ÉCOLE PRATIQUE DES HAUTES ETUDES COMMERCIALES

Avenue du Ciseau, 15
1348 Louvain-la-Neuve

Comparatif des solutions d'accès à distance aux ressources informatiques

---

Travail de fin d'études présenté en vue de l'obtention du diplôme de bachelier en Informatique et Systèmes : finalité Technologie de l'informatique

Auteur :
Christian JUCKLER 3TL1
Rapporteur :
Virginie Van den Schriek

Année Académique 2014-2015

Résumé

Test

Table des matières

Liste des tableaux

Table des figures

Introduction

Un VPN est un réseau privé construit sur une infrastructure publique. Il permet à une entreprise de connecter de façon sécuriser des sites et des utilisateurs distants. De nos jours, l'infrastructure publique utilisée est le réseau Internet. La sécurité des VPN est un enjeu essentiel pour éviter la fuite d'information sensible. Cette sécurité est garantie par l'utilisation de tunnel VPN IPSec et SSL.

Première partie

La problématique des accès distants en entreprise

Chapitre 1

Les besoins habituels

Chapitre 2

Les besoins spécifiques

Chapitre 3

La situation actuelle

Chapitre 4

Méthodologie et objectifs

Pour réaliser ce TFE, j'ai commencé par m'approprier les connaissances théoriques nécessaire à la création d'un tunnel VPN. C'est-à-dire que j'ai étudié l'ensemble des protocoles utilisés, en me focalisant sur les principaux à savoir IPSec, SSL et TLS.

Cette première analyse théorique m'a permis de cerner les difficultés à créer un tunnel sécurisé.

En plus des technologies, j'ai lu les guides d'administration des principales appliances que j'allais utiliser. Mais avant de les configurer pour créer les tunnels VPN, je les ai testé pour m'assurer du fonctionnement.

Après m'être familiarisé avec le matériel, il a fallu le configurer et l'installer pour créer une infrastructure ressemblant à celle d'une entreprise.

Finalement, j'ai réalisé les tests.

Deuxième partie

Technologies d'accès distant

Chapitre 5

Description des technologies

Dans ce chapitre, je présente les technologies d'accès distants. Dans un premier temps, je commence par expliquer les notions des VPN et de tunnel VPN. De plus, j'y expose les protocoles de tunneling et de chiffrement. Par la suite, je détaille les protocoles qui peuvent former des tunnels VPN sécurisés. Finalement, je passe en revue les solutions propriétaires et celles des fournisseurs d'accès pour les accès distants.

5.1 Virtual Private Network - VPN

Le terme VPN est un acronyme pour « Virtual Private Network ». Un VPN est par définition un réseau virtuel qui transfère des données privées en créant un tunnel à travers un réseau public.

Historiquement, les VPN étaient construits sur des lignes louées, mais le coût de ces infrastructures était trop important. Maintenant, les VPN sont basés sur l'Internet dont l'avantage principal est son faible coût. Mais en utilisant l'Internet, les données privées deviennent accessibles à tout le monde. Il a donc été nécessaire de fournir des protocoles permettant d'assurer la confidentialité et l'intégrité des données à travers le réseau public.

Un tunnel VPN est monté entre deux passerelles VPN ou entre deux hôtes. Les deux équipements sont d'un point de vue logique connectées directement l'une à l'autre. Le tunnel permet d'envoyer des données du réseau privé à travers le réseau public. Il encapsule les données dans un protocole compris par les deux extrémités. L'émetteur encapsule les données et le destinataire récupère les données. En plus de l'encapsulation des données, les tunnels VPN réalisent du chiffrement.

5.1.1 Les types de VPN

Il existe deux grands types de VPN : les VPN site-à-site et les VPN client-à-site ou VPN "remote access".

Les VPN site-à-site

Ils sont utilisés pour connecter des sites entre eux. Le tunnel est monté entre deux passerelles VPN dont les configurations sont connues. Le réseau Internet opère comme une liaison WAN entre les sites. Les employées peuvent échanger des informations entre les différents sites comme s'ils sont connectés sur le site distant. Dans ce cas-ci, nous utilisons principalement des tunnels VPN IPsec.

Les VPN client-à-site

Ils sont généralement utilisés par des travailleurs pour accéder aux ressources de l'entreprise depuis des emplacements non fiables. L'utilisateur se connecte via son ordinateur ou son smartphone à la passerelle VPN de son entreprise. Dans ce cas, la configuration n'est pas connue, car selon la localisation de l'utilisateur, les paramètres de connexion changent. Il est souvent nécessaire d'installer sur l'appareil mobile un

client VPN. Nous utilisons majoritairement des tunnels VPN SSL pour leur simplicité de configuration. Bien qu'il soit possible de créer des tunnels de ce type avec IPSec. L'utilisation de SSL est soumise à controverse depuis la découvert de *Hearthbleed*.

