StrongSwan

Installation et configuration

Documentation version 1.1 créée le 10 mai 2005

Dernière mise à jour le 20 juin 2005

Auteurs :

EMONET Jean-Bruno jbemonet@ext.cri74.org

Haute-Savoie
Centre de Ressources Informatiques

Bâtiment le Salève I
Site d'Archamps
F-74160 ARCHAMPS

Tél. : +33 (0)4 50 31 56 30
Fax : +33 (0)4 50 95 38 17

E-mail : info@cri74.fr
Web : www.cri74.fr

SIRET : 400 210 646 000 14
APE : 913 E

Le CRI dépend de l'Agence Économique Départementale - BP 2444 - F-74041 ANNECY CEDEX

Table des matières

1 – Présentation

1.1 – L'objectif

Cette documentation explique l'installation,la configuration et l'utilisation de strongSwan dans le but de mettre en place un VPN (Virtual Private Network) entre 2 sous-réseaux.

Deux types de configuration vont être abordées avec plusieurs modes d'authentification.

A la fin de cette documentation, les erreurs les plus courantes seront expliquées le plus clairement possible.

1.2 – Les fonctionnalités

StrongSwan :

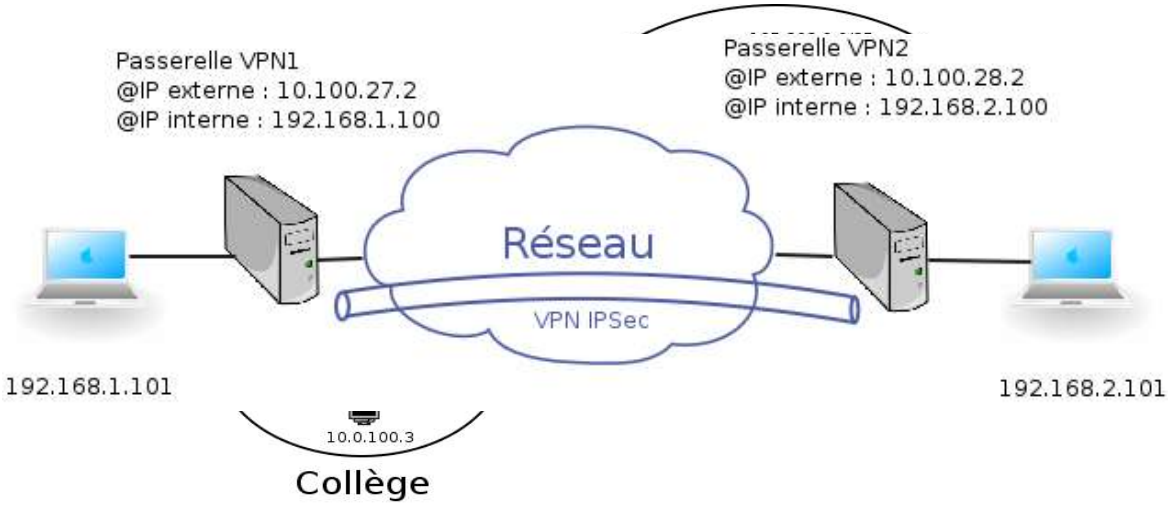
- Fonctionne sur kernel 2.4 avec KLIPS et 2.6(Native IPSec)
- Supporte les algorithmes de chiffrement : 3DES, AES, Serpent, Twofish or Blowfish
- Utilise une authentification par secrets partagés, clés RSA ou certificats X.509

Pour plus de précisions :

