# SUPPORT DE COURS DE SÉCURITÉ INFORMATIQUE ET CRYPTO.

Raphael Grevisse Yende

# SUPPORT DE COURS DE SECURITE INFORMATIQUE ET CRYPTO.


## YENDE RAPHAEL Grevisse, Ph.D.

**Docteur en Télécoms et Réseaux Inf.**


**Cours dispensé aux Facultés Africaine BAKHITA en Première Licence : Réseaux informatiques.**

# BIBLIOGRAPHIE


- **Jain, R. M. Bolle, S. Pankanti**, "*Biometrics: Personal Identification in Networked Society*" , Kluwer Academic Press, 1998.
- **R.M. Bolle, J.H. Connell, S. Pankanti, N.K. Ratha, A.W. Senior**, "*Guide to Biometrics*", Springer-Verlag, New York, 2004
- **D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar**, *"Handbook of Fingerprint Recognition",* Springer-Verlag, New York, 2003.
- **H. NSENGE MPIA**, « *Sécurité Informatique »*, Institut Supérieur Emmanuel d'Alzon / Butembo, cours inédit, 2017.
- **LESCOP Yves, «** *Sécurité Informatique***»,** 2002
- **Simon Singh,** « *Histoire des codes secrets* », JC Latt_es _editeur, 1999.
- **Claude Shannon,** "*A mathematical theory of communication*", Bell system technical journal, 1948.
- **Touradj Ibrahimi, Franck Leprevost, Bertrand Warusfel**, « *Enjeux de la sécurité multimedia »*, Hermès Science Lavoisier, 2006
- **Laurent BLOCH et Christophe WOLFHUGEL**, « *Sécurité informatique. Principes et méthode à l'usage des DSI, RSSI et administrateurs »,* 2e édition, Eyrolles, Paris, 2009.
- **Mark Allen Ludwig**, « *Naissance d'un virus : Technologie et principes fondamentaux »*, Diff. Bordas, 1993, 47 p
- **Peter Szor,** *"The Art of Computer Virus Research and Defense"*, Addison-Wesley Professional, 2005, 744 p
- **J. Dressler**, "*Cases and Materials on Criminal Law"*, Thomson/West, 2007 « United States v. Morris »
- **Greg Hoglund et James Butle**r; « *Rootkits : infiltrations du noyau Windows* », Paris, Campus Press, 2006 (1re éd. 2005), 338 p.
- **Ric Vieler,** "Professional rootkits*, Indianapolis, IN, Wiley/Wrox, coll. « Wrox professional guides »",* 2007, 334 p.
- **William Stallings**, "*Cryptography and Network Security: Principles and Practice"*, 3eme ed. Prentice Hall, 2003.

# TABLE DES MATIERES

# AVERTISSEMENTS

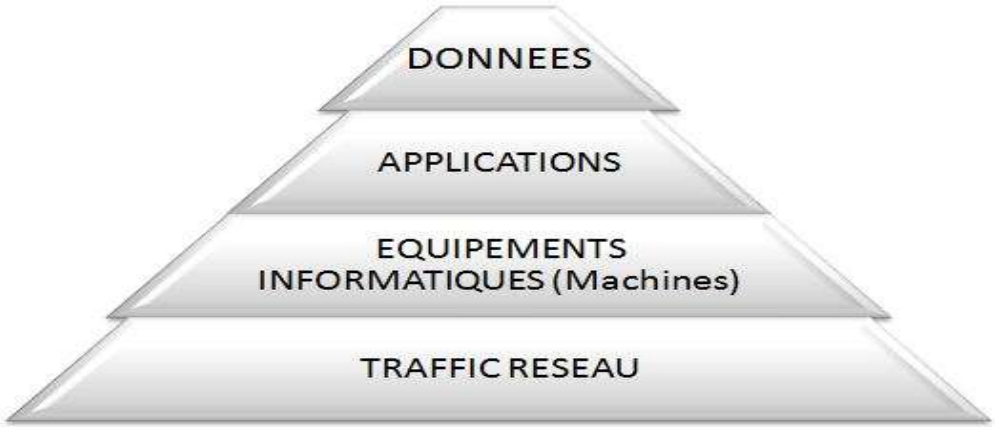
Ce support de *« SECURITE INFORMATIQUE & CRYPTOLOGIE » du Docteur YENDE RAPHAEL Grevisse »,* demande avant tout, un certain entendement de l'informatique et des connaissances de base de sécurité des réseaux informatiques et principalement une prédisposition d'analyse inéluctable et cartésienne ; Vu que l'apport de ce cours, met l'accent sur les concepts de la protection des systèmes d'information reposant sur une compréhension technique approfondie de la gestion des matériels informatiques et leurs modes de communication modernes. Le cours de sécurité informatique et cryptologie des systèmes d'information se veut pour objectif primordial de donner aux étudiants ayant participés des techniques des conceptions et de exécutions des cryptosystèmes, de l'antiquité à nos jours, afin que ces derniers puissent assimiler la notion de sécurité dans la mise en place des systèmes informatiques dont ils ont vocation d'implémenter.



# INTRODUCTION

De nos jours, Le monde connaît des avancées très significatives dans le domaine informatique ; les besoins en matière de sécurité sont un peu plus impérieux, et la prédisposition n'est forcément pas à la baisse. Depuis quelques années déjà, on participe à un changement constant des techniques, qu'il s'agisse des techniques visant à sécuriser l'échange des données ou des techniques de mises au point pour contourner les systèmes sécurisés. D'où, la sécurité des données tend à s'améliorer. Et comme prône ce proverbe chinois : « *l'art de la guerre est basé sur la tromperie* », de même par analogie, la sécurité informatique doit représenter une stratégie qui éradique cette tromperie.

Il est sans ignorer que, le matériel informatique est quasiment partout. En effet, d'une part le matériel est accessibles à un prix très abordable, et d'autre part, les logiciels tendent à se simplifier et permettent une prise en main rapide. En plus, les entreprises, informatisées, nécessitent un réseau sécurisé pour le transfert des données aussi bien entre les machines de ladite entreprise qu'avec des machines externes. Cela étant, la sécurité de façon générale est présente à plusieurs niveaux, qu'il s'agisse des différentes portées de l'information. La sécurité est à prendre dans sa totale dimension comme l'illustre la figure ci-après:




***Niveaux de la sécurité informatique***

La sécurité informatique s'intéresse à la protection contre les risques liés à l'informatique ; elle doit prendre en compte :

- les éléments à protéger : matériels, données, utilisateurs ;
- leur vulnérabilité ;
- leur sensibilité : quantité de travail impliqué, confidentialité…
- les menaces qui pèsent sur eux
- les moyens d'y faire face (préventifs et curatifs) : complexité de mise en œuvre, coût…

# OBJECTIFS DU COURS

L'objectif général de ce cours est d'offrir aux étudiants ayant participés à cet enseignement, des méthodes et techniques de conception et des réalisations de protections des systèmes informatiques afin de les permettre d'intégrer les différentes notions et concepts de la sécurité dans la mise en place des systèmes informatiques dont ils sont (seront) appelés à implémenter dans leurs activités quotidiennes.