Avant de présenter les protocoles IPSec, SSL et TLS, intéressons-nous au protocole de tunneling.

5.1.2 Les protocoles de tunneling de couche 2

Avant de créer le tunnel, ces protocoles se mettent d'accord sur des paramètres de session, comme le protocole de chiffrement. Une fois les paramètres acceptés par les deux extrémités, le tunnel est monté. Les données sont envoyées via ce tunnel entre les deux extrémités. Il est possible de transmettre des données issues de protocole non-routable à travers le réseau public. Les données sont encapsulés dans un autre protocole qui est routable.

Par exemple, le protocole AppleTalk n'est pas supporté par l'Internet, mais grâce au tunnel, nous encapsulons le paquet AppleTalk dans un paquet IP. Ce paquet est envoyé sur l'Internet qui va le transmettre au destinataire. Ce dernier désencapsule le paquet IP et récupère le paquet AppleTalk.

Il est nécessaire que les deux extrémités du tunnel comprennent le protocole encapsulé. Sinon, la destination ne saura pas capable de traiter le paquet encapsulé.

Point-to-Point Tunneling Protocol - PPTP

C'est un protocole propriétaire de Microsoft. Il est implémenté depuis Windows 2000 dans toutes les machines Windows et il existe des clients PPTP pour Linux et OS X. Sa configuration est simple, mais la sécurité qu'il propose n'est pas exempt de tout reproche.

Ce protocole encapsule des frames PPP [1] dans un tunnel IP. Il fonctionne en quatre phases dont une est optionnelle : Link Establishment Phase, Authentication Phase, Callback Control Phase, Network Control Phase.

1. La phase une sert à établir, maintenir et terminer la connexion physique entre les deux hôtes. C'est aussi à ce moment que les protocoles d'authentification sont choisis.
2. Le client PPTP est authentifié en utilisant le protocole PPP. L'authentification peut se faire en claire ou en utilisant des protocoles comme PAP et CHAP.
3. La phase trois est optionnel et elle permet une sécurité accrue. Elle déconnecte le client et le serveur. Ensuite, le serveur rappelle le client.
4. La dernière phase sert à négocier et implémenter les protocoles de compression et de chiffrement.

Layer 2 Tunneling Protocol - L2TP

Ce protocole a été normalisé en utilisant les avantages des protocoles PPTP et L2F de Cisco [2]. Mais il ne permet toujours pas la confidentialité du trafic. Il est possible de faire de l'authentification et du chiffrement avec les paquets PPP, mais la connexion reste vulnérable au niveau de la couche transport. Il est donc intéressant d'associer L2TP avec un autre protocole de sécurité comme IPSec.

PPTP et L2TP ne possèdent pas un niveau de sécurité élevé pour être utilisé comme tunnel VPN à travers Internet. Ils sont le plus souvent remplacer par des tunnels IPSec ou SSL, qui possèdent un meilleur niveau de sécurité.

5.2 Internet Protocol Security - IPSec

IPsec est un ensemble de protocole visant à sécuriser les données au niveau de la couche réseau. Il est composé de trois protocoles : AH (*Authentication Header*), ESP (*Encapsulating Security Payload*) et IKE (*Internet Key Exchange*). IKE est utilisé lors de la négociation des paramètres du tunnel VPN. Les deux autres protocoles fournissent la sécurité des données en les encapsulant au sein du tunnel VPN.

1. Point-to-Point Protocol
2. Layer 2 Forwarding

5.2.1 Les protocoles AH et ESP

Le protocole AH fournit une authentification sur la source du paquet, l'intégrité des données et une protection contre les attaques par rejeu. Le protocole ESP fournit les mêmes sécurités que AH, sauf qu'il n'authentifie l'en-tête IP des paquets. Il fournit en plus la confidentialité des données en chiffrant une partie du paquet. AH et ESP peuvent travailler en mode tunnel ou en mode transport.

Le mode transport est utilisé lorsque les terminaisons du tunnel VPN sont les destinataires finaux de la communication. Ce mode offre une sécurité de bout en bout. Au niveau des paquets, les en-têtes AH et ESP sont placés après l'en-tête IP, il n'y a donc qu'un seul en-tête IP.