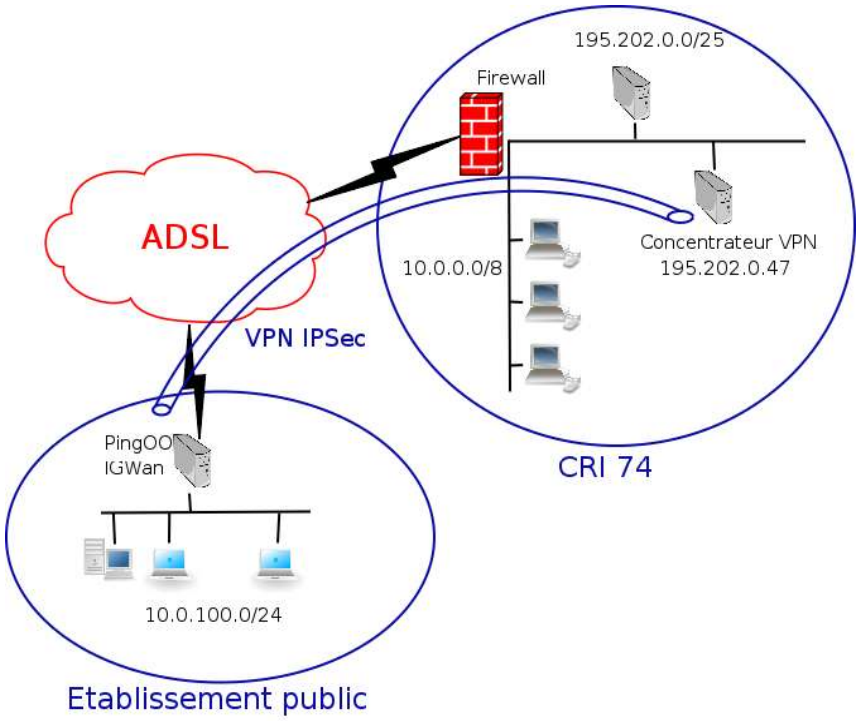
- http://www.strongswan.org/index.htm

1.3 – Les configurations réseaux

1.3.1 – 1ère configuration



1.3.2 – 2nde configuration



1.4 – L'environnement

Dans notre cas, les deux passerelles fonctionnent sous une distribution Debian Sarge avec un noyau 2.6 permettant les fonctionnalités IPSec suivantes :

- PF_KEY socket interface de communication permettant de définir les politiques de sécurité et les associations de sécurité propres à IPSec.
- AH transformation : implémentation du protocole AH de IPSec
- ESP transformation : implémentation du protocole ESP de IPSec
- IPComp transformation : implémentation du protocole IPComp (compression) de IPSec.
- IPSec user configuration interface : interface avec les outils de configuration en mode utilisateur.

Les clients fonctionnent sous Linux avec n'importe quelle distribution.

2 – Installation

2.1 – Pré-requis

2.1.1 – Librairies

Pour pouvoir installer StrongSwan, il faut tout d'abord installer les librairies GMP (Gnu MultiPrecision) : libgmp3 et libgmp3-dev dans le cas d'une distribution Debian Sarge.

2.1.2 – Modules

Il faut vérifier la présence des modules : esp4, ah4, ipcomp, af_key, xfrm_user.

2.1.3 – Paquetages

Vérifier que le paquetage iproute est bien installé. Celui-ci permet à strongSwan d'établir les routes entre les sous-réseaux.

2.2 – Installation depuis un tar.bz2

Le paquetage est disponible sur le site de StrongSwan: http://www.strongswan.org.

Pour l'installer :

•Décommenter les options voulues dans le fichier pluto/Makefile. Pour plus de précisions, il faut voir la page officielle de StrongSwan(en Anglais).

- http://www.strongswan.org/docs/install.htm#chapter_2

```
# make programs
# make install
```

3 – Configuration

3.1 – 1ère configuration

StrongSwan se configure à l'aide de 2 fichiers :

- /etc/ipsec.conf : contient la configuration des connexions
- /etc/ipsec.secrets : contient un secret partagé ou une clé RSA. Les droits d'accès à ce fichier doivent être à 600.

3.1.1 – Clés RSA

Les clés RSA sont utilisées pour l'authentification. Chaque partie possède une clé RSA(clé publique+clé privée). La clé publique est diffusée tandis que la clé privée reste secrète. Seule le possesseur de la clé privée pourra déchiffrer un message reçu, chiffrer avec sa clé publique. Ce chiffrement asymétrique est utilisé pour s'échanger une clé de session. Celle-ci sera utilisée pour le chiffrement et déchiffrement. Cette clé de session est renégociée périodiquement.

3.1.1.1 – Générer une clé RSA

Voici la commande qui génère une clé RSA avec une clé publique de 2048 bits :

```
# ipsec newhostkey -bits 2048 -output cleRSA.key
```

Cette clé doit avoir des droits d'accès restreints : chmod 600 cleRSA.key

3.1.1.2 – ipsec.secrets

Ce fichier va contenir la clé RSA. Il suffit de renommer la clé RSA déjà crée ou de générer la clé de la façon suivante :

```
# ipsec newhostkey -bits 2048 -output /etc/ipsec.secrets
```

