Chapitre 3

Vue d'ensemble sur la sécurité des réseaux informatique

La suite de protocoles TCP/IP a été conçue pour favoriser la communication entre hôtes distants de bonne volonté, dont le but était d'échanger des informations. Dès que l'utilisation a été largement répandue, comme toujours dans ces cas-là, certaines personnes ont essayé de récupérer des informations qui ne leur étaient pas destinées, soit par malice (pour montrer qu'on peut le faire), soit avec de mauvaises intentions. Il a donc fallu penser à **sécuriser** les réseaux informatique.

3.1 Un exemple de faille de sécurité

Depuis l'adoption d'Unix comme système d'exploitation pratiquement universel et puisqu'il s'agit d'un système d'exploitation multi-utilisateurs, chaque utilisateur doit déclarer un **identificateur** (*identifier* en anglais) et un **mot de passe** (*password* en anglais), ne serait-ce que pour ne pas détruire les fichiers du voisin par maladresse. Cette politique a été adoptée dès l'apparition des premiers réseaux et on demande une **authentification** pour l'accès à toute ressource distante, par exemple la récupération du courrier électronique.

Le problème est que ce mot de passe transite en clair sur le réseau, ce qui veut dire qu'on peut le récupérer en analysant les trames si on se trouve dans le même réseau local, grâce à un analyseur de trames, un peu plus difficilement mais c'est encore possible si on ne se trouve pas dans le même réseau local.

`Wireshark` peut servir à titre pédagogique, comme nous l'avons fait jusqu'à maintenant. Il peut également servir à espionner, par exemple à récupérer les mots de passe. Nous n'allons

donner qu'un seul exemple.

Faites démarrer une capture, appuyer sur le bouton « Recevoir et envoyer » de votre application de courrier électronique préférée puis arrêter la capture. Ceci doit vous donner quelque chose qui ressemble à la figure 3.1.

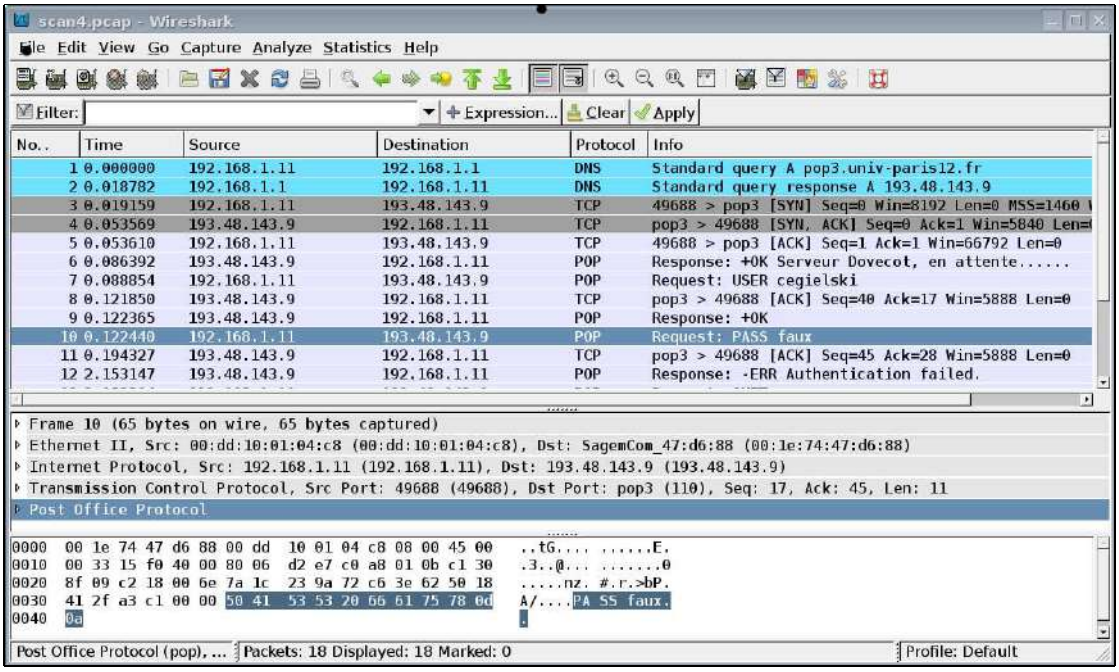

FIG. 3.1 – Problème de sécurité

En général le protocole POP (pour *Post Office Protocol*) est utilisé, que nous ne détaillerons pas ici. On voit que le service DNS est utilisé pour déterminer l'adresse IP du serveur de courrier (en général désigné par un nom), puis TCP pour une ouverture de connexion en trois étapes. Le serveur POP dit alors qu'il est prêt. Le client envoie le nom de l'utilisateur en clair, ce qui est déjà une première faille de sécurité. Le serveur répond qu'un mot de passe est nécessaire pour ce client. Le client envoie celui-ci en clair (ici « faux », qui n'est évidemment pas mon vrai mot de passe ; je l'ai changé temporairement pour les besoins de la cause).