Le mode tunnel est plus souple, mais il consomme plus de bande passante. Les destinataires finaux sont connectés via des passerelles VPN. Il n'y a pas de différence entre les VPN site-à-site et les VPN client-à-site. Les passerelles VPN ont pour objectif d'encapsuler les paquets pour les faire passer dans le tunnel VPN. Nous trouvons donc dans le paquet encapsulé l'en-tête IP du paquet d'origine (voir Fig.I p.10). Comme le paquet est passé à travers la passerelle, il possède un deuxième en-tête IP qui sert au routage entre les deux passerelles.

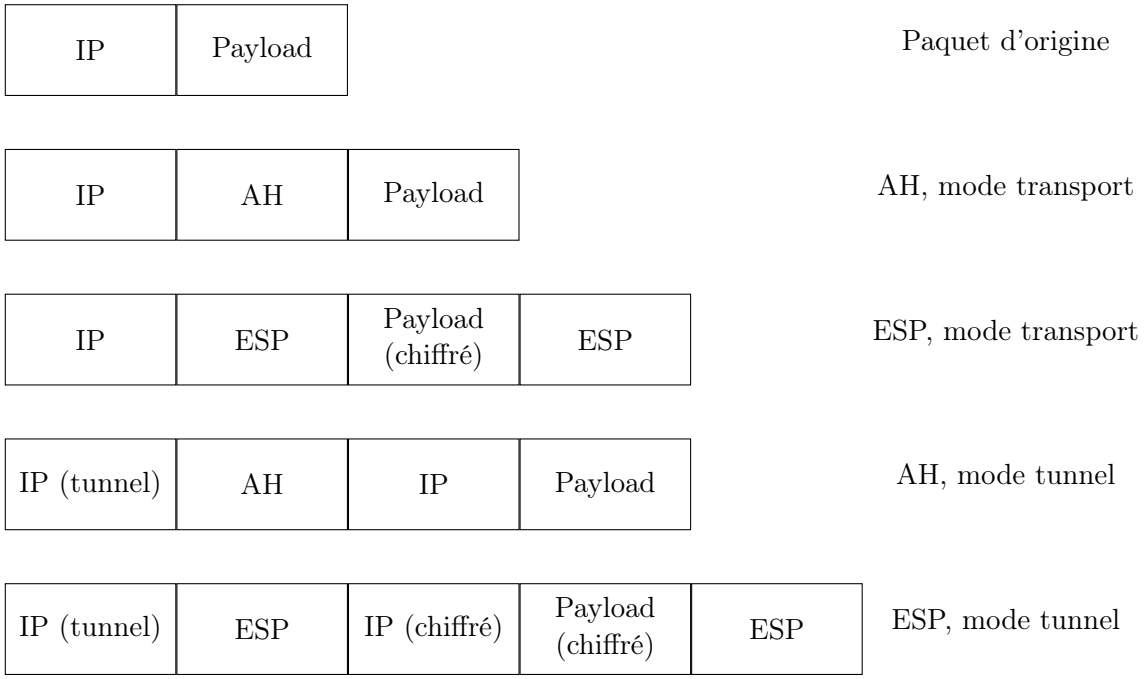

FIGURE I – Schéma des encapsulations IPSec

5.2.2 Security association

Quand nous parlons de monter un tunnel, en réalité, nous synchronisons un état partagé entre les terminaisons du tunnel. Cet état partagé se nomme une SA (*security association*) en IPSec. Une SA contient l'algorithme de chiffrement utilisé et les clés utilisées, l'algorithme d'authentification, un numéro d'identifiant, le *security parameter index* (SPI), ... Plus d'autres paramètres qui servent à maintenir les tunnels VPN. Les SA peuvent être créées manuellement ou gérées par l'IKE. Chaque terminaison possède deux SA, une pour le trafic entrant et une pour le trafic sortant. De plus, chaque pair est lié à un protocole. Les SA se caractérisent par un triplet formé du SPI, de l'adresse de destination et du protocole. Les SA sont stockées dans une SAD (*security association database*). Cette SAD est utilisée pour déterminer quel protocole est utilisé pour les paquets sortants et pour fournir les paramètres pour déchiffrer et/ou authentifier les paquets entrants. Il est possible de combiner les SA pour créer des tunnels VPN complexe.

Les SA sont des éléments simples, c'est-à-dire qu'elles traitent tous les paquets de la même manière. Pour un réglage plus fin, IPSec utilise des policies. Ces policies se basent sur les champs suivants des en-têtes du paquet.