3.1.1.3 – ipsec.conf

Dans ce fichier, « left » correspond au client VPN :10.100.27.2 et « right » au serveur VPN : 10.100.28.2. Il faut respecter cette règle sur les 2 passerelles.

```
# /etc/ipsec.conf - strongSwan IPsec configuration file

version 2.0     # conforms to second version of ipsec.conf specification

config setup
        interfaces=%defaultroute
        plutodebug=control
        klipsdebug=none
        uniqueids=yes

conn %default
        keyingtries=0
        disablearrivalcheck=no
        authby=rsasig
conn net-net
        left=10.100.27.2
        leftnexthop=10.100.27.1    #rightnexthop=10.100.28.1 sur la passerelle 10.100.28.2
        leftsourceip=192.168.1.100 #rightsourceip=192.168.2.100 sur la passerelle
10.100.27.2
```

Le paramètre leftsourceip correspond à l'adresse interne de la passerelle. En effet, la passerelle d'adresse 10.100.27.2 ne peut communiquer directement avec le sous-réseau

distant, il faut qu'elle passe par son adresse interne. Le VPN s'étend entre les 2 interfaces internes des 2 passerelles.

```
        leftsubnet=192.168.1.0/24
        leftid=@pc002.vpn1.stagiaires.tst
        leftrsasigkey=0sAQPFURVBUzWStHTzPzACdNivSk5AZI2qT9IogF5eKulX5sDzBsqE+w9bIL5TybqmX3
b2BfmNJ6jRn+Kr8ayl3MAJWd4pB8xNapIrFlKX4jawoSIpUApTnCXZ3cZ5apqC3sT/q0/A+gSPmGn+DLWYTWY3QFBB
WHug/uFdxOeBAIAFw/sk9VasfcXQn4Ml/JFIleG806Ssf3wZHrVUO0ejUAj05aREQvJP02aapibG/D9tGdDmpJLEqk
9n/iLmr9U6rWd8sp7v/RdVGJNGo7nvzpirUfqQF/I4o2/vYdhfCSeqjVZFDquPsDTjY4aAGtSCsa1U20VsLjh91xzN
SBMT1lUuR4Xrdmy3Zqv3mcaMcJM7+aax
```

Clé publique de 10.100.27.2

```
        rightrsasigkey=0sAQOKCYKsFSXiHCVU2Xv72NdYBc9oda3e7Bat5vMvqXb7iuHaR7eaFGwqVL0QTn2BE
x7pxi+TAwygmNCXQ8csVAWFuvB7qi8+S9uAqzUZC3X2nLUcEfx6qeyu+PYiyGrV2uVq51dpatZx1XlogF0sQOsvdLa
jvFGTxwZKax6xueIcwpBrsudzQUQfTwjKc3jHhuA1RvlS6VL85DTHZbOoaIqr2tP+GhmHF97mKNkWWE6jYNxtdtlws
4nZsssI9+W+yUif5n/Md0FPgKhkscZGVXeSXS/u+ZLYWa2U/rRnnSqByi8Sie+6jmx3As0zDT9EINOvBu4ig4ZwJ3O
IEMdS+1vXAvf/gXhp5tNsKuDZ+B8ZeEZv
```

Clé publique de 10.100.28.2

```
        right=10.100.28.2
        rightsubnet=192.168.2.0/24
        rightid=@pc002.vpn2.stagiaires.tst
```

Le « rightid » et le « leftid » correspond au FQDN de la passerelle.

```
        esp=3des-sha1
        auto=start
```

auto=start : crée les connexions au démarrage => Client

auto=add : autorise les connexions mais ne les lance pas au démarrage. => Serveur

3.1.1.4 – Démarrage de StrongSwan

Pour lancer strongSwan :

```
# /etc/init.d/ipsec start
```

Créer une connexion :

```
# ipsec auto - -up « nom de la connection»
```

Il faut aussi penser à activer l' ip_forward :

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

3.1.1.5 – Commandes utiles

Pour obtenir la clé RSA public :

```
# ipsec showhostkey - -left
```

ou

```
# ipsec showhostkey -right
```

Vérifier la configuration :

```
# ipsec verify
```

Affiche en détail tous les logs :

```
# ipsec barf | more
```

Affiche les connexions :

```
# ipsec auto -status
```

3.1.1.6 – Fichiers de logs

Il y a deux fichiers intéressants pour les logs :

- /var/log/daemon.log
- /var/log/auth.log

3.1.2 – Certificat X.509

Les certificats X.509 sont crées et signés à l'aide d'une infrastructure à gestion de clés(PKI). Chaque passerelle VPN possède un certificat signé(= clé publique) par une autorité de certification(CA) de confiance, une clé privé et le certificats de l'autorité de certification. L'avantage des certificats est que chaque élément du VPN ne doit pas connaître au départ toutes les clés publiques des autres. Cela simplifie la configuration. La passerelle reçoit un certificat,vérifie la signature d'après le certificat de la CA et obtient la clé publique de la passerelle à qui elle souhaite envoyer des données.