De manière spécifique, ce cours de sécurité vise à :

- Fournir les concepts relatifs à la sécurité informatique en se focalisant sur les risques et les défauts de sécurité existants ainsi que l'établissement d'une bonne stratégie de la politique de sécurité ;

- Comprendre les différentes failles de sécurité, pouvant résulter des systèmes de communications informatiques, au moyen de la piraterie, des attaques ainsi que de l'espionnage informatique ;

- Appréhender d'une façon générale, les notions liées à la cryptographie et à la cryptanalyse des informations des systèmes traditionnels jusqu'aux systèmes dits modernes (informatiques), en y appliquant les méthodes correspondantes ;

- Acquérir les notions et les concepts liés à la biométrie informatique, son principe de fonctionnement et les diverses techniques utilisées par cette dernière.

**YENDE RAPHAEL Grevisse, PhD.**
***Professeur associé***

# PREMIER CHAPITRE – GENERALITES SUR LA SECURITTE INFORMATIQUE

## I.1. DEFINITION & CONTEXTE D'ETUDES

*La sécurité informatique* c'est l'ensemble des moyens mis en œuvre pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles. L'objectif de la sécurité informatique est d'assurer que les ressources matérielles et/ou logicielles d'un parc informatique sont uniquement utilisées dans le cadre prévu et par des personnes autorisées.

Il convient d'identifier les exigences fondamentales en sécurité informatique, qui caractérisent ce à quoi s'attendent les utilisateurs de systèmes informatiques au regard de la sécurité :

- **La confidentialité -** Seules les personnes habilitées doivent avoir accès aux données. Toute interception ne doit pas être en mesure d'aboutir, les données doivent être cryptées, seuls les acteurs de la transaction possèdent la clé de compréhension.

- **L'intégrité -** Il faut garantir à chaque instant que les données qui circulent sont bien celles que l'on croit, qu'il n'y a pas eu d'altération (volontaire ou non) au cours de la communication. L'intégrité des données doit valider l'intégralité des données, leur précision, l'authenticité et la validité.

- **La disponibilité -** Il faut s'assurer du bon fonctionnement du système, de l'accès à un service et aux ressources à n'importe quel moment. La disponibilité d'un équipement se mesure en divisant la durée durant laquelle cet équipement est opérationnel par la durée durant laquelle il aurait dû être opérationnel.

- **La non-répudiation -** Une transaction ne peut être niée par aucun des correspondants. La non-répudiation de l'origine et de la réception des données prouve que les données ont bien été reçues. Cela se fait par le biais de certificats numériques grâce à une clé privée.

- **L'authentification -** Elle limite l'accès aux personnes autorisées. Il faut s'assurer de l'identité d'un utilisateur avant l'échange de données.

Bref, on mesure la sécurité d'un système entier à la sécurité du maillon le plus faible. Ainsi, si tout un système est sécurisé techniquement mais que le facteur humain, souvent mis en cause, est défaillant, c'est toute la sécurité du système qui est remise en cause.

# I.2. ETUDES DES RISQUES LIES A LA SECURITE INFORMATIQUE

Les coûts d'un problème informatique peuvent être élevés et ceux de la sécurité le sont aussi. Il est nécessaire de réaliser une analyse de risque en prenant soin d'identifier les problèmes potentiels avec les solutions avec les coûts associés. L'ensemble des solutions retenues doit être organisé sous forme d'une politique de sécurité cohérente, fonction du niveau de tolérance au risque. On obtient ainsi la liste de ce qui doit être protégé. Il faut cependant prendre conscience que les principaux risques restent : *câble arraché, coupure secteur, crash disque, mauvais profil utilisateur* … Voici quelques éléments pouvant servir de base à une étude de risque :

- Quelle est la valeur des équipements, des logiciels et surtout des informations ?
- Quel est le coût et le délai de remplacement ?
- Faire une analyse de vulnérabilité des informations contenues sur les ordinateurs en réseau (programmes d'analyse des paquets, logs…).
- Quel serait l'impact sur la clientèle d'une information publique concernant des intrusions sur les ordinateurs de la société ?

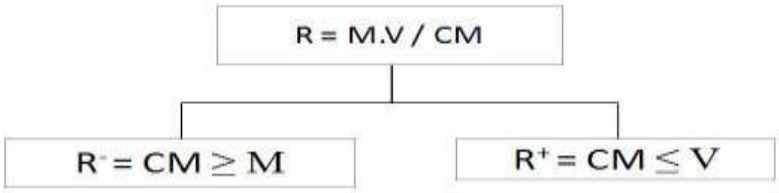
En fait, avec le développement de l'utilisation d'internet, nombre d'entreprises ouvrent leur système d'information à leurs partenaires ou leurs fournisseurs, elles sont plus au niveau de l'architecture trois tiers ou n-tiers. Il devient donc nécessaire de connaître les ressources de l'entreprise à protéger et de maîtriser le contrôle d'accès et les droits des utilisateurs du système. En revanche, la sécurité est un compromis entre *coûts, risques et contraintes*. On comprendra mieux le poids d'un risque en se fiant à la formule suivante :

$$\boldsymbol{Risque} = \frac{\textbf{Menace x Vulnerabilite}}{\textbf{Contre mesure}}$$

- **Risque** - C'est la probabilité qu'une menace exploite une vulnérabilité. Autrement dit, c'est une possibilité qu'un fait dommageable se produise.
- **Vulnérabilité** - C'est une faiblesse inhérente à un système (software ou hardware). Appelée parfois faille ou brèche, elle représente le niveau d'exposition face à la menace dans un contexte particulier.
- **Menace** - c'est le danger (interne ou externe) tel qu'un hacker, un virus, etc.
- **Contre-mesure** - c'est un moyen permettant de réduire le risque dans une organisation.

Conséquences de la formule:

- Le risque est d'autant plus réduit que les contre-mesures sont nombreuses ;
- Le risque est plus important si les vulnérabilités sont nombreuses.




