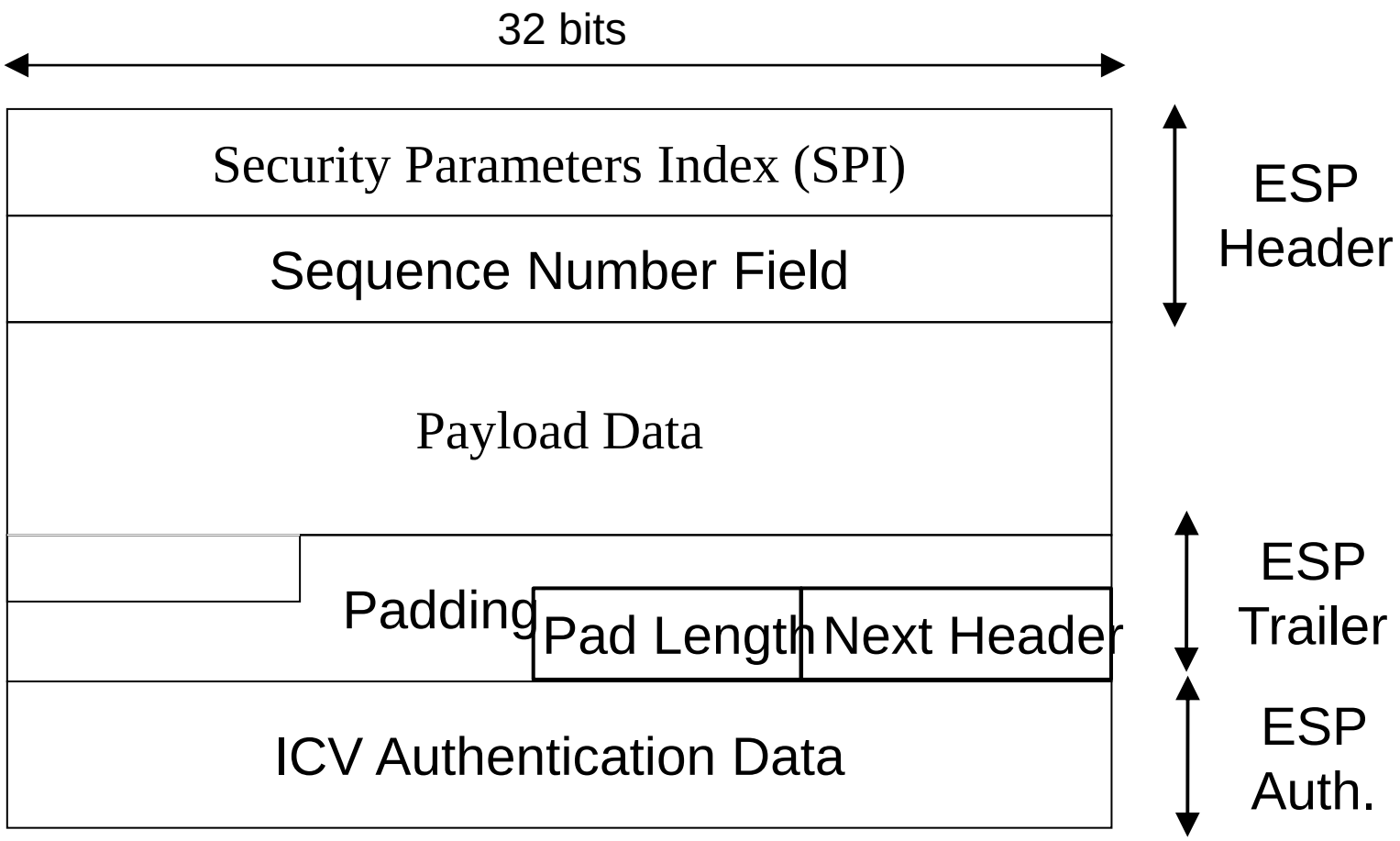
Protocole IPsec

AH : Athentication Header



SPI : Identification de l'association de sécurité SA à utiliser pour ce paquet
Sequence Number : un numéro de séquence obligatoire en émission. Vérifié en réception si le mécanisme anti-rejeu est activé
ICV Integrity Check Value : Contient le MAC (Message Authentication Code)

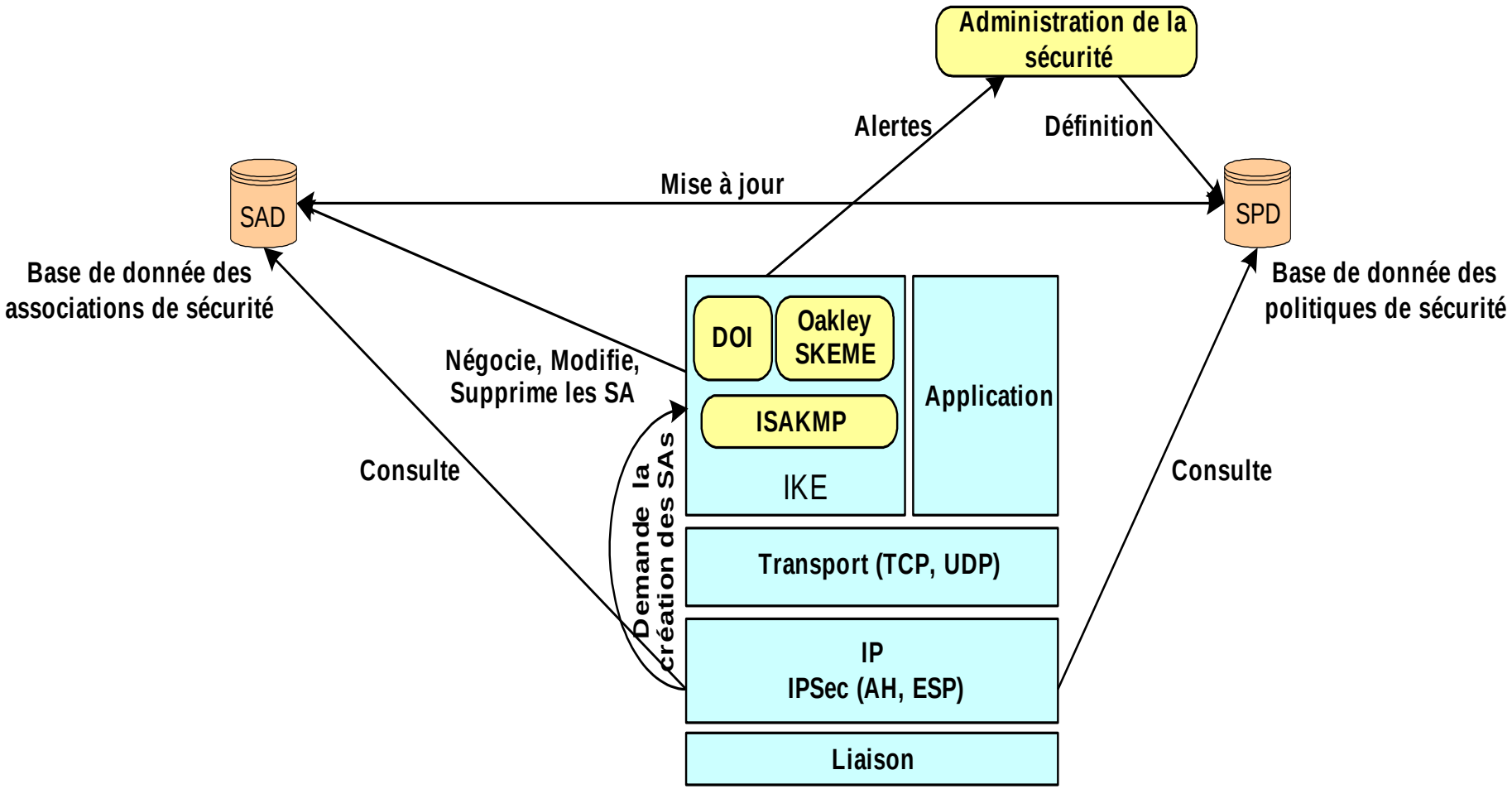
Protocole IPsec



Architecture générale VPN

- IPSec = IP security Protocol
  - Standard développé à l'IETF
  - Partie commune entre IPv4 et IPv6 (obligatoire en IPv6)
- Services de sécurité
  - Confidentialité, Intégrité, Authentification, non rejeu, contre analyse du trafic
- Protocoles
  - IKE (OAKLEY et ISAKMP)
  - AH (identifiant du protocole 51)
  - ESP (identifiant du protocole 50)
- Définitions de politiques de sécurité
- Etablissement d'association
- Méthodes d'authentification: PSK, Kerberos, Cerificat