Le certificat de la passerelle est stocké dans : /etc/ipsec.d/certs

Le certificat de la CA est stocké dans : /etc/ipsec.d/cacerts

La clé privée est stockée dans : /etc/ipsec.d/private

3.1.2.1 – ipsec.secrets

Le fichier ipsec.secrets va contenir l'emplacement de la clé privée et sa passphrase si celle-ci est chiffrée.

Voici un exemple :

```
: RSA /etc/ipsec.d/private/vpn1key.pem "xxx"
```

3.1.2.2 – ipsec.conf

Seule la partie « conn net-net » est modifiée :

```
conn net-net
        left=10.100.27.2
        leftcert=pc002_vpn1_stagiaires_tst.pem
        leftnexthop=10.100.27.1
        leftsourceip=192.168.1.100
        leftsubnet=192.168.1.0/24
        leftid="C=FR, ST=Haute-Savoie, ..."
```

Le leftid correspond au champs « subject » (= DN) dans le certificat de la passerelle 10.100.27.2

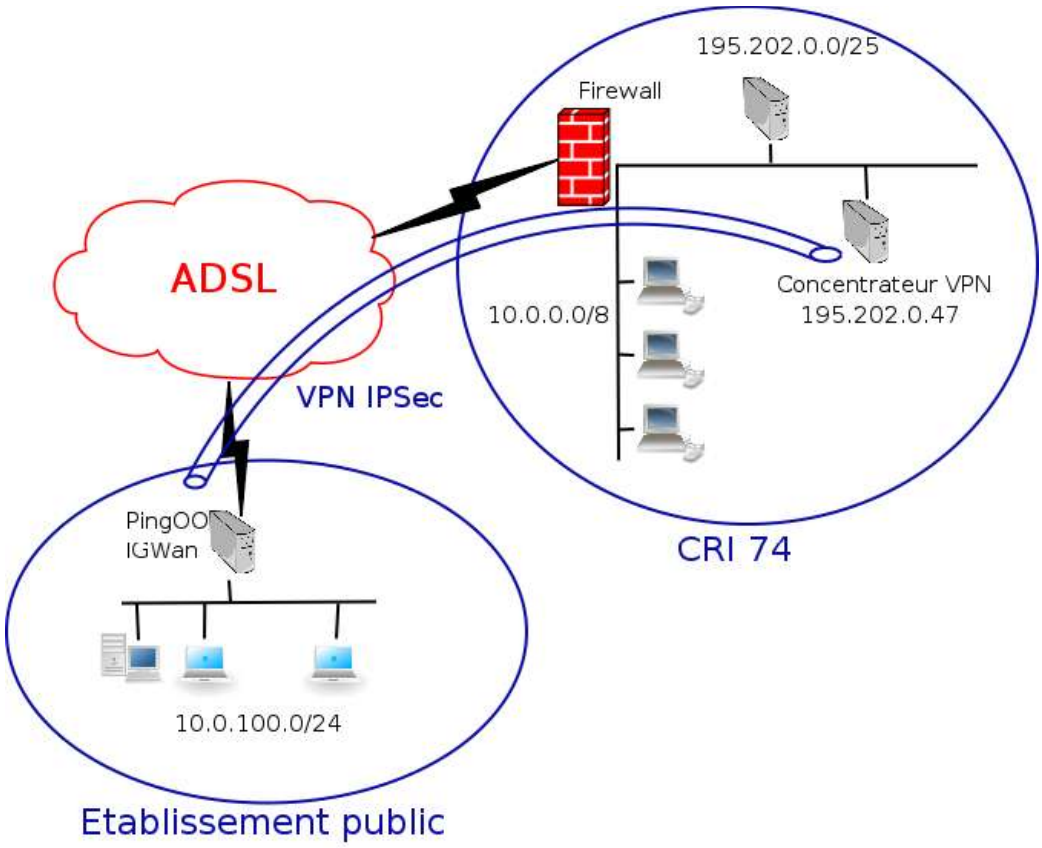
```
        right=10.100.28.2
        rightsubnet=192.168.2.0/24
        rightid="C=FR, ST=Haute-Savoie, ..."
```

Le rightid correspond au champs « subject » dans le certificat de la passerelle 10.100.28.2

```
        esp=3des-sha1
        auto=start
```

3.2 – 2nd configuration

Cette configuration demande la mise en place de 2 connexions VPN. Une pour relier le sous-réseau de l'établissement public et le réseau 195.202.0.0 et l'autre pour relier les sous-réseaux de l'établissement public et 10.0.0.0/8



3.2.1 – Clés RSA

Seul le fichier de configuration ipsec.conf change.