L'utilisation de l'outil informatique est susceptible de nous exposer aq plusieurs types des risques. Il importe donc de pouvoir mesurer ces risques en fonction de la probabilité ou de la fréquence de leurs survenances et aussi en mesurant leurs effets possibles. Ces effets peuvent avoir des conséquences négligeables ou catastrophiques :

- Le traitement informatique en cours échoue : il suffit de le relancer, éventuellement par une autre méthode si on craint que la cause ne réapparaisse ;
- L'incident est bloquant et on doit procéder à une réparation ou une correction avant de poursuivre le travail entrepris.

Il est cependant à noter que ces mêmes incidents peuvent avoir des conséquences beaucoup plus fâcheuses :

- Données irrémédiablement perdues ou altérées, ce qui les rend inexploitables par la suite ;
- Données ou traitements durablement indisponibles, pouvant entrainer l'arrêt d'une production ou d'un service ;
- Divulgation d'informations confidentielles ou erronées pouvant profiter à des sociétés concurrentes ou nuire à l'image de marque de l'entreprise ;
- Déclenchement d'actions pouvant provoquer des accidents physiques ou induire des humains.

## I.2.1. TYPOLOGIE DES RISQUES INFORMATIQUES

En sécurité informatique, il existe deux grands types des risques à savoir : *les risques humains et les risques matériels.*

### A. RISQUES HUMAINS

Ce sont les plus importants, même s'ils sont le plus souvent ignorés ou minimisés. Ils concernent les utilisateurs mais également les informaticiens eux-mêmes. On peut citer :

- **La maladresse** – commettre des erreurs ou exécuter de traitement non souhaité, ou effacer involontairement des données ou des programmes ; etc.

- **L'inconscience et l'ignorance** – introduire des programmes malveillants sans le savoir (par exemple lors de la réception du courrier). Des nombreux utilisateurs d'outils informatiques sont encore inconscients ou ignorants des risques qu'ils font courir aux systèmes qu'ils utilisent. Réaliser des manipulations inconsidérées (autant avec des logiciels qu'avec du matériel) ;

- **La malveillance** – ces dernières années, il est impossible d'ignorer les différents problèmes de virus et des vers. Certains utilisateurs peuvent volontairement mettre en péril le système d'informations, en y introduisant en connaissance de cause de virus ou en introduisant volontairement des mauvaises informations dans une base des données. On parle même de la « *cybercriminalit*é » ;

- **L'ingénierie sociale** – une méthode pour obtenir d'une personne des informations confidentielles, que l'on n'est pas normalement autorisé à obtenir, en vue de les exploiter à d'autres fins. Elle consiste à :

  → Se faire passer pour quelqu'un que l'on n'est pas (en général un administrateur réseau) ;

  → Demander des informations personnelles (nom de connexion, mot de passe, données confidentielles, etc.) en intervenant un quelconque prétexte (problème dans le réseau, modification de celui-ci, etc.) ;

Elle peut se faire soit au moyen d'une simple communication téléphonique ; soit par mail, soit en se déplaçant directement sur place.

- **L'espionnage** – surtout industriel, emploie les même moyens, ainsi que bien d'autres, pour obtenir des informations sur des activités concurrentes, procédés de fabrication, projets en cours, futurs produits, politique de prix, clients et prospects, etc.

## B. RISQUES MATERIELS

Ils sont liés aux défauts et pannes inévitables que connaissent tous les systèmes matériels et logiciels. Ces incidents sont plus ou moins fréquents selon les soins apportés lors de la fabrication et de l'application des procédures de tests effectués avant que les ordinateurs et les programmes ne soient mis en service. Certaines de ces pannes ont des causes indirectes, voire très indirectes, donc difficiles à prévoir. On peut citer :

- **Les incidents liés au matériel** – la plupart des composants électroniques modernes produits en grandes séries, peuvent comporter des défauts de fabrication. Ils finissent un jour ou l'autre par tomber en panne. Certains de ces pannes sont assez difficiles à déceler car intermittentes ou rares. Parfois, elles relèvent d'une erreur de conception.

- **Les incidents liés au logiciel** – ce sont les plus fréquents. Les systèmes d'exploitation et les programmes sont de plus en plus complexes car ils font de plus en plus de choses. Ils nécessitent l'effort conjoint de dizaines, de centaines, voire de milliers de développeurs. Ces derniers peuvent faire des erreurs de manière individuelle ou collective que les meilleures méthodes de travail et les meilleurs outils de contrôle ou de test ne peuvent pas éliminer en totalité.

- **Les incidents liés à l'environnement** – les machines électroniques les réseaux de communication sont sensibles aux variations de températures ou de l'humidité ainsi qu'aux champs électromagnétiques. Dès lors, il est possible qu'un ordinateur tombe en panne de manière définitive ou intermittente à cause des conditions climatiques inhabituelles ou par l'influence d'installations électriques notamment industrielles.

## I.2.2. GESTION DES RISQUES INFORMATIQUES

La gestion des risques informatiques est un ensemble d'opérations de gérer et de diriger les différentes incidences liées à la manipulation de l'outil informatique. La gestion des risques consiste en trois actions majeures :

- Etudier les risques potentiels (identifier/mettre au jour ces risques) ;
- Imposer des règles de sécurité adéquates pour réduire ces risques ;
- Formation des utilisateurs.

## A. ETUDIER LES RISQUES POTENTIELS

Cette phase consiste à faire un examen intégral de la méthodologie de l'étude des risques informatique en vigueur. Cela se matérialise aux moyens :

- **Définition de l'environnement** - Définition des acteurs et leurs intérêts ; Importance de la sécurité dans la stratégie de l'entreprise ; Type de données impliquées ; Visibilité extérieure de la sécurité (importance pour la clientèle, le public).

- **Etude des menaces** - Identifier la nature de la menace: accidentelles (désastre, bugs…) ou intentionnelles (attaques, vols…) ; S'enquérir des sources de la menace: personnel non autorisé, intrus, logiciel ; Localiser la menace : procédures manuelles, informatique (software, réseau, stockage, hardware), infrastructure (concrète et abstraite).

- **Etude des vulnérabilités** - Etudes des faiblesses engendrées par l'exécution d'une menace.

- **Etude des risques** - Probabilité d'occurrence de ces menaces conduisant à une vulnérabilité.

- **Estimation du risque et du plan stratégique** - *Risque* (Coût des pertes à court, moyen et long terme engendrées, Coût de la mise en place de la contre-mesure tant au niveau logique que logistique, Comparer la perte potentielle au coût de la contre-mesure) ; *Plan stratégique* (Planning de l'implémentation avec prise en compte des besoins futurs en termes de sécurité ou non, Planning du suivi de l'implémentation).