Architecture générale VPN



Protocole d'échange de clés : IKE

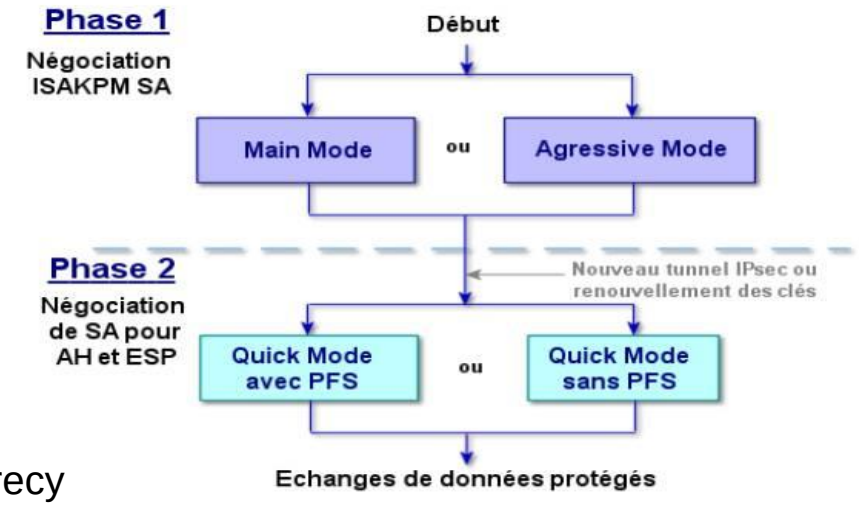
- Protocole IKE (Internet Key Exchange, RFC 2409)
  - IKE (RFC 2407, 2408, 2409, 2412)
  - Au niveau applicatif
  - Indépendant de IP
  - Utilise UDP sur le port 500
  - **Protocole pour la négociation** des paramètres des protocoles de sécurité
  - **Protocole d'authentification** des différentes entités
  - **Protocole de génération et d'échange** de clés secrètes
  - Autres services
    - Gestion dynamique des associations de sécurité
      - Création
      - Modification
      - Destruction (vs manuelle)
    - Secret parfait des clés (PFS Perfect Forward Secrecy) : des clés utilisées antérieurement ou postérieurement par le protocole ne peuvent être dérivées de la clé en cours d'utilisation

Protocole d'échange de clés : IKE

- Appliqué à IPSec, IKE a pour objectif d'établir
  - Un premier tunnel (le tunnel IKE, tunnel de service ou tunnel administratif) entre les deux entités
    - Ceci représente la phase 1 du protocole IKE
    - Ce tunnel ne sert pas à la transmission des données utilisateur
    - Son rôle : gérer les autres tunnels, leur création, le rafraîchissement des clés et autres services
  - Un tunnel secondaire (le tunnel de données ou IPsec tunnel)
    - Ceci représente la phase 2 du protocole IKE
    - On peut établir autant de tunnels secondaires que nécessaire pour la transmission des données

Protocole d'échange de clés : IKE

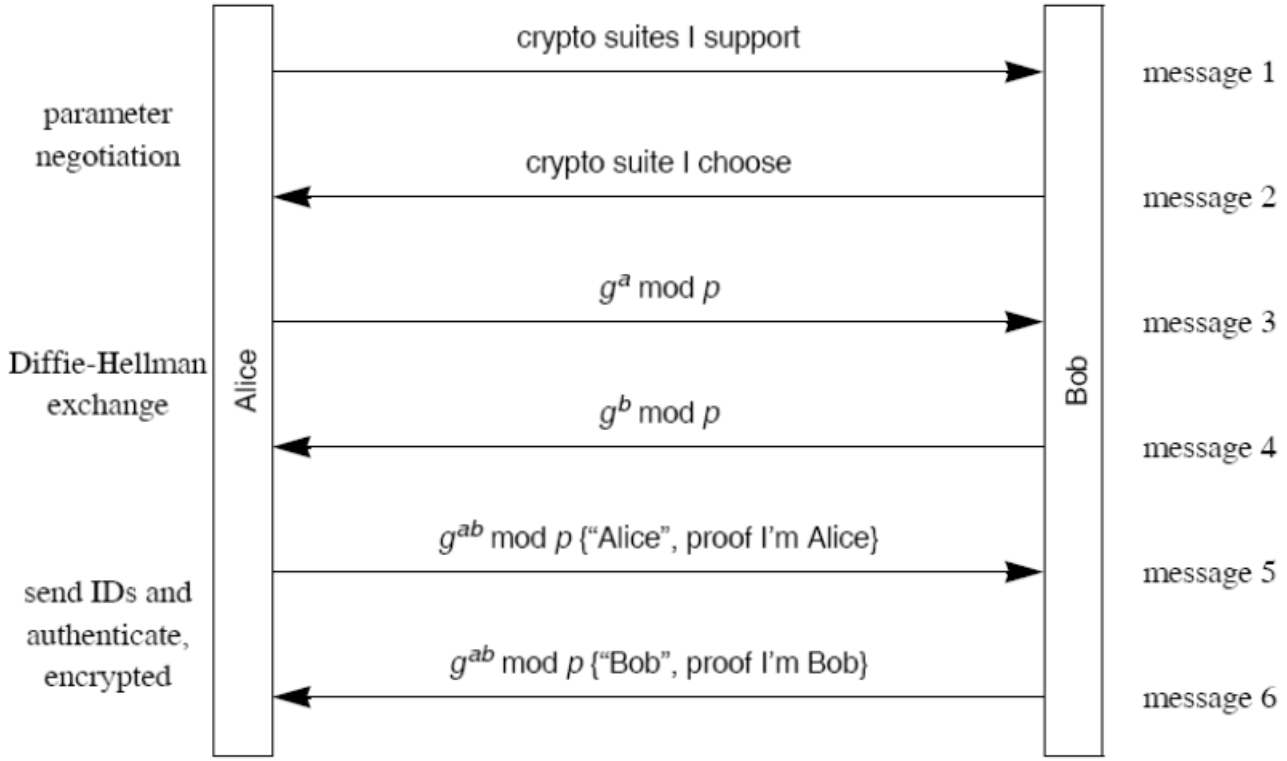
- Modes de IKE
  - Dans la phase1
    - Le mode principal (Main mode)
    - Le mode agressif (Aggressive Mode)
  - Dans la phase 2
    - Le mode rapide (Quick Mode)



PFS : Perfect Forward Secrecy

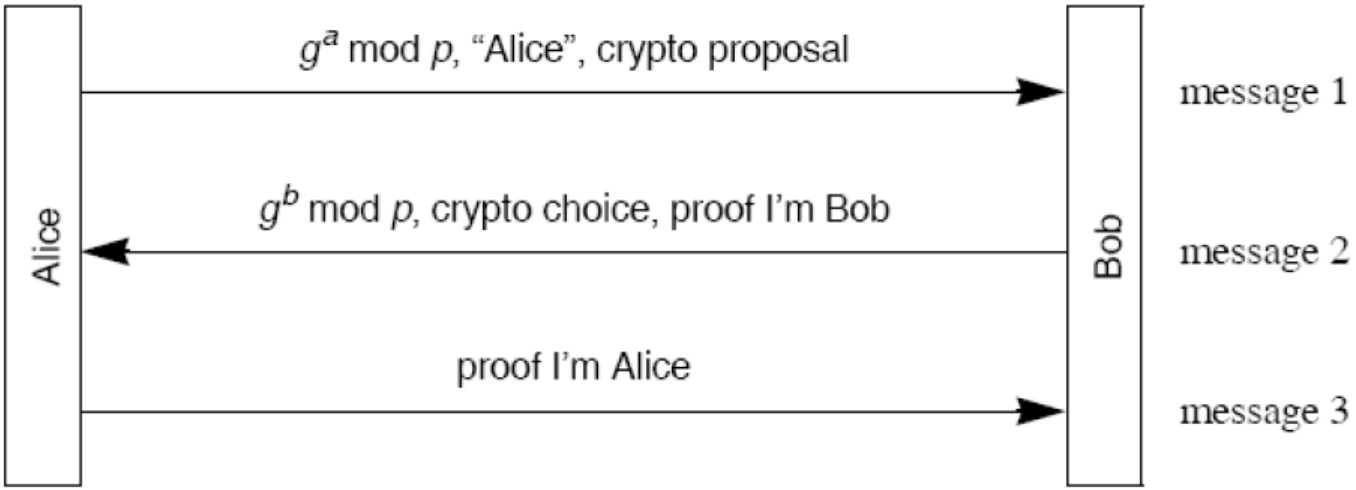
Protocole d'échange de clés : IKE

- IKE en main mode (6 messages)



Protocole d'échange de clés : IKE

- IKE en mode (3 messages)
  - Pas de protection des identités



Protocole d'échange de clés : IKE

- Services IKE :
  - Partage de clés par DH
    - $A \rightarrow B: g^a$
    - $B \rightarrow A: g^b$
  - Authentification des entités
    - $A \rightarrow B: m, A$
    - $B \rightarrow A: n, sig_B(m, n, A)$
    - $A \rightarrow B: sig_A(m, n, B)$
  - Protection des identités (chiffrement de la signature)
    - $A \rightarrow B: g^a, A$
    - $B \rightarrow A: g^b, Enc_K(sig_B(g^a, g^b, A))$
    - $A \rightarrow B: Enc_K(sig_A(g^a, g^b, B))$
- IKE modes
  - Main Mode
  - Agressive mode