3.2.1.1 – ipsec.conf

Sur le client (Etablissement public)

```
config setup
          interfaces=%defaultroute
          plutodebug=control
          uniqueids=yes

conn %default
          keyingtries=0
          disablearrivalcheck=no
          authby=rsasig
          left=%defaultroute
          leftsourceip=10.0.100.2
          leftsubnet=10.0.100.0/24
          leftid=@pc002.vpn1.stagiaires.tst
          leftrsasigkey=0sAQPFURVBUzWStHTzPzACdNivSk5AZI2qT9IogF5eKulX5sDzBsqE+w9bIL5Tybqm
X3b2BfmNJ6jRn+Kr8ayl3MAJWd4pB8xNapIrFlKX4jawoSIpUApTnCXZ3cZ5apqC3sT/q0/A+gSPmGn+DLWYTWY3QF
BBWHug/uFdxOeBAIAFw/sk9VasfcXQn4Ml/JFIleG806Ssf3wZHrVUO0ejUAj05aREQvJP02aapibG/D9tGdDmpJLE
qk9n/iLmr9U6rWd8sp7v/RdVGJNGo7nvzpirUfqQF/I4o2/vYdhfCSeqjVZFDquPsDTjY4aAGtSCsa1U20VsLjh91x
zNSBMT1lUuR4Xrdmy3Zqv3mcaMcJM7+aax
          rightid=@node24.cur-archamps.fr
          rightrsasigkey=0sAQOKCYKsFSXiHCVU2Xv72NdYBc9oda3e7Bat5vMvqXb7iuHaR7eaFGwqVL0QTn2
BEx7pxi+TAwygmNCXQ8csVAWFuvB7qi8+S9uAqzUZC3X2nLUcEfx6qeyu+PYiyGrV2uVq51dpatZx1XlogF0sQOsvd
LajvFGTxwZKax6xueIcwpBrsudzQUQfTwjKc3jHhuA1RvlS6VL85DTHZbOoaIqr2tP+GhmHF97mKNkWWE6jYNxtdtl
ws4nZsssI9+W+yUif5n/Md0FPgKhkscZGVXeSXS/u+ZLYWa2U/rRnnSqByi8Sie+6jmx3As0zDT9EINOvBu4ig4ZwJ
3OIEMdS+1vXAvf/gXhp5tNsKuDZ+B8ZeEZv
          esp=3des-sha1
          auto=start
```

```
conn block
          auto=ignore
conn clear
          auto=ignore
conn private
          auto=ignore
conn private-or-clear
          auto=ignore
conn clear-or-private
          auto=ignore
conn packetdefault
          auto=ignore

conn dist_cluster #Connexion vers le réseau 195.202.0.0/47
          right=195.202.0.47
          rightsourceip=195.202.0.47
          rightsubnet=195.202.0.0/25
          auto=start

conn dist_priv
        rightsubnet=10.0.0.0/8
        auto=start
```

Sur le serveur (CRI74)

```
conn dist_cluster
          right=195.202.0.47
          rightsourceip=195.202.0.47
          rightsubnet=195.202.0.0/25
          auto=add

conn dist_priv
          rightsubnet=10.0.0.0/8
          auto=add
```

La connexion est initialisée mais n'est pas crée automatiquement car c'est le client qui initialise la connexion ==> auto=add

3.2.2 – Certificat X.509

La configuration de Strongswan est similaire à celle utilisée pour les clés RSA.

3.2.2.1 – ipsec.conf

Sur le client(Etablissement public)

```
conn %default
          keyingtries=0
          disablearrivalcheck=no
          authby=rsasig
          left=%defaultroute
          leftcert=pc002_vpn1_stagiaires_tst.pem
          leftsourceip=10.0.100.2
          leftsubnet=10.0.100.0/24
          leftid="C=FR, ST=Haute-Savoie, L=Archamps, O=cri74,
          CN=pc002.vpn1.stagiaires.tst/emailAddress=jbemonet@ext.cri74.org"
esp=3des-sha1
          auto=start

conn dist_cluster
          right=195.202.0.47
          rightsubnet=195.202.0.0/25
```

```
          rightid="C=FR, ST=Haute-Savoie, L=Archamps, O=cri74,
          CN=pc002.vpn2.stagiaires.tst/emailAddress=jbemonet@ext.cri74.org"
          auto=start

conn dist_priv
          right=195.202.0.47
          rightsubnet=10.0.0.0/8
          rightid="C=FR, ST=Haute-Savoie, L=Archamps, O=cri74,
          CN=pc002.vpn2.stagiaires.tst/emailAddress=jbemonet@ext.cri74.org"
        auto=start
```