- **Mise en place du plan de sécurité** - Les mécanismes de sécurité mis en place peuvent gêner les utilisateurs et les consignes et règles y définies peuvent devenir de plus en plus compliquées au fur et à mesure que le réseau s'étend. Ainsi, la sécurité informatique doit être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils puissent utiliser le système d'information en toute confiance. Raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité dont la mise en œuvre s'effectue en quatre phases:

  → Identifier les besoins en terme de sécurité, les risques informatiques pesant sur l'entreprise et leurs éventuelles conséquences ;

  → Elaborer des règles et des procédures à mettre en œuvre dans les différents services de l'organisation pour les risques identifiés ;

  → Surveiller et détecter les vulnérabilités du système d'information et se tenir informé des failles sur les applications et matériels utilisés ;

  → Définir les actions à entreprendre et les personnes à contacter en cas de détection d'une menace.

- **Audit de sécurité** - L'audit de sécurité consiste à s'appuyer sur un tiers de confiance (de préférence une société spécialisée en sécurité informatique) afin de valider les moyens de protection mis en œuvre, au regard de la politique de sécurité. En fait, l'objectif de l'audit est ainsi de vérifier que chaque règle de la politique de sécurité est correctement appliquée et que l'ensemble des dispositions prises forme un tout cohérent.

## B. IMPOSER DES RÈGLES DE SÉCURITÉ ADÉQUATES

Ceci consiste en la définition de procédures internes à l'entreprise basées sur :

- **Des règles administratives** - Suivre des standards de sécurité (normes ISO) ; Suivre les lois.
- **Des règles physiques** - Gardes, caméras, alarmes, verrous et Accès aux locaux sécurisés par biométrie.
- **Des règles techniques** - Déterminer des niveaux de classification des données ; Définir des niveaux d'accès à ces données ; Utiliser la cryptographie pour le traitement et le stockage de l'information ; Mettre en place un firewall matériel et/ou logiciel, ...

## C. FORMATION DES UTILISATEURS

Il est de plus en plus admis que la sécurité est essentielle. Les coûts engendrés par les pertes de données dues aux attaques réseaux et autres malwares diminuent sensiblement d'années en années[1]. Il est beaucoup plus simple de corrompre l'utilisateur et ce qui l'entoure que l'algorithme de chiffrement utilisé comme par exemple :

- L'utilisateur ne connait pas les risques engendrés par la conservation de la liste des mots de passe utilisés à coté de l'ordinateur ;
- Il est souvent plus simple de s'introduire dans l'ordinateur de l'utilisateur afin de retrouver le texte en clair (hacking, vol, . . . ) ;
- Il est possible de l'espionner, le pousser à la délation, pratiquer le shoulder-surfing ou tout autre technique dite de "social engineering", ...

Il ne s'agira donc pas ici d'expliquer aux employés comment fonctionnent les algorithmes qu'ils utiliseront, mais plutôt comment et dans quelles conditions ils devront les utiliser en définissant des règles qui ne devront pas être transgressées. Il y a également plusieurs manières de réagir à un risque, des plus « sûres » aux plus inconscientes :

- Transférer les risques à une compagnie d'assurances ;
- Réduire les risques en implémentant des contre-mesures qui peuvent être :

  → *Dissuasives* : empêcher une attaque ;
  → *Préventives* : faire échouer une attaque ;
  → *Correctrices* : réduire les dommages causés par une attaque ;
  → *Ignorer/Négliger les risques* ;
  → Accepter les risques si les contre-mesures sont trop onéreuses

Certes, il y a toujours un risque, aussi infime soit-il. Il faudra donc peser le pour et le contre lors de la mise en place éventuelle d'une contre-mesure. Toutefois, en 2007, on remarque une remontée de la somme totale des pertes, due à la fraude financière.

[1] Laurent BLOCH et Christophe WOLFHUGEL, *Sécurité informatique. Principes et méthode à l'usage des DSI, RSSI et administrateurs,* 2e édition, Eyrolles, Paris, 2009.

## I.3. ETABLISSEMENT ET ELEMENTS D'UNE POLITIQUE DE SECURITE INFORMATIQUE

L'élément de politique de sécurité est l'ensemble des orientations suivies par une organisation en termes de sécurité. Elle est élaborée au niveau de système de pilotage (Direction), car elle concerne tous les utilisateurs du système. La sécurité informatique de l'entreprise repose sur une bonne connaissance des règles par les employés, grâce à des actions de formation et de sensibilisation auprès des utilisateurs, mais elle doit aussi aller au-delà de cela tout en couvrant les champs ci-après:

- Mise en place des correctifs ;
- Définition de la police de sécurité ;
- Objectifs, Portée, Responsables ;
- Une stratégie de sauvegarde correctement planifiée ;
- Description de la sécurité (de l'infrastructure physique, des données informatiques, des applications, du réseau) ;
- Plan en cas de sinistre (Un plan de reprise après incident) ;
- Sensibilisation du personnel aux nouvelles procédures
- Sanctions en cas de manquements.

Suite à l'étude des risques et avant de mettre en place des mécanismes de protection, il faut préparer une politique à l'égard de la sécurité. C'est elle qui fixe les principaux paramètres, notamment les niveaux de tolérance et les coûts acceptables. Voici quelques éléments pouvant aider à définir une politique :

- Quels furent les coûts des incidents informatiques passés ?
- Quel degré de confiance pouvez-vous avoir envers vos utilisateurs internes ?
- Qu'est-ce que les clients et les utilisateurs espèrent de la sécurité ?
- Quel sera l'impact sur la clientèle si la sécurité est insuffisante, ou tellement forte qu'elle devient contraignante ?
- Y a-t-il des informations importantes sur des ordinateurs en réseaux ? Sont-ils accessible de l'externe ?
- Quelle est la configuration du réseau et y a-t-il des services accessibles de l'extérieur ?
- Quelles sont les règles juridiques applicables à votre entreprise concernant la sécurité et la confidentialité des informations (ex: loi « informatique et liberté », archives comptables…) ?

Il ne faut pas également perdre de vue que la sécurité est comme une chaîne, guère plus solide que son maillon le plus faible. En plus de la formation et de la sensibilisation permanente des utilisateurs, la politique de sécurité peut être découpée en plusieurs parties :

- **Défaillance matérielle -** Tout équipement physique est sujet à défaillance (usure, vieillissement, défaut…) ; L'achat d'équipements de qualité et standard accompagnés d'une bonne garantie avec support technique est essentiel pour minimiser les délais de remise en fonction. Seule une forme de sauvegarde peut cependant protéger les données.