Toute personne munie d'un renifleur peut donc récupérer les mots de passe transitant sur le réseau local (avec un analyseur de trames automatique ou manuellement). Elle peut ensuite récupérer, de n'importe quel endroit, tout le courrier électronique sans même que le destinataire ne s'en aperçoive si elle prend la précaution de laisser le contenu dans la boîte (option de toute application de récupération de courrier électronique actuelle).

Cette faille de sécurité n'est pas trop gênante. La preuve en est que le protocole POP3 est encore largement utilisé. On retrouvait cependant le même problème lors des premières transactions de commerce électronique, avec le code des cartes Visa passé en clair.

3.2 Buts de la sécurité réseau

Le but de la sécurité du réseau est de permettre la disponibilité, l'intégrité et la confidentialité :

- La **disponibilité** signifie que les informations et les services sont accessibles et fonctionnels lorsque nécessaire. Si les systèmes ne sont pas disponibles, les deux autres points n'ont plus vraiment d'importance.
- L'**intégrité** signifie que l'information ou le logiciel est complet, exact et authentique. L'objectif est d'empêcher tout processus ou personne non autorisée d'apporter une quelconque modification, intentionnelle ou volontaire.

  Dans le cas d'une intégrité réseau, il s'agit de s'assurer que le message reçu est bien celui qui a été envoyé. Son contenu doit être intégral et non modifié.
- La **confidentialité** empêche des informations sensibles d'être divulguées sans votre consentement ou d'être interceptée sous forme intelligible.

3.3 Les méthodes fondamentales de sécurisation

Les mécanismes de sécurité des résaux informatique sont très variés mais ils reposent tous sur trois méthodes fondamentales : le chiffrement, le filtrage et les méthodes physiques.

3.3.1 Méthodes physiques

Nous appellerons *méthodes physiques* les méthodes qui ne reposent pas sur l'aspect logiciel. Il s'agit de la façon de protéger le site du réseau local (contrôle d'accès, éventuellement biométrique, gardiennage,...), de protéger la ligne spécialisée entre deux sites (pour empêcher le reniflage), et ainsi de suite.

Il ne s'agit pas d'une spécialité informatique proprement dite. Bien que très importante, nous n'en parlerons pas du tout ici. On peut trouver quelques éléments sur ce point, ainsi qu'une vue d'ensemble de la sécurité des systèmes d'information et des réseaux, par exemple dans [PAN–04].

3.3.2 Le chiffrement

3.3.2.1 Le principe du chiffrement

Le problème.- Supposons qu'un **expéditeur** veut envoyer un **message** à un **destinataire**. Cet expéditeur veut envoyer le message de manière sûre : il veut s'assurer qu'aucune oreille indiscrète ou œil indiscret ne puisse s'informer du mesage.

La solution du chiffrement.- Le message originel à envoyer est appelé **message en clair** ou **texte en clair** (*plain text* en anglais).

Le **chiffrement** (ou **cryptage**) du message originel consiste à le traduire en un message incompréhensible. Le résultat de ce processus de chiffrement est appelé **texte chiffré** (*ciphertext* en anglais), **message codé** ou **cryptogramme**.

Le **déchiffrement** (ou **décryptage**) consiste à traduire le message chiffré en message originel.

Le **système de chiffrement** est la méthode qui permet le chiffrement et de le déchiffrement.

Exemple.- L'un des premiers systèmes de chiffrement est le **chiffre de Jules César**, dans lequel chaque caractère du texte en clair est remplacé par celui qui se trouve trois places plus loin dans l'alphabet modulo 26 ('A' est remplacé par 'D', 'B' par 'E', ..., 'W' par 'Z', 'X' par 'A',...).