Sur le serveur (CRI74)

```
conn %default
          keyingtries=0
          disablearrivalcheck=no
          authby=rsasig
          left=%any
          leftid="C=FR, ST=Haute-Savoie, L=Archamps, O=cri74,
          CN=pc002.vpn1.stagiaires.tst/emailAddress=jbemonet@ext.cri74.org"
          leftsubnet=10.0.100.0/24
          esp=3des-sha1
          auto=add

conn dist_cluster
          rightcert=pc002_vpn2_stagiaires_tst.pem
          rightsourceip=195.202.0.47
          right=195.202.0.47
          rightsubnet=195.202.0.0/25
          rightid="C=FR, ST=Haute-Savoie, L=Archamps, O=cri74,
          CN=pc002.vpn2.stagiaires.tst/emailAddress=jbemonet@ext.cri74.org"
          auto=add

conn dist_priv
          rightcert=pc002_vpn2_stagiaires_tst.pem
          rightsourceip=195.202.0.47
          right=195.202.0.47
          rightsubnet=10.0.0.0/8
          rightid="C=FR, ST=Haute-Savoie, L=Archamps, O=cri74,
          CN=pc002.vpn2.stagiaires.tst/emailAddress=jbemonet@ext.cri74.org"
          auto=add
```

4 – Erreurs

Voici des erreurs courantes de configuration :

- « Unable to get my private key » : L'emplacement de la clé privée dans ipsec.secrets est inexacte. ==> Dans le cas de certificats
- « Invalid ID Information » : « leftid » et « rightid » sont faux.
- « Invalid Key Information » : Incohérence entre le contenu des certificats et « leftid » ou « rightid »
- « KLIPS module not loaded » : Cette erreur arrive quand on fait :

```
# ipsec setup --status
```

Il ne faut pas tenir compte de cet erreur. KLIPS n'est pas utilisé sur un kernel 2.6

- « netlink response for Add SA » : Difficile à cerner cette erreur. Elle découle peut-être d'autres erreurs comme par exemple l'oubli d'installation de iproute.

5 – Webliographie

- http://lists.virus.org/users-openswan-0410/
- http://www.strongswan.org
- www.linux-tarn.org/documentation/linuxtarn/Laurent_R/strongswan/html/VPN-STRONGSWAN-GREENBOW.htm
- www.thegreenbow.fr/doc/tgbvpn_cg_Linux_fr.pdf
- www.freeswan.org
- http://sylvestre.ledru.info/howto/securite/tunnels_et_vpn.pdf

6 – Lexique

AES : *AES* est le sigle d'*Advanced Encryption Standard*, en français « *standard de chiffrement avancé* ». Il s'agit d'un algorithme de chiffrement symétrique, choisi en octobre 2000 par le NIST pour être le nouveau standard de chiffrement pour les organisations du gouvernement des États-Unis.

Authentification: Il y a plusieurs types d'authentification. L'authentification des connexions et l'authentification des données. Lors d'une interconnexion entre sites distants, les interlocuteurs sont des équipements réseau, tels que des routeurs et/ou des firewalls. Dans ces conditions, l'authentification consiste à s'assurer que chacune des extrémités de la communication est bien celle qui est attendue, et qu'elle le reste tout au long de sa communication. La procédure doit évidemment être automatisée. En revanche, s'il s'agit de la connexion d'un utilisateur nomade(roadwarrior), il est nécessaire de s'assurer qu'il s'agit bien de l'utilisateur en question(et non plus seulement de son ordinateur) avant « d'ouvrir » le réseau de l'entreprise.

Concernant l'authentification des données, l'objectif est de s 'assurer que l'émetteur des données en question est bien celui qu'il prétend être. Le mécanisme de signature des données permet cela.

Autorité de certification : entité qui crée des certificats. C'est une autorité morale qui définit les règles d'entrée des ressources et des individus dans la PKI. En pratique, il s'agit de définir les critères et les modalités d'attribution de certificats numériques.

Autorité d'enregistrement : entité chargée de recueillir les demandes de certificats et de contrôler l'identité de la personne ainsi que les critères d'attribution.