— L'adresse de destination
— L'adresse source
— Le protocole de la couche transport
— Le port source
— Le port de destination

Elles servent à déterminer quels paquets à émettre sur quel tunnel, à dropper les paquets ne correspondant à aucune des règles décrites dans les policies. De la même manière que les SA, les policies sont stockées dans une SPD (*security policy database*). Le fonctionnement est similaire, pour chaque paquet entrant ou sortant, le système consulte la SPD pour déterminer les règles à appliquer au paquet. Si une règle est trouvé, le système cherche après la SA correspondante.

5.2.3 Le protocole IKE

IKE a un seul objectif : procéder à des échanges de clé Diffie-Hellman pour sécuriser un tunnel VPN. Il négocie le chiffrement, l'authentification nécessaire au tunnel, qui satisfont les policies.

IKE dérive du *Internet Security Association and Key Management* Protocol (ISAKMP). ISAKMP est un framework qui fournit des outils pour la sécurisation des échanges et l'échange de clé. De plus, IKE utilise différents mode du protocole OAKLEY. Il établit une SA en deux phases et il possède cinq modes d'échange, dont trois découlent ISAKMP. Les deux derniers modes ne sont utilisés que lors de la phase deux.

La phase 1 d'IKE

La phase 1 crée un canal sécurisé entre les terminaux du tunnel pour déterminer les SA. Le canal sécurisé est créé après l'authentification des terminaux. Les SA de la phase 1 sont bidirectionnelles, c'est-à-dire qu'une SA sécurise le trafic entrant et sortant. Pour l'échange des SA de la phase 1, IKE possède deux modes d'échanges :

— Main mode
— Agressive mode

Pour l'authentification d'un terminal, il existe quatre méthodes :

— a shared secret
— a digital signature
— public key encryption
— revised public key encryption

Le mode *"main"* d'IKE travaille en trois étapes (voir Fig.II p.12). Premièrement, l'initiateur envoie un message contenant une liste des méthodes de sécurisation qu'il utilise. Le receveur choisit dans la liste reçue la méthode correspondant à ses policies et envoie sa décision à l'initiateur. Ensuite, ce dernier envoie sa clé privée pour créer le secret partagé de l'algorithme de Diffie-Hellman. Le receveur fait de même. Ils sont donc capables tous les deux de créer les clés. Les clés dépendent des méthodes d'authentification choisies. Finalement, l'initiateur envoie son identité et des informations sur l'authentification. Ces messages sont chiffrés et masquent donc l'identité des terminaux. L'échange se fait en six messages.

À la fin de ce mode, les terminaux sont d'accord sur les algorithmes de chiffrement et de confidentialité des données. Ils possèdent également les clés pour les algorithmes sélectionnés.

Le mode *"agressive"* fait le même travail de façon plus rapide, il n'utilise que trois messages (voir Fig.III p.12). Lors du premier envoi, l'initiateur émet la liste des méthodes de sécurisation, son identité et sa clé. Le receveur répond par son choix de sécurisation, sa clé, son identité et ses identifiants. Finalement l'initiateur s'authentifie auprès du receveur. Ce dernier message peut être chiffré.

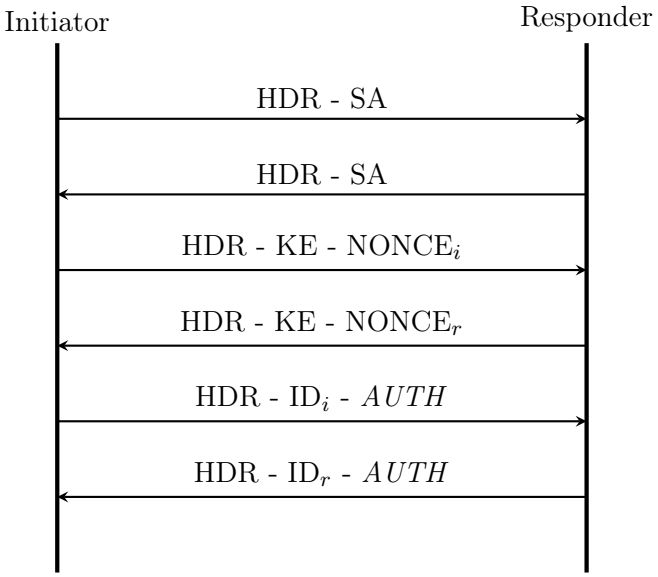

FIGURE II – IPsec : mode "main"