Vocabulaire.- L'art et la science de trouver des méthodes de chiffrement est appelée **cryptographie**, ce qui signifie « écriture secrète » au sens étymologique, pratiquée par des **cryptographes**. L'art et la science d'essai de décryptage des messages chiifrés sans connaître le système de chiffrement et la clé utilisés est appelée la **cryptanalyse**, pratiquée par les **cryptanalystes**. La branche des mathématiques qui traite de la cryptographie et de la cryptanalyse s'appelle la **cryptologie**, la science du chiffrement, pratiquée par des **cryptologues**.

Il existe une très longue histoire des méthodes de cryptage, chaque méthode ayant été déjouée à un certain moment.

Remarque. L'expression *texte en clair* pourrait laisser entendre qu'il n'est possible de protéger que des messages textuels. Mais en réalité, la protection s'applique aussi à l'image, à la voix, à des vidéos et à d'autres types de données. Le terme « texte » a une origine historique dans le sens où la cryptographie concernait initialement des objets textuels.

D'un point de vue informatique, on le *message en clair* et le *message codé* sont des suites de bits.

3.3.2.2 Clés

Les processus de chiffrement reposent à la fois sur un algorithme (la méthode de chiffrement), utilisé de manière identique quel que soit le message à traiter, et sur des paramètres, appelé une **clé** (*key* en anglais). Un algorithme identique associé à différentes clès produit différents cryptogrammes à partir du même message en clair. Les méthodes de chiffrement sont peu nombreuses, si bien qu'il est impossible de les garder secrètes. Par conséquent, c'est la clé qui doit être confidentielle.

Exemple.- Par exemple pour la méthode du chiffre de Jules César, la clé peut être le décalage de lettres dans l'alphabet.

Modélisation.- On a :

$$cipher = encrypt(plain,\ key)$$

où $encrypt()$ est la **fonction de cryptage** (*encryption* en anglais), ou plus exactement :

$$cipher = encrypt_{ALG}(plain,\ key)$$

pour indiquer que le cryptage dépend de la méthode de cryptage ALG.

La fonction $encrypt()$ doit être récursive (au sens de calculable sur ordinateur) et doit posséder une fonction réciproque également récursive pour qu'on puisse retrouver le texte en clair à partir du texte codé et de la clé :

$$plain = decrypt_{ALG}(cipher,\ key)$$

On parle de **fonction de décryptage** (*decryption* en anglais).

Dans la pratique, les fonction $encrypt()$ et $decrypt()$ doivent être non seulement récursives mais également rapides.

Méthodes à clés disymétriques.- Dans l'exemple donné du chiffrement de Jules César, la clé est unique pour le chiffrement et le déchiffrement. Il existe des méthodes de chiffrement avec une **clé de chiffrement** et une **clé de déchiffrement** différentes. Le modèle présenté jusqu'à maintenant ne concerne donc que les **méthodes de cryptage à clé symétrique**. À la fin des années

1970 furent présentées des **méthodes de cryptage à clès disymétriques** :

$$plain = decrypt_{ALG}(cipher,\ key2)$$

dans lesquelles la clé pour décrypter est différente de celle pour crypter.

Les différentes méthodes de chiffrement.- Nous avons dit qu'il existe une longue histoire d'invention de méthodes de cryptages car, jusqu'à maintenant, toute méthode proposée a été déjouée un jour ou l'autre. Ceci est également le cas de nos jours : les méthodes utilisées sont suffisamment récentes pour ne pas être encore déjouées mais rien ne certifie qu'elles sont invincibles. Quelques résultats de théorie de la complexité justifie le choix de telle ou telle méthode mais il n'existe pas de théorème disant que telle méthode serait nettement meilleure.

Les détails sur les méthodes de cryptage utilisées en pratique fait l'objet d'un autre cours.

3.3.3 Traduction d'adresse et filtrage

3.3.3.1 Traduction d'adresse

Une méthode utilisée pour faire face au manque d'adresse IP des petits réseaux a trouvé une application dans la sécurité des réseaux car elle ne permet pas d'accéder directement aux machines.

Intérêt de principe de la traduction d'adresse.- Considérons le cas d'un réseau local constitué de six ordinateurs et d'une passerelle. À l'origine, chacune de ces sept machines disposaient d'une adresse IP (visible de tout l'Internet), ce qui exigeait de lui attribuer une plage d'adresse de classe C (pour 255 machines).