- **Défaillance logicielle -** Tout programme informatique contient des bugs. La seule façon de se protéger efficacement contre ceux-ci est de faire des copies de l'information à risque. Une mise à jour régulière des logiciels et la visite des sites consacrés à ce type de problèmes peuvent contribuer à en diminuer la fréquence.

- **Accidents (pannes, incendies, inondations…) -** Une sauvegarde est indispensable pour protéger efficacement les données contre ces problèmes. Cette procédure de sauvegarde peut combiner plusieurs moyens fonctionnant à des échelles de temps différentes : Disques RAID pour maintenir la disponibilité des serveurs ; Copie de sécurité via le réseau (quotidienne) ; Copie de sécurité dans un autre bâtiment (hebdomadaire).

- **Erreur humaine :** Outre les copies de sécurité, seule une formation adéquate du personnel peut limiter ce problème.

- **Vol via des dispositifs physique (disques et bandes) :** Contrôler l'accès à ces équipements : ne mettre des unités de disquette, bandes… que sur les ordinateurs où c'est essentiel. Mettre en place des dispositifs de surveillances.

- **Virus provenant de disquettes :** Ce risque peut-être réduit en limitant le nombre de lecteur de disquettes en service. L'installation de programmes antivirus peut s'avérer une protection efficace mais elle est coûteuse, diminue la productivité, et nécessite de fréquentes mises à jour.

- **Piratage et virus réseau :** Cette problématique est plus complexe et l'omniprésence des réseaux, notamment l'Internet, lui confère une importance particulière. Les problèmes de sécurité de cette catégorie sont particulièrement dommageables et font l'objet de l'étude qui suit.

## I.4. PRINCIPAUX DEFAUTS DE SECURITE INFORMATIQUE

Les défauts de sécurité peuvent être considérés comme des modifications accidentelles ou inconscientes du fonctionnement normal des équipements informatiques. Les défauts de sécurité d'un système d'information les plus souvent constatés sont :

- Installation des logiciels et matériels par défaut ;
- Mises à jour non effectuées ;
- Mots de passe inexistants ou par défaut ;
- Services inutiles conservés (Netbios…) ;
- Traces inexploitées ;
- Pas de séparation des flux opérationnels des flux d'administration des systèmes ;
- Procédures de sécurité obsolètes ;
- Eléments et outils de test laissés en place dans les configurations en production ;
- Authentification faible ;
- Télémaintenance sans contrôle fort.

# DEUXIEME CHAPITRE – FAILLES LA SECURITTE SUR INTERNET ET MODE DE PIRATERIE

En entreprise, c'est le réseau local qui est connecté à Internet. Il est donc indispensable de contrôler les communications entre le réseau interne et l'extérieur. De plus une formation du personnel est indispensable (règles de sécurité, déontologie, attention aux participations aux forums qui sont archivées ...). Les problèmes de sécurité qu'on peut rencontrer sur un réseau d'entreprise ou sur l'Internet relèvent d'abord de la responsabilité des victimes avant d'être imputables aux hackers. Les menaces qui ont sensiblement augmenté au cours de ces dernières années, nous indique la dernière étude du *Computer Security Institute*, un institut professionnel de San Francisco qui réalise chaque année un sondage auprès des entreprises en collaboration avec le FBI. Dans cette étude, plus de 40 % des sociétés interrogées ont déclaré que des intrus s'étaient introduits dans leurs systèmes depuis l'Internet, 38 % des sociétés ont détecté des attaques de type "déni de service", et 94 % ont été infectées par un virus en 2014.

D'autre part, votre sécurité peut dépendre d'autres entreprises dont vous pensez, parfois à tort, qu'elles ont assuré leur propre sécurité. Alors que le gouvernement et les forces de l'ordre cherchent à interpeller les intrus, les sociétés ne se préoccupent trop souvent que de relancer leurs réseaux après une attaque : « Le secteur privé ne cherche pas à savoir qui est responsable, tout ce qui intéresse les entreprises, c'est que l'attaque cesse ».

## II.1. DÉFINITION DES FAILLES SUR L'INTERNET

- ***IP spoofing -*** Usurpation d'adresse IP, on fait croire que la requête provient d'une machine autorisée. Une bonne configuration du routeur d'entrée permet d'éviter qu'une machine extérieure puisse se faire passer pour une machine interne.
- ***DNS spoofing -*** Pousse un serveur de DNS à accepter l'intrus. Solution : séparer le DNS du LAN de celui de l'espace public.
- ***Flooding -*** Raid massif de connexions non terminées.
- ***Smurf -*** Saturation de la bande passante.
- ***Web bug -*** Un mail publicitaire est envoyé en HTML (même si l'apparence est normale) avec une image transparente gif d'un pixel par un lien du type : **<img src="http://www.serveur.xx/image.gif ?email=utilisateur@adresse">.** Si le courrier est ouvert pendant la connexion, la requête de téléchargement de l'image vient confirmer la lecture du message et la validité de votre adresse.

- ***Hoax (rumeur) -*** Un « hoax » est une rumeur que l'on transmet par mail. Ces rumeurs colportent souvent des problèmes de sécurité soit disant découverts par des services officiels ou célèbres… Elles peuvent causer un véritable préjudice à certaines sociétés et de toute façon encombrent le réseau. Avant de retransmettre un tel message il est prudent de vérifier son authenticité.

- ***Hacker et cracker -*** Il existe une communauté, une culture partagée, de programmeurs expérimentés et de spécialistes des réseaux, dont l'histoire remonte aux premiers mini-ordinateurs multiutilisateurs, il y a quelques dizaines d'années, et aux premières expériences de l'ARPAnet. Les membres de cette culture ont créé le mot ***« hacker »***. Ces informaticiens sont généralement discrets, anti-autoritaristes et motivés par la curiosité. Il y a un autre groupe de personnes qui s'autoproclament des "hackers". Ces gens (principalement des adolescents de sexe masculin) prennent leur pied en s'introduisant à distance dans les systèmes informatiques et en piratant les systèmes téléphoniques, généralement à l'aide d'outils écrit par d'autres et trouvés sur Internet. Les vrais hackers appellent ces gens des « ***crackers »*** et ne veulent rien avoir à faire avec eux. Les vrais hackers pensent que les crackers sont des gens *paresseux, irresponsables et pas très brillants*.

- ***Déni de service (DoS) -*** Le but d'une telle attaque n'est pas de dérober des informations sur une machine distante, mais de paralyser un service ou un réseau complet. Les utilisateurs ne peuvent plus alors accéder aux ressources. Les deux exemples principaux, sont le « ping flood » ou l'envoi massif de courrier électronique pour saturer une boîte aux lettres *(mailbombing)*. La meilleure parade est le firewall ou la répartition des serveurs sur un réseau sécurisé.