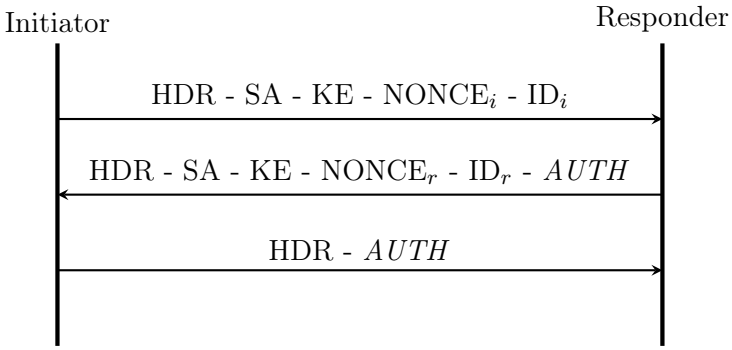

FIGURE III – IPsec : mode "agressive"

La phase 2 d'IKE

Une fois la SA établie, les terminaux peuvent l'utiliser pour négocier les SA de phase 2. La phase 2 est un échange en Quick mode. L'échange se fait en trois messages. Lors de l'échange, il est possible de négocier plusieurs SA en même temps.

5.3 Secure sockets layer - SSL

Netscape a lancé SSL 1 en 1994, dans le but de sécuriser des transactions réalisées avec leur navigateur. Dans la même année, SSL 2 était déjà en route. Mais le protocole montrait déjà des vulnérabilités. Fin 1995, SSL 3 était lancé. Il s'agissait d'une version complètement récrite de SSL, qui introduisait de nouvelles fonctionnalités issues de PCT [3]. Bien que les machines actuelles intègrent SSL 3, elles essaient d'abord de négocier une connexion en SSL 2.

SSL utilise des suites de chiffrement. Ces suites se composent de trois fonctions de chiffrement : la méthode d'échange de clé, l'algorithme de chiffrement et une méthode de hachage. Il existe un large ensemble de suite, les clients envoient la liste des suites qu'ils savent gerer au serveur. Ce dernier choisit, dans la liste, une suite qu'il implémente.

OpenSSL est l'implémentation la plus courante de SSL. Cette implémentation possède un interface

3. Microsoft's *Private Communications Technology*

en ligne de commande, qui permet de générer des clés RSA, signer des certificats, calculer des valeurs de hash, etc..

Dans un effort de standardisation de SSL, l'IETF a lancé le protocole TLS [4]. Il se base principalement sur SSL 3 bien qu'il ne soit pas compatible avec ce dernier. L'utilisation de SSL est déconseillé suite à de nombreuses vulnérabilités détectées ces derniers temps (FREAK (2015), Hearthbleed (2014), BEAST (2013), etc.).

5.3.1 Le protocole SSL

SSL est un protocole de la couche transport, il utilise donc les protocoles de cette couche pour le transfert des données. Pour éviter des problèmes lors de la transmission des données, SSL utilise le protocole TCP.

De manière analogue à TCP, une session SSL se divise en trois phase (voir Fig.IV p.13) :

1. Établissement de la connexion
2. Transfert des données
3. Clôture de la connexion.