Protocole d'échange de clés : IKE

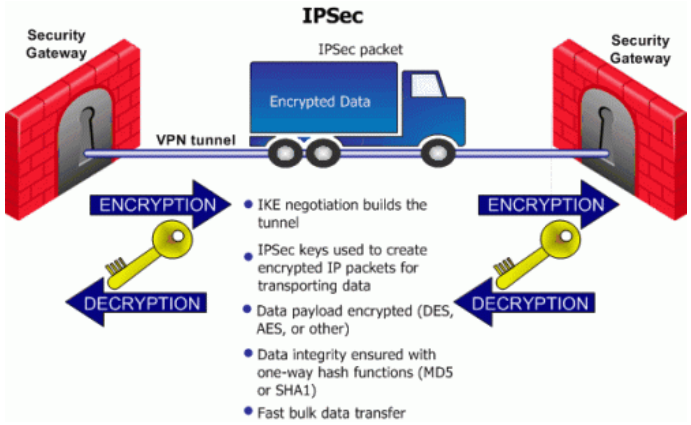
- Comment les deux extrémités se mettent d'accord sur les différents paramètres ?
  - Paramètres négociés via le protocole ISAKMP
  - Mise en place d'associations de sécurité SA (Security association)
    - Ensemble de règles et de clés utilisées pour protéger des informations
    - Services de sécurité à activer, clés à utiliser par AH, ESP et IKE (Internet Key Exchange)
    - Configuration manuelle
    - Négociation automatique (IKEv2 RFC 7296)

Protocole d'échange de clés : IKE

- SA (Security Association)
  - Ensemble de politiques et de clés utilisés pour protéger l'information
  - Connexion qui fournit des services de sécurité au trafic
  - Structure de données servant à stocker l'ensemble des paramètres associés à une communication donnée

- Plusieurs SA sont utiles pour « monter » un tunnel IPSec
  - Il faut une SA pour IKE (tunnel de service, on dit aussi SA ISAKMP)
  - Il faut une SA pour IPSec (tunnel de données)

- Deux types :
  - SAs ISAKMP
  - SAs IPSec



Protocole d'échange de clés : IKE

- IKE phase 1 (appelée ISAKMP-SA)
  - Permet de configurer les algorithmes de hachage, algorithmes de chiffrement et méthode d'authentification
  - Association de sécurité bidirectionnelle
- IKE phase 2 (appelée IPsec-SA)
  - IPsec-SA protégé par le tunnel ISAKMP établi en phase 1.
  - 2 associations unidirectionnelles entre les mêmes extrémités
  - Paramètres négociés
    - Les mêmes que lors de la phase 1 (généralement) algorithme de hachage, algorithme de chiffrement, groupe Diffie Hellman, informations anti rejeu, durée de vie de l'association de sécurité

Protocole d'échange de clés : IKE

- Génération des clés
  - Etablir la clé mère initiale (SKEYID) en se basant sur
    - Un secret partagé appelé pre-shared secret key (PSK)
    - Le chiffrement asymétrique pour échanger un secret
    - Diffie-Hellman ou ECDHE (mode signature) pour échanger le secret mais en signant les échanges par la clé publique (mode signature)
  - SKEYID = hash (DH values, nonces, cookies, PSK si partagé) ⇒ Key seeds

Protocole d'échange de clés : IKE

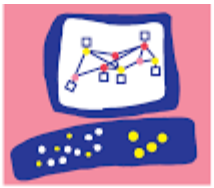
- SKEYID-a : utilisée pour l'authentification
- SKEYID-e : utilisée pour le chiffrement
- SKEYID-d : utilisée comme clé mère des autres associations de sécurité (la phase 2)

- SKEYID_d = prf(SKEYID, (g^{ab}mod p| cookies | 0))
- SKEYID_a = prf(SKEYID, (SKEYID_d | (g^{ab}mod p | cookies | 1))
- SKEYID_e = prf(SKEYID, (SKEYID_a | (g^{ab}mod p | cookies | 2))

**PRF*: fonction pseudo-aléatoire (pseudorandom function*)

Principaux acteurs VPN

- Principaux acteurs des équipements VPN
  - Cisco
  - Juniper
  - Checkpoint
  - Fortinet



Check Point
SOFTWARE TECHNOLOGIES LTD

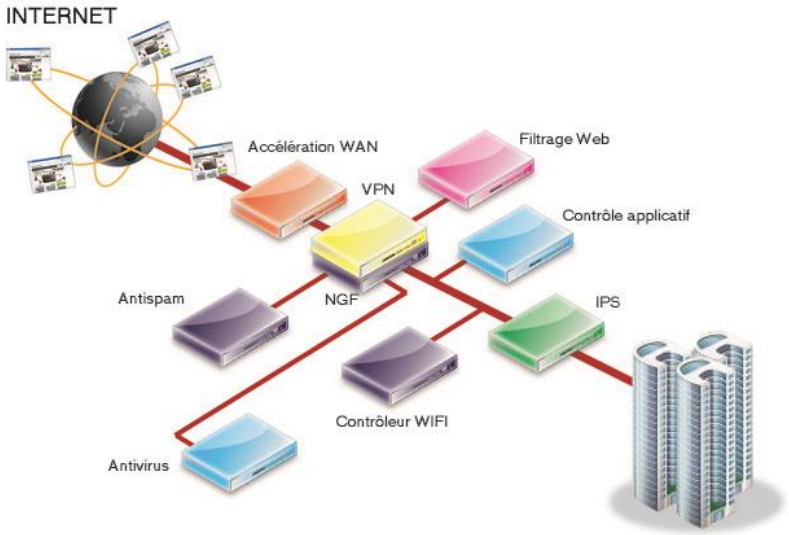
Fortinet

Technologies et solutions

- Unified Threath Management ou UTM)

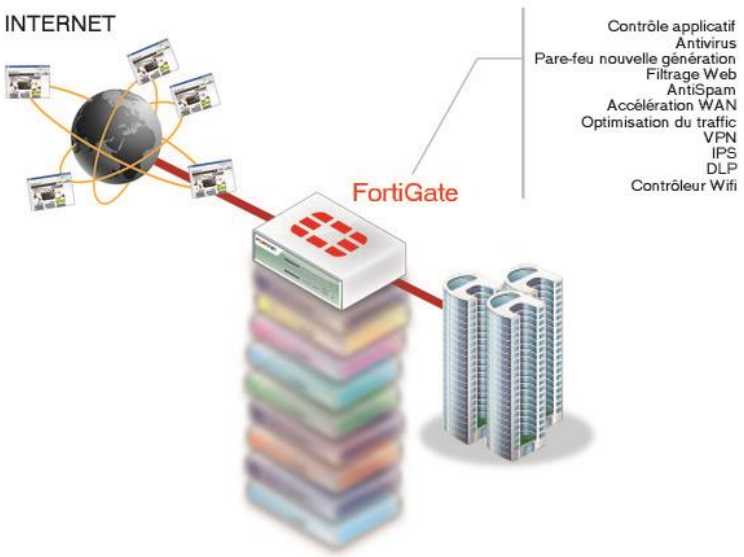
SOLUTIONS TRADITIONNELLES

Une prolifération d'appliances pour une infrastructe complexe et coûteuse



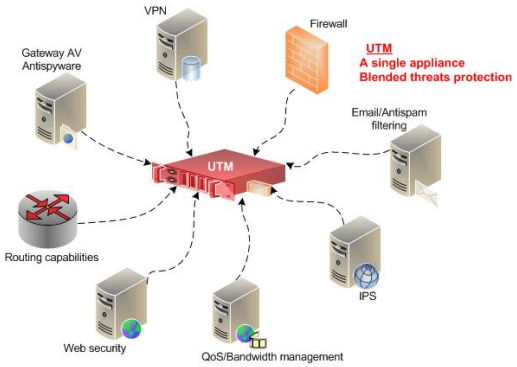
L'UTM SELON FORTINET

Une sécurité consolidée, simple et économique



Technologies et solutions

- UTM (Unified Threat Management)
  - Systèmes de gestion unifiée des menaces
  - Outils les plus couramment utilisés dans la sécurité
  - Concept à la mode
  - Proposent de nombreuses technologies de sécurité
    - Intégrées sur une seule plate-forme
    - Fournies par un seul éditeur
    - Une seule interface d'administration



Les produits Fortigate

- Gestion unifiée des menaces (UTM)
  - Terme de sécurité proposé en 2004
  - Proposé par Charles Kolodgy du cabinet de conseil IDC (International Data Corporation)
  - Solution basée sur des pare-feu avec des fonctionnalités supplémentaires par rapport aux pare-feux traditionnels
- Leaders d'UTM sur le marché
  - Fortinet, Symantec, Juniper, WatchGuard, Kerio, Checkpoint

Technologies et solutions

- Avantages d'une solution UTM
  - Déploiement simple
    - Beaucoup moins d'étapes au niveau de l'installation et de la configuration
  - Administration facile
    - Une seule console d'administration
    - Un seul processus de mise à jour
  - Résolution plus rapide des problèmes
    - Moins de possibilités de conflits entre les modules
    - Support assuré par un seul et même éditeur
  - Logs homogènes avec des corrélations utiles entre les différents types de données.