Très rapidement est apparue une façon de faire qui évite ce gaspillage. L'administrateur réseau d'un interréseau n'attribue qu'une seule adresse IP à ce réseau local, qui est un sous-réseau de cet interréseau, par exemple 83.213.22.34. L'administrateur du réseau local réserve cette seule adresse **routable** à l'interface réseau de la passerelle dirigée « vers l'extérieur », c'est-à-dire vers l'interréseau et donc, éventuellement, l'Internet. Il attribue, d'autre part, aux six machines du réseau local et à l'interface de la passerelle dirigée « vers l'intérieur » des adresses IP **privées**, par exemple 192.168.1.x : 192.168.1.1 est traditionnellement attribué à la passerelle ; disons que les machines reçoivent les adresses 192.168.1.10 à 192.168.1.15.

Lorsqu'un ordinateur du réseau local, disons celui d'adresse 192.168.1.11, veut envoyer un paquet en dehors du réseau local, sa table locale de routage lui dit de l'envoyer, comme d'habitude, vers la passerelle (d'adresse 192.168.1.1), utilisée comme premier routeur. Celui-ci sait vers quel second routeur envoyer le paquet mais il ne peut pas garder l'adresse 192.168.1.11 comme adresse IP source sur Internet : il remplace donc celle-ci par sa propre adresse « externe », à savoir 83.213.22.34. On parle alors de **traduction d'adresses réseau** (**NAT** pour *Network Address Translation*). La passerelle joue alors non seulement le rôle de routeur mais également de **serveur NAT**, avec la partie logicielle correspondante.

Dans le cas où l'ordinateur se contente d'envoyer un paquet sans attendre de réponse, nous avons tout dit. Dans le cas contraire, le destinataire envoie une réponse sur l'interréseau, qui arrive à la passerelle, puisque c'est l'adresse source indiquée. La passerelle doit faire suivre ce paquet à l'ordinateur local, mais comment savoir duquel il s'agit ? Pour cela, on ne va se contenter de regarder l'adresse IP mais également le port (paramètre de la couche de transport). Puisque nous avons parlé d'un requête, rappelons que le sous-système réseau de l'ordinateur a attribué un port éphémère au paquet envoyé. Le serveur NAT doit donc maintenir une **table de correspondance des ports**, qui dit que tel port éphémère correspond à telle machine de son réseau local. Il lui

suffit donc de traduire le paquet en changeant l'adresse IP source et de l'envoyer sur son réseau local.

En fait deux machines pourraient attribuer le même port éphémère, ce qui conduirait à un conflit lors d'une réponse. Pour éviter ce problème, le serveur NAT traduit, non seulement l'adresse IP, mais également le port source lors de l'envoi, tenant une correspondance entre le couple adresse/port des ordinateurs locaux et le port éphémère qu'il attribue (dans sa table de correspondance des ports). Lors d'une réponse, le champ port destination lui permet de retrouver l'adresse IP source et le port source qu'il doit changer dans le paquet avant de le renvoyer sur le réseau local.

Cas des serveurs locaux.- Le processus décrit ci-dessus est suffisant pour faire face à un manque d'adresses routables pour un petit réseau. On peut cependant généraliser le principe au cas des serveurs locaux. Imaginons que l'on veuille installer un serveur Web sur ce petit réseau local. Les requêtes parviendront avec 83.213.22.34 comme adresse IP et 80 comme port. Le service Web peut être installé sur la passerelle (dans le cas où il s'agit d'un ordinateur) et alors il n'y a rien de plus à dire.

En général, on préfère cependant déporter ce service sur un ordinateur du réseau local que l'on spécialise dans ce service. Dans ce cas, on peut paramètrer le serveur NAT pour q'il renvoie tout paquet dont le champ porte source est égal à 80 à cet ordinateur spécialisé (en changeant l'adresse IP source mais pas le port source en général).

Application à la sécurité.- Le service NAT, conçu à l'origine pour pallier au manque d'adresse IP routable, a vu son champ d'application s'élargir avec les problèmes de sécurité des réseaux informatique : comme les ordinateurs externes au réseau local ne peuvent pas accèder directement aux ordinateurs du réseau local, ils ne peuvent pas initier de session (sauf dans le cas des serveurs), et on s'est aperçu qu'il ne peuvent donc pas les attaquer facilement.

3.3.3.2 Le filtrage

Lorsqu'on s'est aperçu que le service NAT a pour conséquence de renforcer la sécurité des réseaux, on a généralisé le principe sous la forme du *filtrage*.