- ***Écoute du réseau (sniffer) -*** Il existe des logiciels qui, à l'image des analyseurs de réseau, permettent d'intercepter certaines informations qui transitent sur un réseau local, en retranscrivant les trames dans un format plus lisible *(Network packet sniffing)*. C'est l'une des raisons qui font que la topologie en étoile autour d'un hub n'est pas la plus sécurisée, puisque les trames qui sont émises en « broadcast » sur le réseau local peuvent être interceptées. De plus, l'utilisateur n'a aucun moyen de savoir qu'un pirate a mis son réseau en écoute. L'utilisation de *switches* (commutateurs) réduit les possibilités d'écoute mais en inondant le commutateur, celui-ci peut se mettre en mode « HUB » par « sécurité ». La meilleure parade est l'utilisation de mot de passe non rejouable, de carte à puce ou de calculette à mot de passe.

- ***Social engineering :*** En utilisant les moyens usuels (téléphone, email…) et en usurpant une identité, un pirate cherche à obtenir des renseignements confidentiels auprès du personnel de l'entreprise en vue d'une intrusion future. Seule une formation du personnel permet de se protéger de cette attaque.

## II.2. PRINCIPALES ATTAQUES INFORMATIQUES

Dans le domaine informatique, il y a une pléthore d'attaques; certaines sont *connues des utilisateurs*, d'autres tenues *cachées par les experts*. Toutes ces attaques visent à modifier le comportement d'un SI. À côté de ces attaques, nous rencontrons diverses actions ou manipulations des logiciels malicieux visant à atteindre le noyau. L'objectif de ces attaques est de compromettre le système. Une fois que l'intrus s'introduit dans le système, il entreprend des actions profitant de vulnérabilités afin d'utiliser le système et dans la majorité des cas, de pérenniser son accès à l'insu des utilisateurs légitimes. Les vulnérabilités que nous avons sus-évoquées dans les points précédents sur le noyau Windows constituent une brèche favorisant les attaques. Or les attaques existent selon les objectifs et les finalités. Parmi les attaques les plus connues, on peut citer : *vers informatiques, virus, cheval de Troie, Rootkit,* etc.

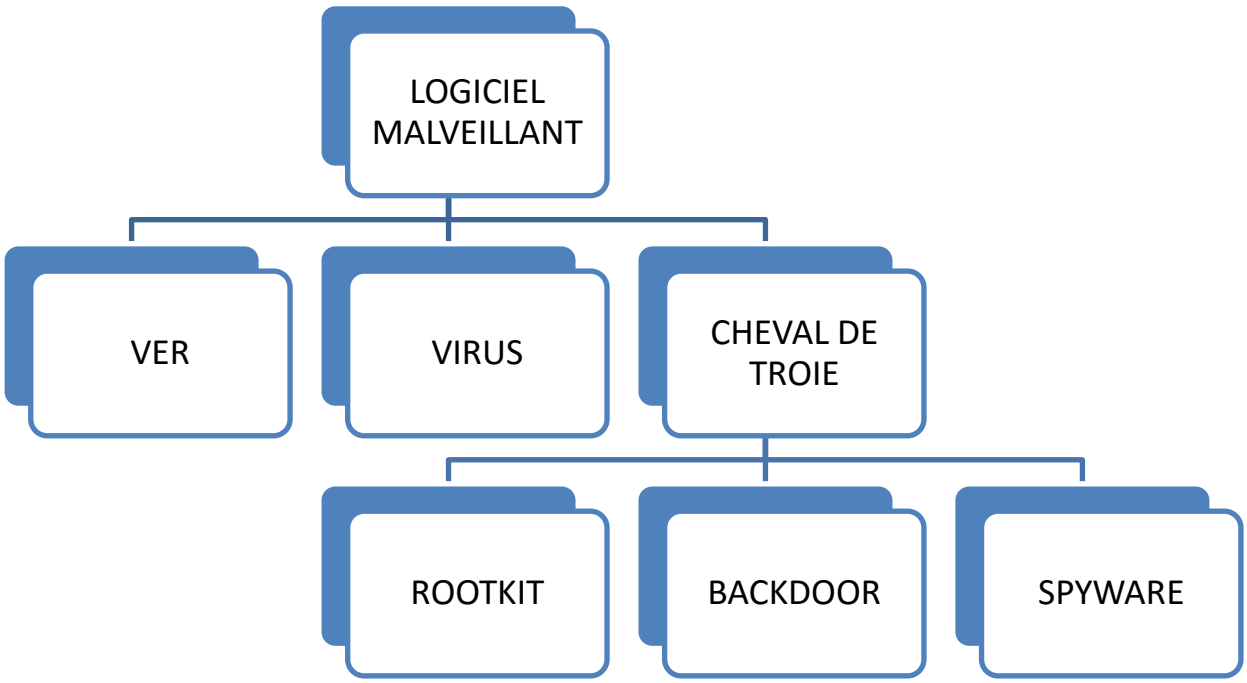
### II.2.1. LES LOGICIELS MALVEILLANTS

Un *logiciel malveillant ou maliciel,* aussi dénommé *logiciel nuisible* ou *programme malveillant* ou *pourriciel* (« *malware* » en anglais), est un programme développé dans le but de nuire à un système informatique, sans le consentement de l'utilisateur dont l'ordinateur est infecté. De nos jours, le terme « virus » est souvent employé, à tort, pour désigner toutes sortes de logiciels malveillants. En effet, les maliciels englobent les virus, les vers, les chevaux de Troie, ainsi que d'autres menaces. La catégorie des virus informatiques, qui a longtemps été la plus répandue, a cédé sa place aux chevaux de Troie en 2005. Les logiciels malveillants peuvent être classés en fonction des trois mécanismes suivants :

- le **mécanisme de propagation** (par exemple, un *ver* se propage sur un réseau informatique en exploitant une faille applicative ou humaine) ;
- le **mécanisme de déclenchement** (par exemple, la *bombe logique* comme la bombe logique surnommée *vendredi 13* se déclenche lorsqu'un évènement survient) ;
- la **charge utile** (par exemple, le *virus* Tchernobyl tente de supprimer des parties importantes du BIOS, ce qui bloque le démarrage de l'ordinateur infecté).

La classification n'est pas parfaite, et la différence entre les classes n'est pas toujours évidente. Cependant, c'est aujourd'hui la classification standard la plus couramment adoptée dans les milieux internationaux de la sécurité informatique.