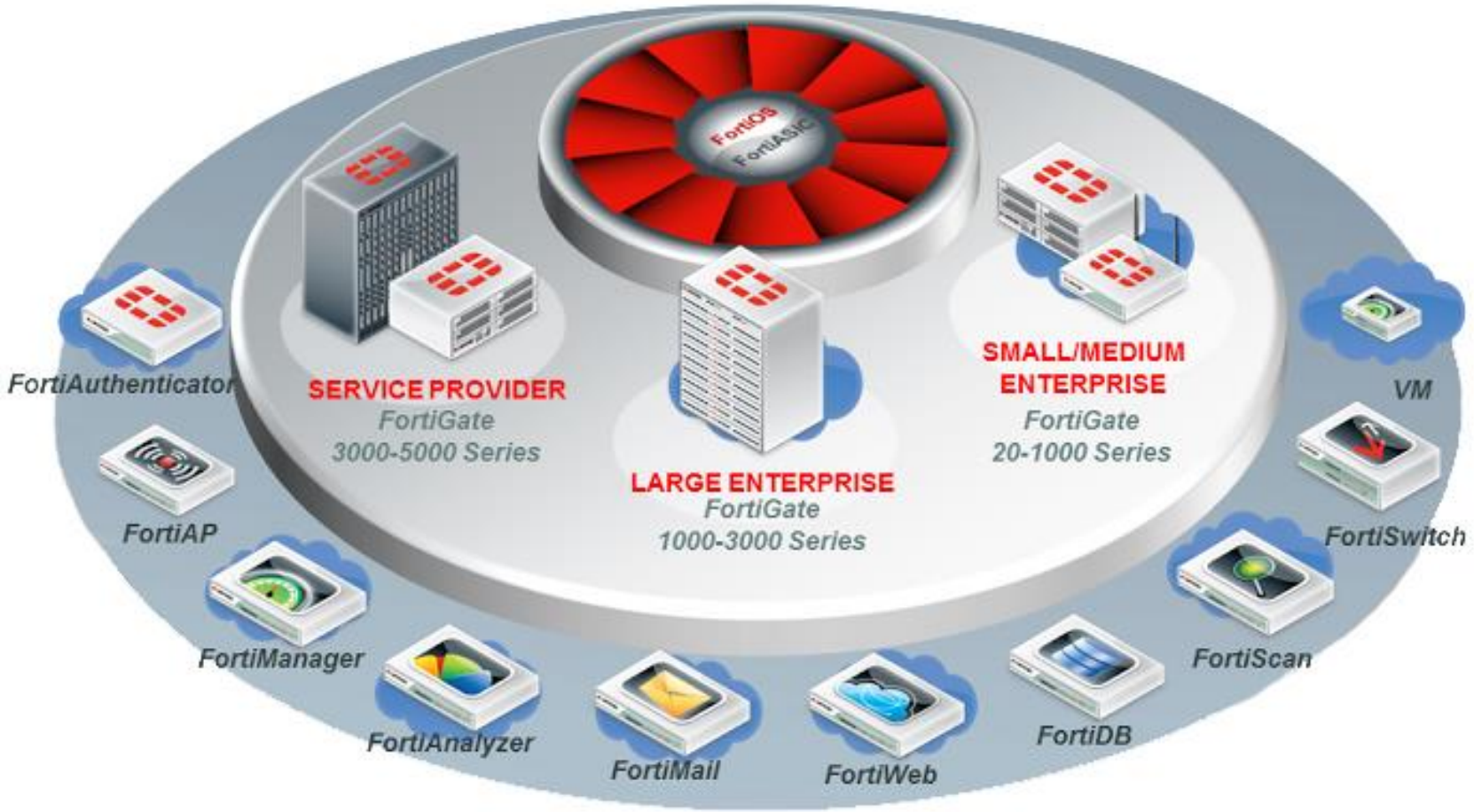
Fortinet

- Fortinet
  - Société américaine créee en 2000
  - Siège social : Sunnyvale, CA, États-Unis
  - Fondateurs : Ken Xie, Michael Xie
  - Leader mondiale pour la technologie UTM
  - Solutions de sécurité pour les entreprises
    - Grandes
    - Moyennes
    - Petites
    - Opérateur
  - Son produit Fortigate est parmi les premières solutions dans ce domaine
  - Fortigate => all in one solution
  - Produits : FortiGate 5000, 3000, 1000 Series

Les produits Fortigate

- FortiGate
  - Appliances de sécurité intégrant différentes fonctionnalités de sécurité aussi bien réseau qu'applicatif.
  - Appliances extrêmement performantes du fait des processeurs propriétaires FortiASIC.
  - FortiOS compile l'ensemble des fonctionnalités en un seul système

Les produits Fortinet

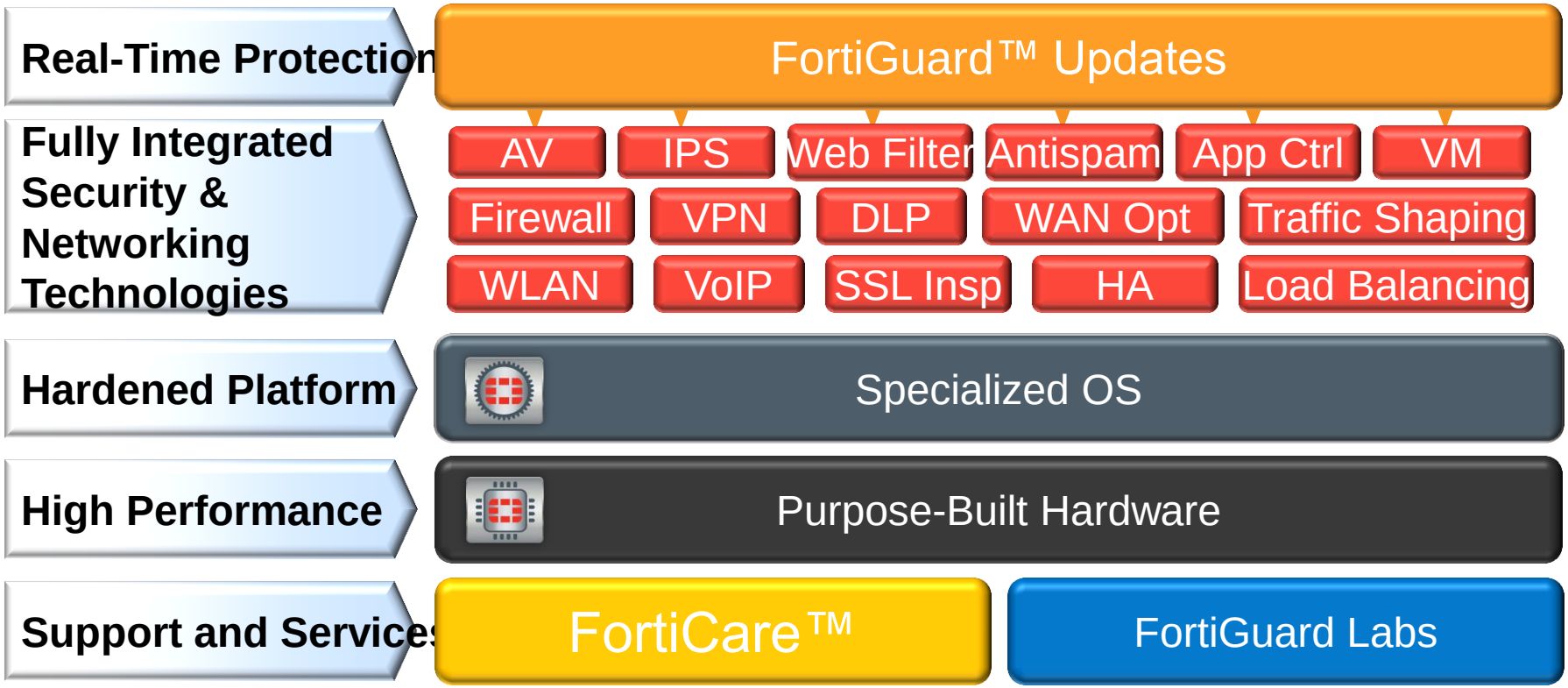


Les produits Fortinet

- *FortiGate*
  - Boitiers UTM (pare-feu, Antivirus, IPS, VPN (IPSec et SSL), filtrage Web, Antispam, VDOM, compression de données, routage
- *FortiMail*
  - Boitiers de sécurisation de la messagerie électronique par les techniques Antivirus et Antispam
- *FortiAnalyzer*
  - Boitier pour centraliser les journaux des équipements Fortinet (FortiGate, FortiMail, FortiManager et FortiClient)
- *FortiManager*
  - Boitiers permettant la supervision et l'administration centralisée des équipements Fortinet (FortiGate, FortiAnalyzer et FortiClient)
- *FortiClient*
  - Client VPN IPSec (PC windows et smartphone)
  - Protège les équipements contre virus, intrusions, spams, spywares
- *FortiWeb*
  - Boitier spécialisé pour les applications Web et XML
  - Permettant le *load-balancing* entre plusieurs serveurs