Bi-clé : couple de clés composé d'une clé privée et d'une clé publique

Blowfish est un algorithme de chiffrement symétrique par blocs conçu par Bruce Schneier en 1993.

Certificat : Une identité électronique qui est émise par une tierce partie de confiance pour une personne ou une entité réseau. Chaque certificat est signé avec la clé privée de signature d'une autorité de certification. Il garantit l'identité d'un individu, d'une entreprise ou d'une organisation. En particulier, il contient la clé publique de l'entité et des informations associées à cette entité.

Certificat Auto signé : un certificat auto signé contient comme tout certificat une clé publique. Sa particularité réside dans le fait que ce certificat est signé avec la clé secrète associée. Dans ce cas précis, l'autorité de certification est donc le détenteur du certificat.

Certificat X.509 : Il s'agit d'une norme sur les certificats largement acceptée et conçue pour supporter une gestion sécurisée et la distribution des certificats numériquement signés sur le réseau Internet sécurisé. Le certificat X.509 définit des structures de données en accord avec les procédures pour distribuer les clés publiques qui sont signées numériquement par des parties tierces.

Certificat X.509v3 : Les certificats X.509v3 ont des extensions de structures de données pour stocker et récupérer des informations pour les applications, des informations sur les points de distribution des certificats, des CRLs et des informations sur les politiques de certification. Chaque fois qu'un certificat est utilisé, les capacités de X.509v3 permettent aux applications de vérifier la validité de ce certificat. Il permet aussi à l'application de vérifier si le certificat est dans une CRL. Ces certificats et CRLs sont normalisés auprès de l'IETF dans la RFC 2459.

Clé : Une quantité utilisée en cryptographie pour chiffrer/déchiffrer et signer/vérifier des données.

Contrôle d'intégrité : Le contrôle d'intégrité a pour objectif de garantir que les données n'ont pas été modifiées (volontairement ou non) au cours de leur transmission. Un code spécifique (MAC : Message Authentication Code) est généré pour chaque message et transmis avec ce dernier . A l'issue de la transmission, le MAC est régénéré par le destinataire et doit correspondre à celui reçu avec le message correspondant.

Un tel système n'est cohérent que si le mécanisme de génération de MACs s'appuie sur un mécanisme de hashage irréversible et dont les probabilités de collisions sont acceptables. C'est la raison pour laquelle les algorithmes tels que SHA-1 et MD5 sont le plus utilisés.

CRL : (Certificat Revocation List) se sont les listes de révocations de certificats.

Clé Publique : quantité numérique, attachée à une ressource ou un individu, qui la distribue aux autres afin qu'ils puissent lui envoyer des données chiffrées ou déchiffrer sa signature.

Clé Privée : quantité numérique secrète attachée à une ressource ou à un individu, lui permettant de déchiffrer des données chiffrées avec la clé publique correspondante ou d'apposer une signature au bas de messages envoyés vers des destinataires.

Le Data Encryption Standard (DES) est une méthode de chiffrement utilisant des clés de 56 bits. Son emploi n'est plus recommandé aujourd'hui, du fait de sa lenteur en logiciel et de son espace de clés trop petit permettant une attaque systématique en un temps raisonnable. Quand il est encore utilisé c'est généralement en Triple DES, ce qui ne fait rien pour améliorer ses performances. DES a notamment été utilisé dans le système de mots de passe Unix.

Le Triple DES (aussi appelé 3DES) est un algorithme de chiffrement symétrique enchaînant 3 applications successives de l'algorithme DES sur le même bloc de données de 64 bits, avec 2 ou 3 clés DES différentes. Même quand 3 clés de 56 bits différentes sont utilisées, la force effective de l'algorithme n'est de que 112 bits et non 168 bits, à cause d'une attaque *rencontre au milieu*. Cette attaque reste cependant peu praticable, en effet elle nécessite un stockage de données de l'ordre de 2^{56} mots de 64 bits, de plus ce stockage doit être « interrogeable » en un temps très court. C'est pour éviter ce genre d'attaque que le Double DES est simplement proscrit et que l'on passe directement à du Triple DES, le Double DES n'assure en effet qu'une force effective de 57 bits.

Diffie-Hellman : W. Diffie et M. Hellman sont les inventeurs de la cryptographie à clé publique; ce concept a été présenté en 1976 comme faisant intervenir une paire de clés (une pour le chiffrement et une pour le déchiffrement). Ils ont démontré qu'on ne pouvait pas déduire la première de la seconde. Il existe beaucoup d'algorithmes basés sur leur travaux mais peu sont sûrs (au le sens mathématique). DH n'est pas utilisé pour chiffrer mais est un algorithme de gestion de clés.