Dans une publication, J. Rutkowska propose une taxonomie qui distingue les logiciels malveillants suivant leur mode de corruption du noyau du système d'exploitation : ne touche pas au noyau (*applications, micrologiciel*), corruption d'éléments fixes (*code*), corruption d'éléments dynamiques (*données*) et au-dessus du noyau *(hyperviseurs*).




Les programmes malveillants ont été développés pour de nombreux systèmes d'exploitation et applications. Pourtant, certains d'entre eux n'ont jamais été concernés. En effet, les auteurs de virus privilégient les systèmes d'exploitation largement utilisés ; les systèmes comportant des vulnérabilités ; et ceux pour lesquels une documentation détaillée est disponible (*puisqu'elle inclut des descriptions des services et des règles en vigueur pour écrire des programmes compatibles*). Le volume de logiciels malveillants destinés à Windows et Linux est à peu près proportionnel à leurs parts de marché respectives.

## II.2.2. LES VIRUS INFORMATIQUES

Un *virus informatique* est un automate auto réplicatif à la base non malveillant, mais aujourd'hui souvent additionné de code malveillant (donc classifié comme logiciel malveillant), conçu pour se propager à d'autres ordinateurs en s'insérant dans des logiciels légitimes, appelés « hôtes ». Il peut perturber plus ou moins gravement le fonctionnement de l'ordinateur infecté. Il peut se répandre par tout moyen d'échange de données numériques comme les réseaux informatiques et le CD-ROM, les clefs USB, les disques durs, etc.

Tout comme le virus biologique, le virus informatique poursuit 3 objectifs :

- se dissimuler le plus longtemps possible aux yeux de l'utilisateur infecté ;
- Il contamine tout ce qui est à sa portée ;
- Il tente de se répandre, sans se cantonner au support sur lequel il se trouve.

Sur Internet, les virus peuvent contaminer une machine de plusieurs manières :

- Téléchargement de logiciel puis exécution de celui-ci sans précautions ;
- Ouverture sans précautions de documents contenant des macros ;
- Pièce jointe de courrier électronique (exécutable, script type vbs…) ;
- Ouverture d'un courrier au format HTML contenant du javascript exploitant une faille de sécurité du logiciel de courrier;
- Exploitation d'un bug du logiciel de courrier (effectuer souvent les mises à jour).

L'appellation « **virus informatique** » provient d'une analogie avec le virus biologique puisqu'il présente des similitudes dans sa manière de se propager en utilisant les facultés de reproduction de la cellule hôte. On attribue le terme à l'informaticien et spécialiste en biologie moléculaire **Leonard Adleman**. Les virus informatiques ne doivent pas être confondus avec les vers informatiques, qui sont des programmes capables de se propager et de se dupliquer par leurs propres moyens sans contaminer de programme hôte. Au sens large, on utilise souvent et abusivement le mot *virus* pour désigner toute forme de logiciel malveillant.

Voici les quelques virus les plus célèbres du monde informatique :

- **Cabir** est considéré comme le tout premier virus informatique proof of concept recensé se propageant par la téléphonie mobile grâce à la technologie Bluetooth et du système d'exploitation Symbian OS.

- **MyDoom.A** est un virus informatique qui se propage par les courriels et le service P2P de Kazaa. Les premières infections ont eu lieu le 26 janvier 2004.
- **Psyb0t** est un virus informatique découvert en janvier 2009. Il est considéré comme étant le seul virus informatique ayant la capacité d'infecter les routeurs et modem haut-débit.
- **Le virus Tchernobyl ou CIH** est connu pour avoir été un des plus destructeurs. Il détruisait l'ensemble des informations du système attaqué et parfois il rendait la machine quasiment inutilisable. Il a sévi de 1998 à 2002.
- **Le ver Conficker** exploite une faille du Windows Server Service utilisé par Windows 2000, Windows XP, Windows Vista, Windows 7, Windows Server 2003 et Windows Server 2008.
- **Cryptolocker** est un logiciel malveillant dont la présence sur le web a augmenté de 700 % entre 2012 et 2014. Selon les calculs du FBI en juin 2014, il a causé pour 27 millions de dollars de pertes aux utilisateurs. Sous couvert d'une mise à jour Adobe Flash, le logiciel malveillant chiffre les fichiers des victimes et exige un paiement (pouvant aller de 100 dollars à 400 dollars) pour les décrypter[2].
- **Zeus Bot** est responsable d'environ 4 millions d'infections rien qu'aux États-Unis. Il a provoqué pour 70 millions de dollars de pertes pour les entreprises et consommateurs américains avant d'être démantelé par le FBI début 2014. Il exploite les vulnérabilités présentes dans Adobe Reader et Adobe Flash pour infecter les machines[3].

## II.2.2.1. BREF HISTORIQUE

Les premiers programmes autonomes n'avaient pas le but qu'ils ont aujourd'hui. Les tout premiers logiciels de ce type étaient de simples divertissements ; par exemple, un jeu a été inventé en 1970 par trois informaticiens **Victor A. Vyssotsky**, **Robert Morris Sr**. et **M. Douglas McIlroy** des Bell Labs et appelé « *Core War »*. Pour ce jeu, chaque joueur écrit un programme et le charge en mémoire vive. Le système d'exploitation, qui se doit juste d'être multitâche, exécute tour à tour une instruction de chacun des programmes. L'objectif du jeu est de détruire les programmes adverses tout en assurant sa propre prolifération. Les joueurs ne connaissent pas l'emplacement du programme adverse. Les programmes sont capables de se recopier, de se réparer, de se déplacer en différentes zones de la mémoire et « d'attaquer » les programmes adverses en écrivant de façon non déterministe dans d'autres zones de la mémoire. La partie se termine au bout d'un temps défini ou lorsque l'un des joueurs voit tous ses programmes inactifs ou détruits. Le vainqueur est celui qui possède le plus grand nombre de copies actives. C'est l'acte de naissance de la programmation des virus.

[2] Mark Allen Ludwig (traduction de Jean-Bernard Condat), *Naissance d'un virus : Technologie et principes fondamentaux*, Diff. Bordas, 1993, 47 p
[3] Peter Szor, *The Art of Computer Virus Research and Defense*, Addison-Wesley Professional, 2005, 744 p

En 1984, le magazine *Scientific American* a présenté un jeu informatique consistant à concevoir de petits programmes entrant en lutte et se dupliquant en essayant d'infliger des dégâts aux adversaires, fondant ainsi les bases des futurs virus. En 1986, l'ARPANET fut infecté par *Brain*, virus renommant toutes les disquettes de démarrage de système en (C) Brain. Les créateurs de ce virus y donnaient leurs noms, adresse et numéro de téléphone car c'était une publicité pour eux.