Les produits Fortigate

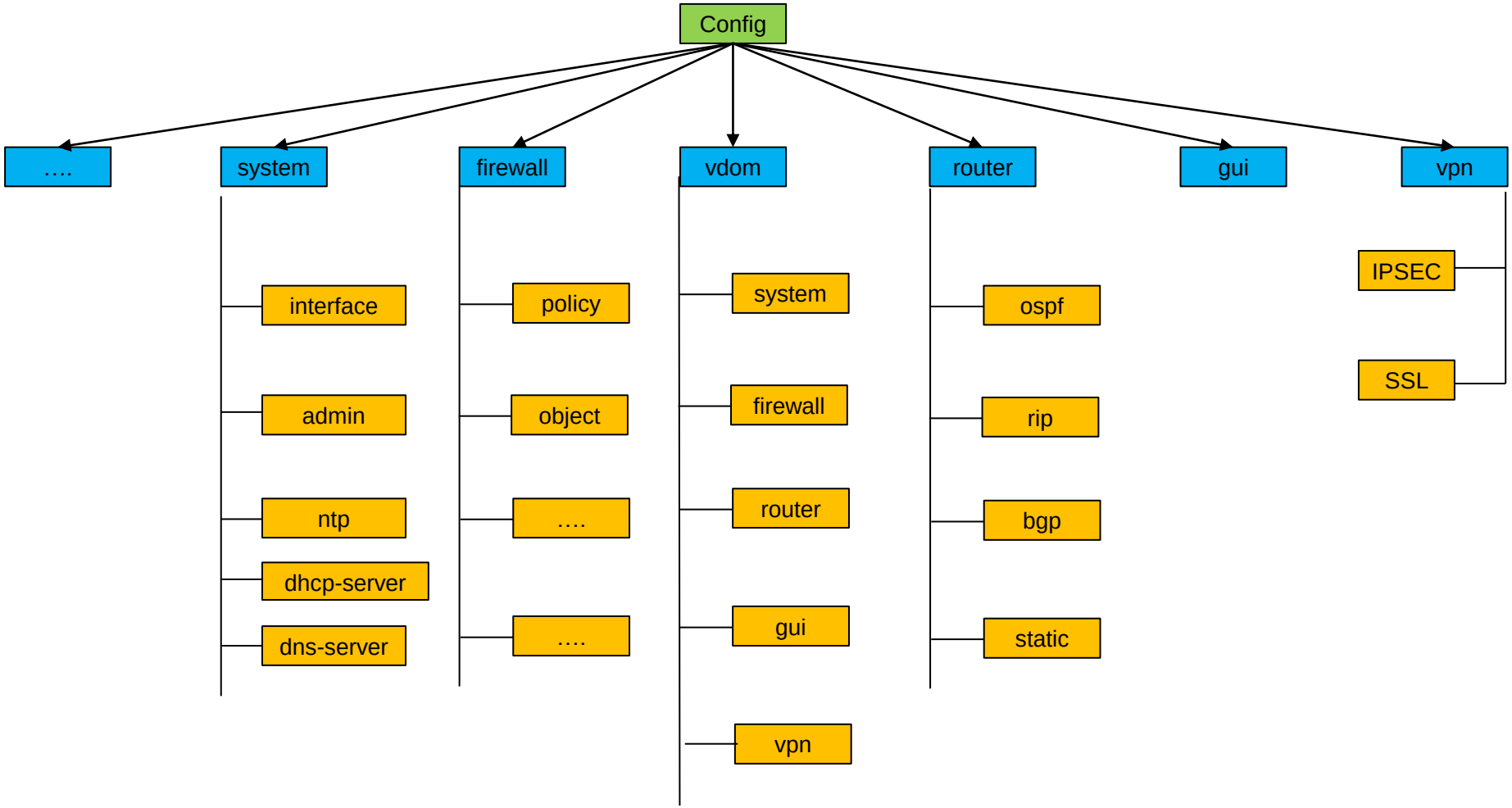
Architecture intégrée



- Purpose-built to deliver overlapping, complementary security
- Provides both flexibility & defense-in-depth capabilities

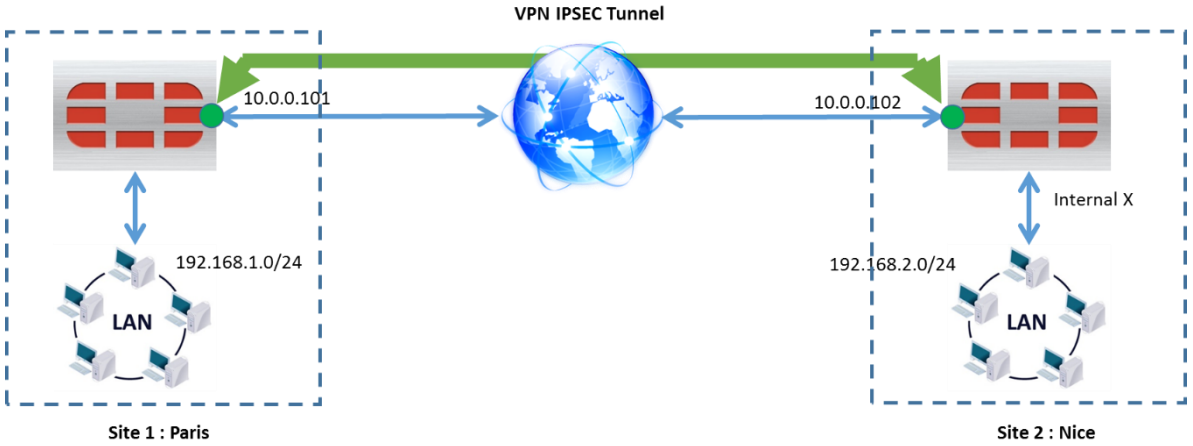
Administration et configuration : CLI

Arborescence



Exemple d'architecture VPN site-to-site

- Etapes de construction d'un tunnel
  - Phase1
  - Phase2



- Local interface 10, IP gateway 10.0.0.101
- Local interface 11, IP gateway 10.0.0.102
- Ajouter une route statique pour traverser par le tunnel :
  - Destination: 192.168.2.0/255.255.255.0, device « nom tunnel crée»
- Configurer les deux phases
- Ajouter les règles du Firewall

Configuration tunnel VPN

- Etapes de construction du tunnel

```
config vdom
            edit Router1
config vpn ipsec phase1-interface
    edit "Phase1-1"
        set interface "10"
        set proposal 3des-sha1 aes128-sha1
        set remote-gw 10.10.10.2
        set psksecret 123456
    next
end
config vpn ipsec phase2-interface
    edit "phase1-2"
        set phase1name "Phase1-1"
        set proposal 3des-sha1 aes128-sha1
    next
end
```

```
config vdom
            edit Router2

config vpn ipsec phase1-interface
    edit "Phase1-2"
        set interface "11"
        set proposal 3des-sha1 aes128-sha1
        set remote-gw 10.10.10.1
        set psksecret 123456
    next
end
config vpn ipsec phase2-interface
    edit "phase2-2"
        set phase1name "Phase1-2"
        set proposal 3des-sha1 aes128-sha1
    next
end
```

Configuration tunnel VPN

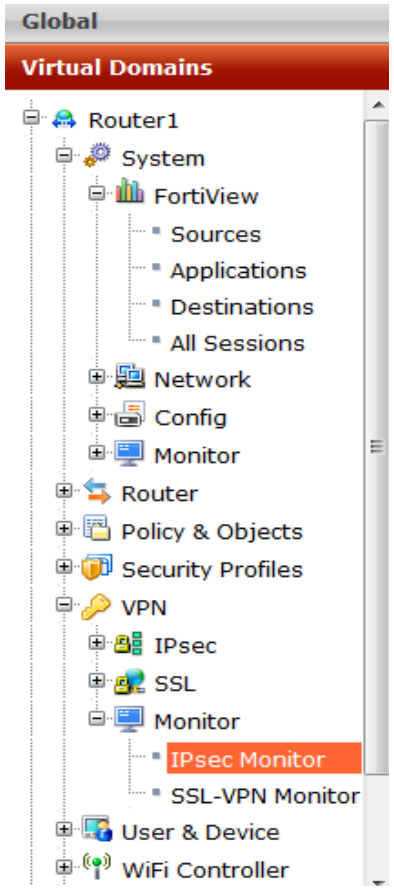
- Ajouter une route statique pour traverser par le tunnel

```
config vdom
            edit Router1
config router static
    edit 1
        set device "wan1"
    next
    edit 2
        set device "10"
        set dst 192.168.2.0 255.255.255.0
        set gateway 10.10.10.2
    next
    edit 3
        set device “Nom du tunnel"
        set dst 192.168.2.0 255.255.255.0
    next
end
```

```
config vdom
            edit Router2
config router static
    edit 1
        set device "11"
        set gateway 10.10.10.1
    next
    edit 2
        set device "11"
        set dst 192.168.1.0 255.255.255.0
        set gateway 10.10.10.1
    next
    edit 3
        set device “Non du tunnel"
        set dst 192.168.1.0 255.255.255.0
    next
end
```