Filtrage des ports.- Une première étape consiste à écarter tout paquet entrant sur le réseau local pour accéder à certains services (donc dont les ports correspondent à certains ports bien connus, comme ceux correspondants à Telent, FTP ou SMTP) ou, le plus souvent, à tout port (bien connu) sauf ceux explicitement autorisés. La machine, en général la passerelle, chargée de ce filtrage est appelée **pare-feu** (*firewall* en anglais) de filtrage de segments.

On a généralisé de même pour les paquets sortants.

Filtrage des adresses IP.- On peut de même filtrer les adresses IP entrantes ou sortantes, grâce à un pare-feu de filtrage de paquets.

3.3.3.3 Serveur proxy

Un **serveur proxy** est un paquetage logiciel et/ou matériel qui permet de mettre en cache des pages Web. Il s'agit en général d'un ordinateur du réseau local à deux interfaces réseau : l'une servant à l'accès Internet, l'autre menant au réseau local. Le serveur proxy traite les requêtes Web des ordinateurs du réseau local : lorsqu'un ordinateur du réseau local émet une requête HTTP, la requête est récupérée par le serveur proxy, le serveur cherche si la page est déjà présente dans le cache (ce qui élimine le temps du chargement) et sinon est retransmise avec l'adresse publique

du proxy.

La mise en cache est divisée en deux groupes : avec une mise en cache **active**, le serveur proxy récupère les documents dont il pense qu'ils pourront être demandés par les clients ; la mise en cache **passive** attend l'arrivée d'une requête avant de récupérer le document, après quoi le serveur décide si les données doivent être mises en cache.

Les serveurs proxy, à l'origine utilisés pour accélérer la récupération de documents ou pour faire face au manque d'adresses IP publiques (comme pour les serveurs NAT), sont également utilisés pour sécuriser les réseaux informatique car la requête semble provenir du serveur, la destination n'ayant aucunement conscience du réseau situé derrière le proxy.

3.4 Panorama de quelques solutions

La sécurité des réseaux informatique est un domaine très actif. Plusieurs solutions ont été apportées. Énumérons quelques-unes de ces solutions, sans entrer dans le détail : la description de chaque solution sérieuse est pratiquement l'objet d'un cours complet. Il est traditionnel de classifier les solutions suivant la couche à laquelle elle s'adresse, en commençant par la couche la plus élevée.

3.4.1 Sécurité de la couche d'application

Intérêts et inconvénients.- Un mécanisme de sécurité au niveau de la couche d'application procure une sécurité point à point entre une application s'exécutant sur un hôte *via* le réseau jusqu'à l'application sur un autre hôte. Il n'a pas besoin de tenir compte du mécanisme de transport sous-jacent. Ce n'est cependant pas une solution universelle, puisque chaque serveur et chaque client de l'application doivent être adaptés.

Quelques exemples.- Il existe plusieurs essais de tels mécanismes de sécurité. Citons-en un largement utilisé et un autre maintenant presqu'abandonné.

- **PGP** (pour *Pretty Good Privacy*) est un produit utilisé pour la confidentialité et la signature numérique des messages électroniques, créé en 1991 par Phil Zimmermann. Il procure une sécurité point à point pour le transport de fichiers de l'expéditeur au destinataire, et peut également être utilisé pour chiffrer les fichiers.
  Obtenir la clé publique de l'interlocuteur peut être délicat à réaliser de façon fiable. PGP utilise un modèle dit **toile de confiance** ou **réseau de confiance**, dans lequel tout utilisateur peut attester de l'identité d'un autre utilisateur. Vous ne devez jamais avoir confiance dans une clé dont la filiation vous est inconnue.
- **S-HTTP** (pour *Secure Hyper Text Transport Protocol*) a été conçu pour offrir une sécurité aux applications fondées sur le web. Il étend HTTP en ajoutant des balises pour des transactions chiffrées et fiables. Le serveur S-HTTP négocie avec le client le type de chiffrement utilisé. S-HTTP n'exige pas que les clients possèdent des certificats de clé publique, car il utilise des clès symétriques, transmises à l'avance à l'aide d'une communication hors-bande.

  S-HTTP n'est pas largement utilisé, HTTPS reposant sur SSL l'ayant détrôné.

3.4.2 Sécurité de la couche de transport

Intérêts et inconvénients.- De nombreux mécanismes de sécurité au niveau de la couche de transport demandent la modification des applications (HHTPS au lieu de HTTP, par exemple) afin

d'obtenir les avantages de la sécurité. Les applications sécuriéses interviennent en remplacement des applications non sécuriées par le fait que les serveurs utilisent des ports différents.