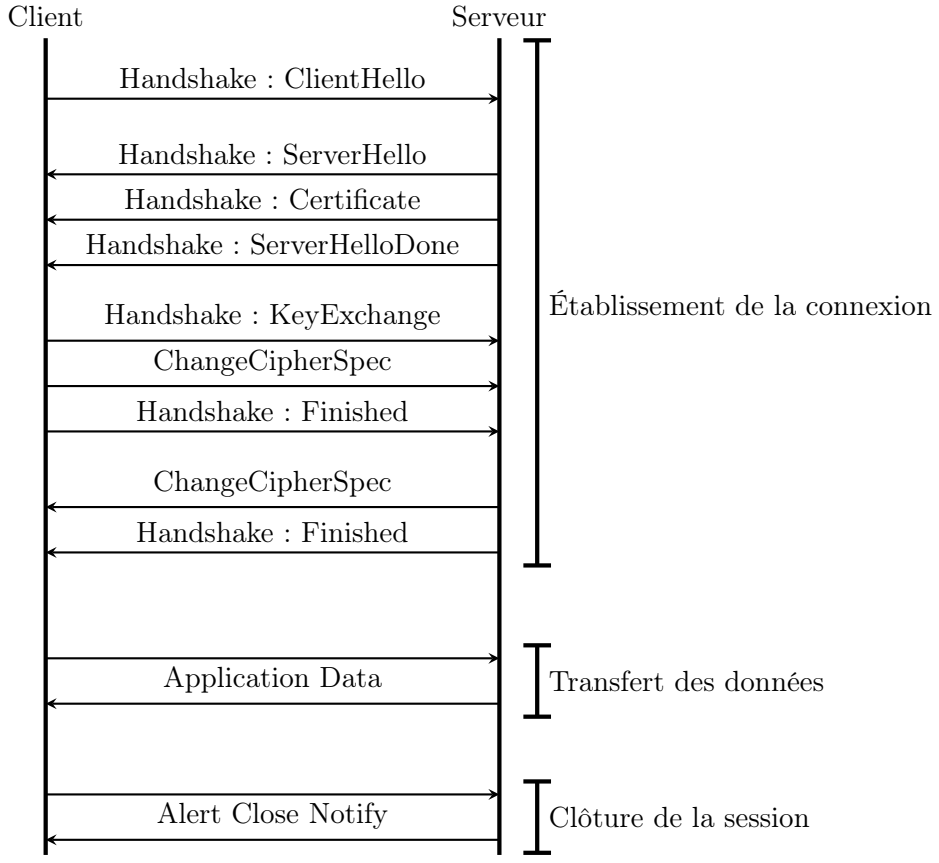

FIGURE IV – Session SSL

La session commence par le *triple handshake*. Le client envoie un message ClientHello, qui indique la version de SSL supportée, la liste des suites de chiffrement et les algorithmes de compression. La version de SSL est signalée par deux champs dans l'en-tête : version mineur et version majeure. SSL3 a une version majeure de 3 et une version mineure de 0, et TLS a un version majeure de 3 et une version mineure de 1.

Le serveur répond par trois messages.

4. Transport Layer Security

1. Le message *ServerHello* indique au client la suite de chiffrement et l'algorithme de compression à utiliser.
2. Le certificat du serveur permet au client de vérifier l'identité du serveur et contient la clé publique du serveur. Cette clé va servir à générer les différents clés pour la session.
3. Le message *ServerHelloDone* précise la fin de la séquence *Hello*.

Suite à ces trois messages, le client donne au serveur des inputs pour la génération des clés (*ClientKeyExchange*), signal au serveur qu'il utilise les nouvelles clés pour le chiffrement et l'authentification (*ChangeCipherSpec*) et qu'il a fini le handshake (*Finished*). Le serveur répond avec son message *ChangeCipherSpec* et son *Finished*.

Le client et le serveur sont capables de s'échanger des données de façon sécurisé.

Finalement, la session est clôturée.

5.4 Transport Layer Security - TLS

TLS est le successeur de SSL. À l'heure actuelle, TLS est en version 1.2 [5]. La version 1.3 est en *draft*. De la même manière que SSL, TLS a pour objectif de sécuriser les communications entre deux entités sur un réseau.

5.4.1 Détails du protocole

TLS se base sur des enregistrements pour l'échange de données. Un enregistrement est caractérisé par un champ *content type*, un champ version et un champ longueur.

Avant de pouvoir échanger des données applicatives, TLS réalise le *TLS Handshake* dont le *content type* est 22. Ce *handshake* fournit les paramètres de chiffrement pour l'échange de données.

En TLS, il existe le *basic handshake*, qui authentifie uniquement le serveur, et le *client-authenticated handshake*, qui authentifie les deux *peers* (voir Fig.V p.15).

Basic handshake

1. Le client envoie un message `ClientHello`, qui contient les versions de TLS supportées, une liste des suites de chiffrement.
2. Le serveur répond par un message `ServerHello`. Ce message contient la version de TLS et la suite de chiffrement à utiliser. Il envoie un message `Certificate` ou `ServerKeyExchange` si la suite de chiffrement l'exige. Et il transmet un message `ServerHelloDone` pour signaler la fin des négociations.
3. Le client répond avec un message `ClientKeyExchange`. Ce message contient soit un *PreMasterSecret*, soit une clé publique, soit rien.

   Le client et le serveur utilisent les éléments reçus précédemment pour générer un secret maitre. Ce secret sert à créer toutes les autres clés qui seront utilisées pour cette connexion.
4. Le client envoie un enregistrement `ChangeCipherSpec` dont le *content type* est 20. Cet enregistrement signale au serveur que les messages du client seront authentifiés et chiffrés à l'avenir par la suite de chiffrement définit lors du *handshake*.
5. Finalement, le client émet un message `Finished`. Ce messages est authentifié et chiffré. Si le serveur n'arrive pas à vérifier ce message, il ferme la connexion.

   Le serveur envoie également un message `Finished`.
6. Le client et le serveur peuvent s'envoyer les données applicatives en tout sécurité. L'enregistrement utilisé possède un *content type* de 23.

5. RFC 5246 - `http://tools.ietf.org/html/rfc5246`

Client-authenticated handshake

Le déroulement est identique à celui du *basic handshake*. Sauf qu'avant que le serveur envoie son `ServerHelloDone`, il demande le certificat du client via le message `CertificateRequest`.