Configuration tunnel VPN

- Etat du tunnel
  - Si le pre-shared key n'est pas le même -> Down



Name	Type	Remote Gateway	Username	Status	Incoming Data	Outgoing Data	Phase 2 Proposal
Phase1-1	Static IP or Dynamic DNS	10.10.10.2		Up			Phase1-2

Changer le pre-shared key et vérifier l'état du tunnel

SSL/TLS

SSL/TLS

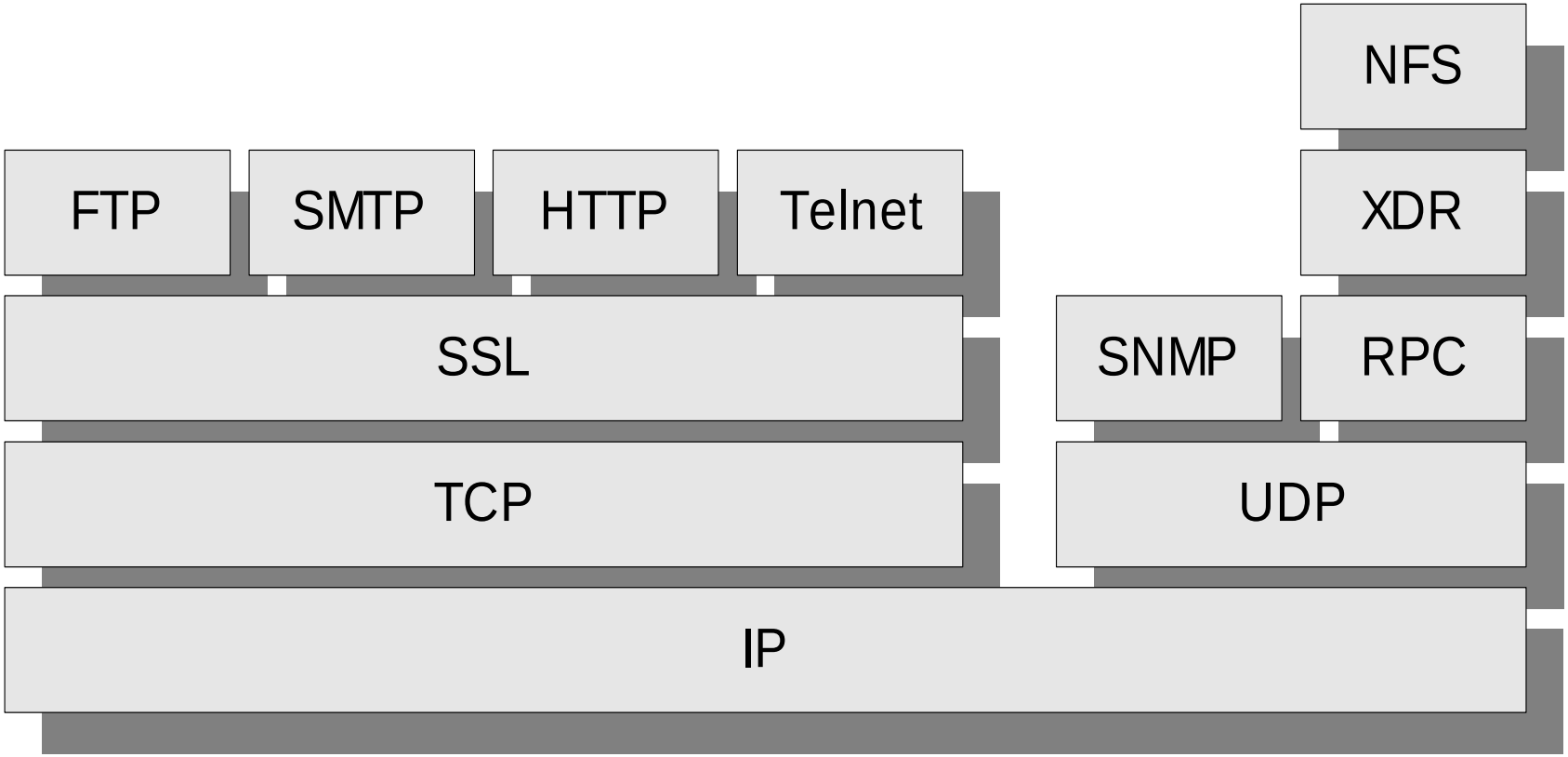
- SSL proposé par Netscape* et intégré au browser
  - 1994 : Première version de SSL testé en interne (pas déployée)
  - 1994 : Première version de SSL diffusé : SSL 2.0
  - 1996 : SSL 3.0, version stable du protocole, a été soumise à l'IETF pour une standardisation formelle
  - TLS est la nouvelle version de SSL 3.0, reprise par l'IETF

- IETF au sein du groupe Transport Layer Security
  - RFC2246 : première version 1.0 publiée par IETF en 1999
  - RFC2817 : addition de Kerberos** à TLS, 2000
  - RFC2818 : HTTP sur TLS, 2000
  - RFC3268 : utilisation d'AES pour TLS, 2002
  - RFC4346 : TLS version 1.1, 2006