Quelques exemples.- Il existe plusieurs essais de tels mécanismes de sécurité. Citons-en trois largement utilisés.

- **SSL** (*Secure Sockets Layer*) a été conçu par Netscape (en 1994, à l'époque où son navigateur web était largement prédominant) et est largement utilisé sur Internet pour des transactions web telles que l'envoi de données de carte de crédit, avec **HTTPS**. En fait SSL est plus général et peut également être utilisé pour d'autres protocoles d'application comme Telnet, FTP, LDAP, IMAP et SMTP, mais les versions sécurisées grâce à SSL n'ont pas connu de succès.

  TLS (*Transport Layer Scurity*) est un standard ouvert proposé à l'IETF fondé sur SSL 3.0, défini par [RFC 2246, RFC 2712, RFC 2817, RFC 2818].

  SSL et TLS ne procurent la sécurité qu'à une session TCP à la fois, sur laquelle n'importe quelle quantité de données peut être envoyée en toute sécurité. Le serveur et le navigateur doivent être activés SSL ou TLS pour qu'une connexion web sécurisée puisse être établie.
- **SSH** (*Secure SHell*) est un protocole, spécifié dans un ensemble de documents de brouillon Internet, qui procure une ouverture de session à distance sécurisée, qui remplace avantageusement *telnet* (voir [B-S–01]).
- Le **filtrage** permet de bloquer certains segments et datagrammes sur des appareils de couche de transport. Les décisions de routage ou d'abandon sont fondées sur les règles d'une **liste de contrôle d'accès** (**ACL** pour *Access Control List*), dites **étendues** car les listes de contrôle d'accès proprement dit concernent la couche réseau.
  Les options de filtrage TCP comprennent les connexions établies, les numéros de port ou plages de numéros de port, ainsi que les valeurs des types de service.
  Les options de filtrage UDP s'effectuent sur les numéros de port.

3.4.3 Sécurité de la couche réseau

Intérêts et inconvénients.- Les mécanismes de sécurité au niveau de la couche réseau sécurisent le trafic pour toutes les applications et protocoles de transport des couches supérieures. Les applications n'ont pas à être modifiées.

Quelques exemples.- Il existe essentiellement deux tels mécanismes de sécurité largement utilisés.

- Les protocoles **IPSec** (*IP SECurity*, décrit dans [RFC 2401]) peuvent fournir le contrôle d'accès, l'authentification, l'intégrité des données et la confidentialité pour chaque paquet IP entre deux nœuds réseau (hôte ou passerelle). Aucune modification du matériel ou du logiciel réseau n'est nécessaire pour router IPSec. Les applications et les protocoles de niveau supérieur peuvent rester inchangés.

  IPSec ajoute deux protocoles de sécurité à IP : **AH** (*Authentification Header*) et **ESP** (*Encapsulating Security Payload*).

  IPSec est à la bse des **réseaux virtuelx** (`VPN` pour *Virtual Private Net*) qui permettent d'accéder à l'intranet de son université ou de son entreprise sur un site délocalisé.
- Le **filtrage** permet de bloquer certains paquets au niveau des routeurs ou autres appareils de couche réseau. Les décisions de routage ou d'abandon sont fondées sur les règles d'une **liste de contrôle d'accès** proprement dites (elles sont dites étendues lorsqu'elles concernent également la couche de transport).
  Les listes d'accès standard effectuent un filtrage d'après l'adresse source. Les listes d'accès avancées peuvent effectuer un filtrage d'après les protocoles ICMP, IGMP ou IP de la

couche réseau.

3.4.4 Sécurité de la couche d'accès

La sécurité pour la couche d'accès est effectuée entre deux points, par exemple sur une ligne louée ou un circuit virtuel permanent. Des périphériques matériels dédiés sont situés à chaque extrémité du lien pour effectuer le chiffrement et le déchiffrement. Cette solution n'est pas applicable à de grands intérréseaux puisque les paquets ne peuvent pas être routés sous leur forme chiffrée.

Ces mécanismes sont surtout susceptibles d'être utilisés par les militaires et les organisations financières comme les banques.

Nous venons de citer quelques mécanismes de sécurité des réseaux informatique (PGP, SSL, filtrage, IPSec,...). Nous avons dit également que l'étude de chacun d'entre eux donnerait lieu à un cours complet. Nous devons donc en choisir un ou deux à titre d'exemple.