Le client répond en envoyant son certificat dans le message `Certificate`. De plus, il émet un message `CertificateVerify`. Ce dernier est signé par le clé privée du client. Le serveur vérifie cette signature en utilisant la clé publique contenu dans le certificat du client reçue auparavant.

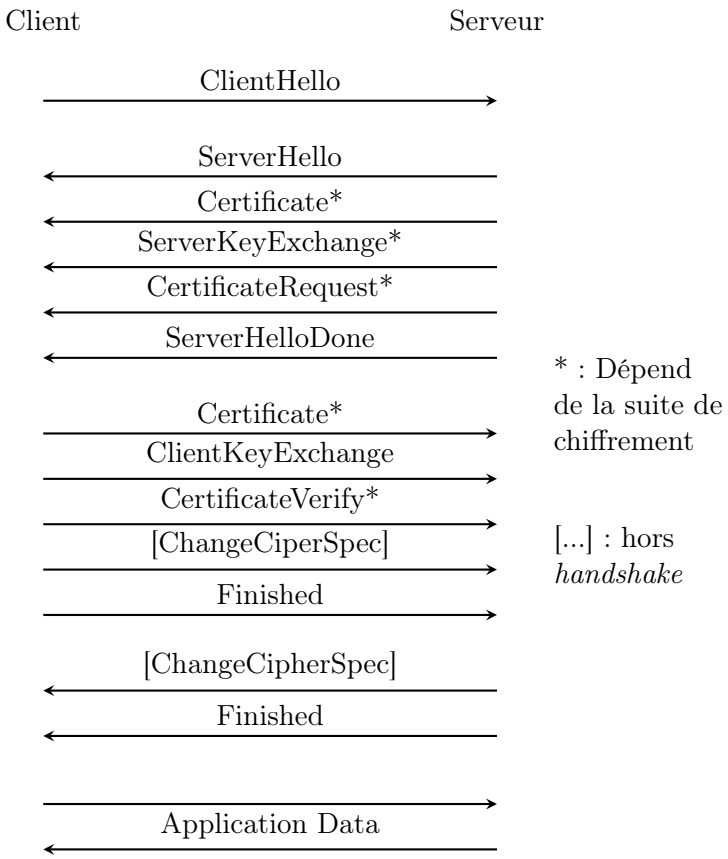

FIGURE V – Échange pour un *handshake* TLS

L'ID de session

Le message `ServerHello` contient un id de session. Cet id est lié aux paramètres de chiffrement dont le secret maitre. Il est surtout utiliser lors d'une reconnexion pour éviter toute la procédure du *handshake*. Cette dernière peut être lourde en terme de ressources pour le serveur.

Un schéma est disponible voir Fig.VI p.16.

Le ticket de session

Le ticket de session permet à des serveurs de remettre en état une session SSL sans que l'état de la session soit enregistré sur le serveur. Le ticket est transmis au client du façon sécurisé. Ce dernier va le retransmettre au serveur lorsqu'il se reconnectera au serveur.

Un schéma de l'échange est disponible à la Fig.VII p.16.

Le client prévient le serveur qu'il supporte ce système en ajoutant dans le `ClientHello` une extension `SessionTicket SSL`.

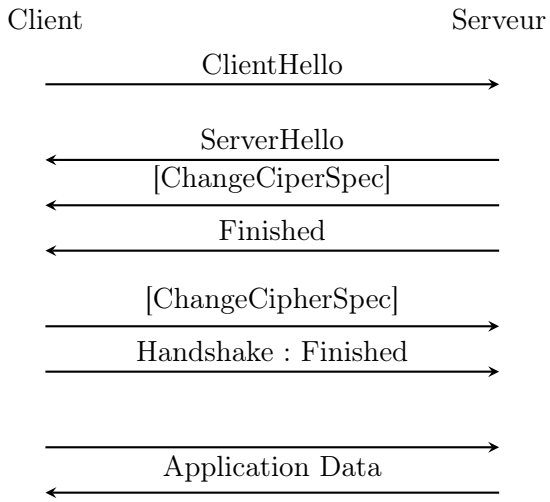

FIGURE VI – Échange pour un *handshake* TLS avec un ID de session

Le serveur stocke dans un ticket les informations sur la session (suite de chiffrement et le secret maitre). Le ticket est chiffré par une clé connue uniquement du serveur. Il est ensuite transmis au client en utilisant le message `NewSessionTicket`. Le message est envoyé avant que le serveur envoie son `ChangeCipherSpec` et après que le serveur ait vérifié l'authenticité du client.