** Netscape Navigator : navigateur ayant dominé le marché au milieu des années 1990*

*** Kerberos : protocole d'authentification : tickets au lieu de mots de passe*

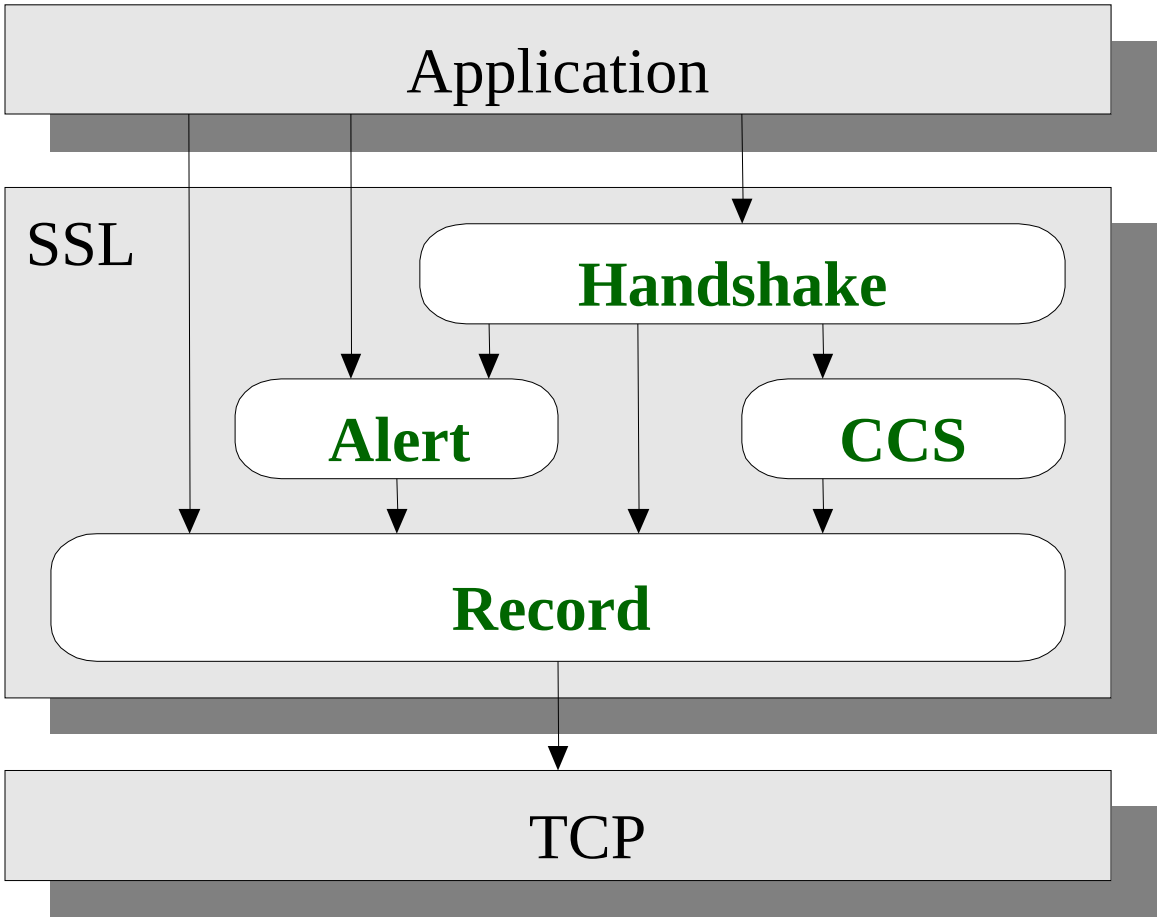
SSL/TLS



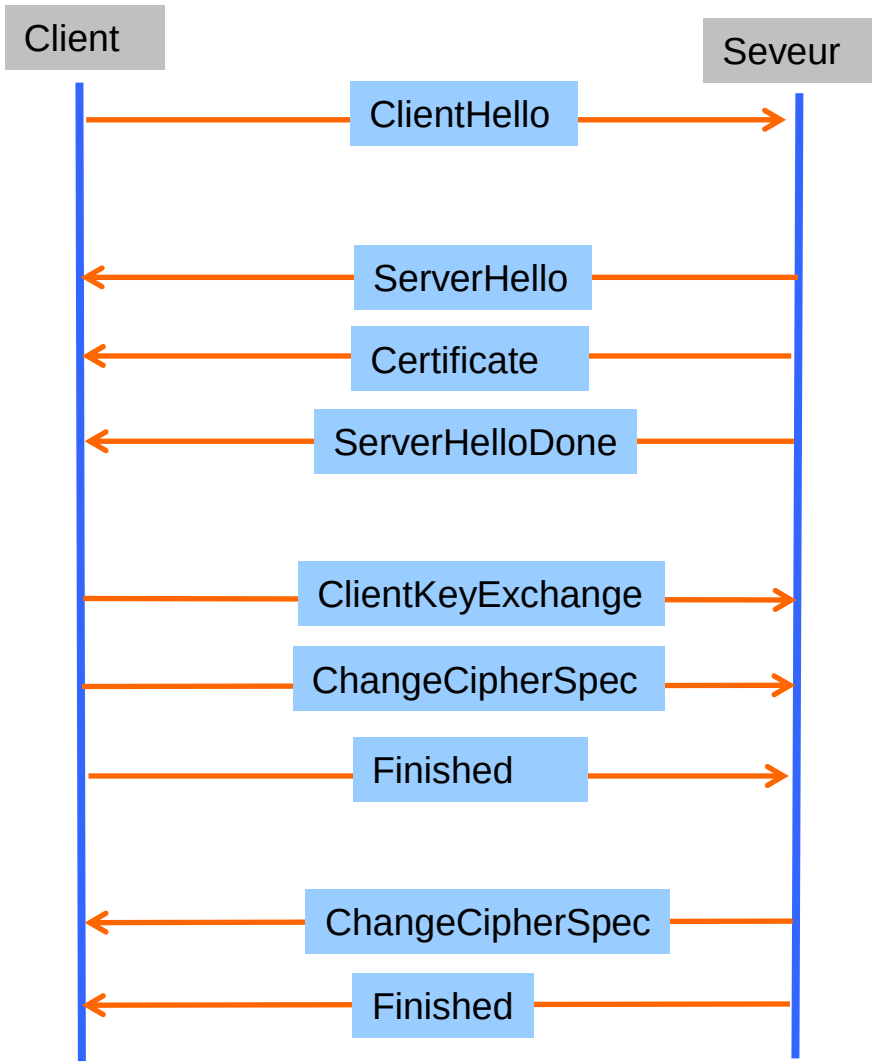
SSL/TLS

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

SSL/TLS



SSL/TLS



- *ClientHello* : ce message contient la version de SSL, un nombre aléatoire permettant de générer les clés secrètes, et l'ensemble d'algorithmes proposés : DH, RSA, 3DES
- *Server Hello* : dans ce message le serveur choisit les algorithmes proposés par le client et choisit la longueur de clés. Ainsi, il génère un nombre aléatoire permettant de générer les clés secrètes
- *Certificate* : le navigateur du client vérifie immédiatement la validité du certificat
- *ServerHelloDone : Ok => à ce moment le client peut vérifier le certificat du serveur et échanger les clés*
- *ClientKeyExchange :* le client génère la clé pre-master et le chiffre par la clé publique du serveur. Enfin, il envoie le message chiffré au serveur
- *ChangeCipherSpec* : le client informe le serveur que tous les messages suivants vont être chiffrés par la clé symétrique échangée dans le message d'avant
- *Finished* : le client envoie un message chiffré par la nouvelle clé pour vérification
- Le serveur fait la même procédure pour vérifier la bonne utilisation de la clé

SSL/TLS

Les solutions de sécurité exemple d'une requête ClientHello

```
⊞ Frame 740 (217 bytes on wire, 217 bytes captured)
⊞ Ethernet II, Src: Dell_2b:76:54 (00:1e:c9:2b:76:54), Dst
⊞ Internet Protocol, Src: 10.10.1.37 (10.10.1.37), Dst: 62
⊞ Transmission Control Protocol, Src Port: 55538 (55538),
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 158
    ⊟ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 154
        Version: TLS 1.0 (0x0301)
      ⊞ Random
        Session ID Length: 0
        Cipher Suites Length: 68
      ⊞ Cipher Suites (34 suites)
        Compression Methods Length: 1
      ⊞ Compression Methods (1 method)
        Extensions Length: 45
      ⊞ Extension: server_name
      ⊞ Extension: elliptic_curves
      ⊞ Extension: ec_point_formats
      ⊞ Extension: SessionTicket TLS
```

Algorithmes proposés par le client

```
⊟ Cipher Suites (34 suites)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
    Cipher Suite: TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x0088)
    Cipher Suite: TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA (0x0087)
    Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
    Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038)
    Cipher Suite: TLS_ECDH_RSA_WITH_AES_256_CBC_SHA (0xc00f)
    Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (0xc005)
    Cipher Suite: TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x0084)
    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_RC4_128_SHA (0xc007)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
    Cipher Suite: TLS_ECDHE_RSA_WITH_RC4_128_SHA (0xc011)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
    Cipher Suite: TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (0x0045)
    Cipher Suite: TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA (0x0044)
    Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
    Cipher Suite: TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x0032)
    Cipher Suite: TLS_ECDH_RSA_WITH_RC4_128_SHA (0xc00c)
    Cipher Suite: TLS_ECDH_RSA_WITH_AES_128_CBC_SHA (0xc00e)
    Cipher Suite: TLS_ECDH_ECDSA_WITH_RC4_128_SHA (0xc002)
    Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (0xc004)
    Cipher Suite: TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (0x0041)
    Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
    Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
    Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA (0xc008)
    Cipher Suite: TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xc012)
    Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x0016)
    Cipher Suite: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x0013)
    Cipher Suite: TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA (0xc00d)
    Cipher Suite: TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA (0xc003)
    Cipher Suite: SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA (0xfeff)
    Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
  Compression Methods Length: 1
```

SSL/TLS

réponse *ServerHello*

```
741 17.003003   62.161.94.179        10.10.1.37          TLSv1    Server Hello, Certificate, Server Hello Done
⊞ Frame 741 (1058 bytes on wire, 1058 bytes captured)
⊞ Ethernet II, Src: Cisco_d2:49:3f (00:1f:6c:d2:49:3f), Dst: Dell_2b:76:54 (00:1e:c9:2b:76:54)
⊞ Internet Protocol, Src: 62.161.94.179 (62.161.94.179), Dst: 10.10.1.37 (10.10.1.37)
⊞ Transmission Control Protocol, Src Port: https (443), Dst Port: 55538 (55538), Seq: 1, Ack: 164, Len: 1004
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 999
    ⊟ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 70
        Version: TLS 1.0 (0x0301)
      ⊞ Random
        Session ID Length: 32
        Session ID: 08010000EE91A59A714BD60A7F42FCA1FFE867C1207CCFE9...
        Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
        Compression Method: null (0)
    ⊟ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 917
        Certificates Length: 914
      ⊞ Certificates (914 bytes)
    ⊟ Handshake Protocol: Server Hello Done
        Handshake Type: Server Hello Done (14)
        Length: 0
```

Certificat

- Numéro de série du certificat
- Spécificités techniques du certificat : format, algorithmes utilisés pour la signature
- Dates de validité du certificat
- Objet d'utilisation

- Nom, adresse, e-mail…du propriétaire du certificat
- Clé publique du propriétaire

- Nom du tiers de confiance, identifiant du tiers de confiance, adresse,…

hachage

Clé privée du tiers de confiance

Signature du tiers de confiance

Certificat= $Nom_{Serveur}$ + Nom_{AC} + $K_{P\text{-}SERV}$ + $K_{S\text{-}AC}$ {$Nom_{Serveur}$ + Nom_{AC} + $K_{P\text{-}SERV}$ }