DSA : le Digital Signature Algorithm, plus connu sous le sigle **DSA**, est un algorithme de signature numérique standardisé par le NIST aux États-Unis, du temps où le RSA était encore breveté. Cet algorithme fait partie de la spécification **DSS** pour *Digital Signature Standard* adoptée en 1993 (FIPS 186). Une révision mineure a été publiée en 1996 (FIPS 186-1) et le standard a été amélioré en 2002 dans FIPS 186-2. Il est couvert par la patente n° 5231668 aux USA (26 juin 1991), attribuée à David Kravitz, ancien employé de la NSA.

IGC(Infrastructure de Gestion Clés)

IPSec : IPSec est un protocole (couche 3 modèle OSI) permettant le transport de données chiffrées sur le réseau IP.

Il est définit par la RFC 2401

MD5 : L'algorithme de signature MD5, pour Message Digest 5, est une fonction de hachage encore très populaire, mais qui n'est plus considéré comme un algorithme sûr. On suggère maintenant d'utiliser plutôt un algorithme basé sur le SHA-1 ou de type RIPEMD-160. Le MD5 a été inventé par Ronald Rivest.

Noyau (en anglais « kernel ») : le noyau est la partie fondamentale d'un système

d'exploitation, il est le gestionnaire de ressources de la machine, qui permet aux éléments matériels et logiciels de fonctionner ensemble. Pour ces raisons, il est le premier logiciel chargé en mémoire (hors gestionnaire de boot).

PKI (Public Key Infrastructure), aussi communément appelée IGC (Infrastructure de Gestion de clés) ou ICP (Infrastructure à clés Publiques), est un ensemble de composants physiques (des ordinateurs, des cartes à puces), de procédures humaines (vérifications, validation) et de logiciels (système) en vue de créer et de gérer le cycle de vie des certificats numériques.

RSA : RSA est un algorithme asymétrique de cryptographie à clé publique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Cet algorithme a été décrit en 1977 par Ron Rivest, Adi Shamir et Len Adleman; les lettres **RSA** sont les initiales de leurs noms. RSA a été breveté par le MIT en 1983 aux États-Unis d'Amérique, mais le brevet a expiré le 21 septembre 2000.

SHA-1 : Secure Hash Algorithm Number 1.SHA est un algorithme de hachage

Signature électronique : C'est un mécanisme qui permet de prouver l'identité de l'émetteur d'un message.

- L'expéditeur chiffre l'empreinte du message avec sa clé privée = signature
- Le destinataire déchiffre la signature avec la clé publique de l'expéditeur
- Le destinataire crée une empreinte du message. Si celle-ci est identique à celle déchiffrée alors l'expéditeur est bien celui qu'il prétend être.

Pour la signature, l'algorithme asymétrique couramment utilisé est DSA.

SSL : (Secure Socket Layer), c'est un protocole de sécurisation conçu par Netscape qui se situe entre la couche transport (TCP) et les protocoles de la couche application. Il assure les services de sécurité suivantes : confidentialité, 'intégrité et 'authentification du serveur et du client.

VPN : (Virtual Private Network ou Réseau Privé Virtuel), un réseau privé virtuel repose sur un protocole, appelé protocole de tunnelisation (tunneling), c'est-à-dire un protocole permettant aux données passant d'une extrémité du VPN à l'autre d'être sécurisées par des algorithmes de cryptographie.

Le terme de « tunnel » est utilisé pour symboliser le fait qu'entre l'entrée et la sortie du VPN les données sont chiffrées et donc incompréhensible pour toute personne située entre les deux extrémités du VPN, comme si les données passaient dans un tunnel. Dans le cas d'un VPN établi entre deux machines, on appelle client VPN l'élément permettant de chiffrer les données à l'entrée et serveur VPN (ou plus généralement serveur d'accès distant) l'élément déchiffrant les données en sortie.

De cette façon, lorsqu'un utilisateur nécessite d'accéder au réseau privé virtuel, sa requête va être transmise en clair au système passerelle, qui va se connecter au réseau distant par l'intermédiaire d'une infrastructure de réseau public, puis va transmettre la requête de façon chiffrée. L'ordinateur distant va alors fournir les données au système pare-feu de son réseau local qui va transmettre la réponse de façon chiffrée. À la réception sur le proxy de l'utilisateur, les données seront déchiffrées, puis transmises à l'utilisateur