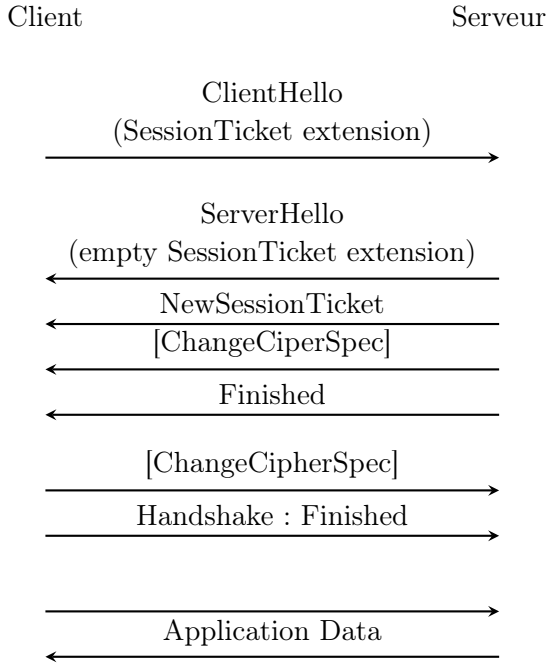

FIGURE VII – Échange pour un *handshake* TLS avec un ticket de session

5.5 Secure Shell - SSH

SSH a pour objectif de créer une connexion sécurisée. De la même manière que SSL, SSH est un protocole de la couche transport et utilise TCP. Par contre les applications ne doivent pas forcément intégrer SSH pour être utilisées via SSH.

SSH est principalement utilisé pour remplacer `telnet`, mais il est également possible de faire du VPN. En effet, SSH fournit de l'authentification et du chiffrement pour les communications entres les machines.

Les tunnels SSH sont peu utilisés, car ils manquent de performance. Mais ils sont simple à mettre en place.

5.6 Secure Socket Tunneling Protocol - SSTP

SSTP est utilisé pour transporter du trafic PPP/L2TP via du SSL3. Son avantage réside dans le fait qu'il peut passer à travers les NAT, les proxys et les firewalls.

5.7 HTTP over TLS/SSL - HTTPS

Il s'agit rien de plus qu'une connexion HTTP chiffré par SSL ou TLS. Il fournit une authentification pour les serveurs Web et un chiffrement bidirectionnel pour les communications entre le navigateur Web et le serveur Web.

La confiance envers un site Web est liée à un certificat. Ce dernier doit provenir d'une autorité certificative.

5.8 Les technologies propriétaires

Les fabricants ont développé leur propre implémentation d'accès à distance.

5.8.1 Remote Desktop Services de Microsoft

Cette solution est apparu sous le nom de "Terminal Services" dans Windows NT 4.0. Depuis Windows Server 2008 R2, le service s'appelle "Remote Desktop Services".

Ce service permet à un utilisateur de prendre le contrôle d'une machine à distance avec les droits associés au compte qu'il utilise. Ainsi l'administrateur peut gérer les accès aux serveurs accessibles.

5.9 Fournisseurs d'accès

Chapitre 6

Comparaison théorique

Dans ce chapitre, je réalise un comparatif théorique des technologies décrites dans le chapitre précédent. Je précise, dans un premier temps, les critères de comparaison. Puis, je présente le tableau. Finalement, je réalise une analyse de ce tableau.

6.1 Les critères de comparaison

Tous les protocoles décrits dans le chapitre 5 visent un même but, mais en ayant des implémentations différentes. Pour pouvoir les différencier, il est utile de réaliser un comparatif sur des critères pertinents. Les critères que j'ai sélectionné sont :

— La facilité de configuration
— Le type de VPN
— La couche de protection
— L'intégrité des données
— L'authentification
— Le chiffrement

D'un point de vue purement technique, la facilité de configuration précise le degré de complexité pour mettre en place un tunnel VPN en utilisant une technologie.

6.2 Tableau de comparaison

Protocoles	Facilité de configuration
IPSec (AH, tunnel)	++
IPSec (ESP, tunnel)	++
IPSec (AH, transport)	+++
IPSec (ESP, transport)	+++
SSL/TLS	++++
SSH	++++
SSTP	++++
HTTPS	+++++

Table 6.1 – Tableau de comparaison théorique des technologies d'accès à distance

Chapitre 7

Architecture

Troisième partie

Comparaison des solutions commerciales

Chapitre 8

Choix des solutions

Chapitre 9

Installation des solutions

Chapitre 10

Scénario de test

Chapitre 11

Critères de comparaison

Test	de	tableau

Table 11.1 – Test de tableau

Conclusion