Protocole SSL/TLS – Génération des clés

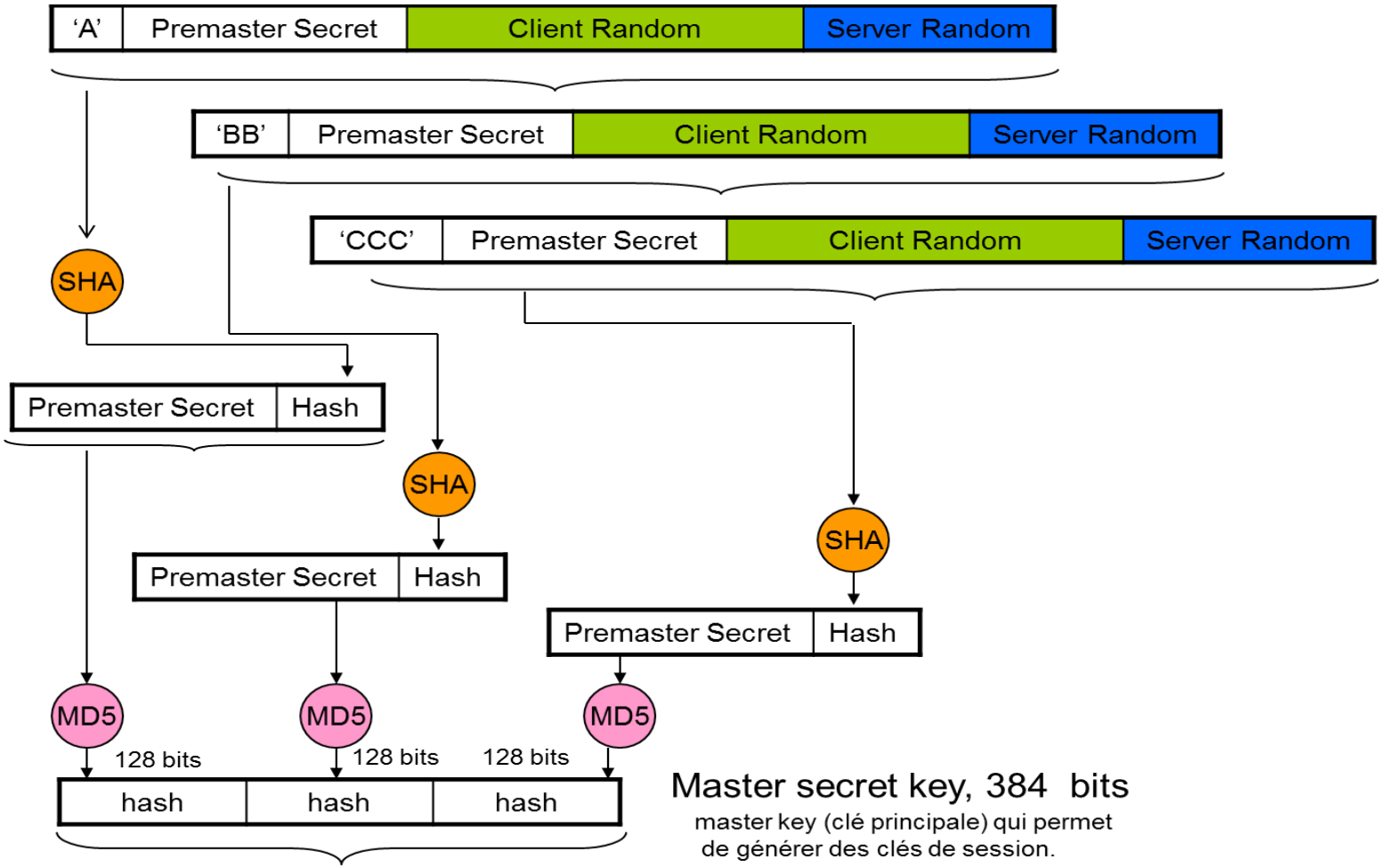
- Construction du *Master secret key* à l'ouverture d'une session
  - Calculé par le client et le serveur
  - master_secret =

    MD5(pre_master_secret || SHA('A' || pre_master_secret || ClientHello.random || ServerHello.random))||
    MD5(pre_master_secret || SHA('BB' || pre_master_secret || ClientHello.random ||ServerHello.random)) ||
    MD5(pre_master_secret || SHA('CCC' || pre_master_secret || ClientHello.random || ServerHello.random))

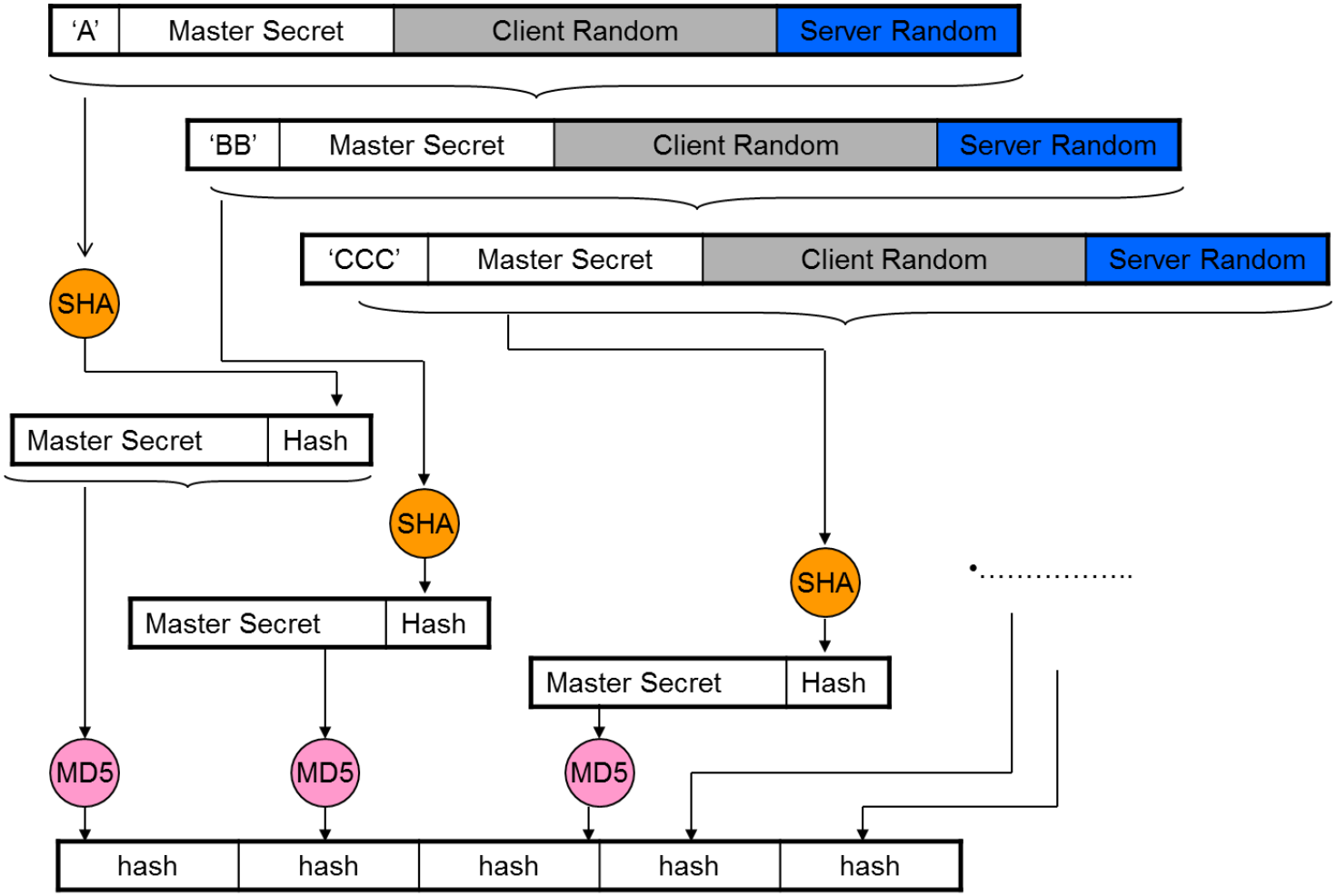
- Génération de secrets à l'ouverture d'une session ou connexion
  - key_block =

    MD5(master_secret || SHA('A' || master_secret || ServerHello.random || ClientHello.random))||
    MD5(master_secret || SHA('BB' || master_secret || ServerHello.random || ClientHello.random))||
    MD5(master_secret || SHA('CCC' || master_secret || ServerHello.random || ClientHello.random))||
  - Key_block= 2 clés MAC + 2 clés chiffrement

Protocole SSL/TLS – Génération des clés



Protocole SSL/TLS – Génération des clés



VPN - SSL

- SSL VPN (Secure Sockets Layer Virtual Private Network)
  - Tunnel au dessus de SSL.
  - Permet aux utilisateurs d'établir une connexion sécurisée au réseau intranet depuis n'importe quel navigateur Web