## II.2.2.2. CARACTERISTIQUES DES VIRUS

- **le chiffrement** - à chaque réplication, le virus est chiffré (afin de dissimuler les instructions qui, si elles s'y trouvaient en clair, révéleraient la présence de ce virus ou pourraient indiquer la présence de code suspect) ;
- **le polymorphisme** - le virus est chiffré et la routine de déchiffrement est capable de changer certaines de ses instructions au fil des réplications afin de rendre plus difficile la détection par l'antivirus ;
- **le métamorphisme** - contrairement au chiffrement simple et au polymorphisme, où le corps du virus ne change pas et est simplement chiffré, le métamorphisme permet au virus de modifier sa structure même et les instructions qui le composent ;
- **la furtivité** - le virus « *trompe* » le système d'exploitation (et par conséquent les logiciels antivirus) sur l'état des fichiers infectés. Des rootkits permettent de créer de tels virus. Par exemple, l'exploitation d'une faille de sécurité au niveau des répertoires permet de masquer l'existence de certains fichiers exécutables ainsi que les processus qui leur sont associés.

## II.2.2.3. CLASSIFICATION DES VIRUS

Il n'existe pas de classification stricte des virus. Cependant on peut en retenir 4 grandes classifications :

- **Classification selon le format visé** (exécutable ou documents) ;
- **Classification selon leur comportement** (rapide, lent, résident, polymorphe…) ;
- **Classification selon l'organe visé** (boot sector, driver…)
- **Classification selon le langage utilisé** (virus assembleur, macrovirus, virus interprété…).

C'est grâce à cette classification que nous pouvons maintenant en mesure de distinguer les différents types de virus qui existent :

- **Le virus classique -** est un morceau de programme, souvent écrit en assembleur, qui s'intègre dans un programme normal (le Cheval de Troie), le plus souvent à la fin, mais cela peut varier. Chaque fois que l'utilisateur exécute ce programme « infecté », il active le virus qui en profite pour aller s'intégrer dans d'autres programmes exécutables. De plus, lorsqu'il contient une charge utile, il peut, après un certain temps (qui peut être très long) ou un événement particulier, exécuter une action prédéterminée. Cette action peut aller d'un simple message anodin à la détérioration de certaines fonctions du système d'exploitation ou la détérioration de certains fichiers ou même la destruction complète de toutes les données de l'ordinateur. On parle dans ce cas de « bombe logique » et de « charge utile ».

- **Un virus de boot** - s'installe dans un des secteurs de boot d'un périphérique de démarrage, disque dur (le secteur de boot principal, le « *Master boot record* », ou celui d'une partition), disquette, ou autre. Il remplace un chargeur d'amorçage (ou programme de démarrage ou « *bootloader* ») existant (en copiant l'original ailleurs) ou en crée un (sur un disque où il n'y en avait pas) mais ne modifie pas un programme comme un virus normal ; quand il remplace un programme de démarrage existant, il agit un peu comme un virus « *prepender* » (qui s'insère au début), mais le fait d'infecter aussi un périphérique vierge de tout logiciel de démarrage le distingue du virus classique, qui ne s'attaque jamais à « rien ».

- **Les macro virus** - qui s'attaquent aux macros de logiciels de la suite Microsoft Office (Word, Excel, etc.) grâce au VBA de Microsoft. Par exemple, en s'intégrant dans le modèle normal.dot de Word, un virus peut être activé à chaque fois que l'utilisateur lance ce programme.

- **Les virus-vers** - apparus aux environs de l'année 2003, ayant connu un développement fulgurant dans les années qui suivirent, sont des virus classiques car ils ont un programme hôte. Mais s'apparentent aux vers (en anglais « *worm* ») car : Leur mode de propagation est lié au réseau, comme des vers, en général via l'exploitation de failles de sécurité. Comme des vers, leur action se veut discrète, et non destructrice pour les utilisateurs de la machine infectée. Comme des vers, ils poursuivent des buts à visée large, tels que l'attaque par saturation des ressources ou attaque DoS (*Denial of Service*) d'un serveur par des milliers de machines infectées se connectant simultanément.

- **Les virus de type batch** - apparu à l'époque où MS-DOS était le système d'exploitation en vogue, sont des virus « primitifs ». Bien que capables de se reproduire et d'infecter d'autres fichiers batch, ils sont lents et ont un pouvoir infectant très faible. Certains programmeurs ont été jusqu'à créer des virus batch cryptés et polymorphes, ce qui peut être qualifié de « prouesse technique » tant le langage batch est simple et primitif.

- **Les virus résidents** - Ils se placent en mémoire RAM et contaminent les fichiers au fur et à mesure de leurs exécutions. Ils peuvent par exemple prendre la forme de fichier pilote de Windows (.vxd). Ils sont alors chargés dès le démarrage du système, avant le chargement de l'antivirus.

- **Les virus lents** - A la différence d'un virus rapide qui infecte les fichiers dès qu'ils sont manipulés par le système, les virus lents n'infectent les fichiers qu'en cas de modification, ce qui rend leur détection plus subtile.

- **Les virus furtifs** - Un virus furtif est un virus qui, lorsqu'il est actif, dissimule les modifications apportées aux fichiers ou aux secteurs de boot. En règle générale, ce phénomène est rendu possible par le virus qui observe les appels aux fonctions de lecture des fichiers et falsifie les résultats renvoyés par ces fonctions. Cette méthode permet au virus de ne pas être détecté par les utilitaires anti-virus qui recherchent des modifications éventuelles apportées aux fichiers. Néanmoins, pour que cela soit possible, le virus doit être actif en mémoire résidente, ce qui est détectable par les anti-virus.

- **Les virus polymorphes** - Un virus polymorphe est un virus qui produit des copies variées de lui-même, mais qui restent opérationnelles. Ces stratégies ont été employées dans l'espoir que les utilitaires anti-virus ne puissent pas détecter toutes les variantes du virus.

- **Les virus « cavité »** - Les virus cavités sont des virus qui écrasent une partie du fichier hôte qui est constitué d'une constante (*en général, des 0*) sans augmenter la taille du fichier et tout en préservant sa fonctionnalité.

- **Les virus compagnons** - Un virus compagnon est un virus qui, au lieu de modifier un fichier existant, crée un nouveau programme qui est exécuté à l'insu de l'utilisateur au lieu du programme voulu. Le programme original est ensuite exécuté de telle sorte que tout apparaît normal à l'utilisateur. Sur un PC, ceci est généralement accompli en créant un nouveau fichier .COM portant le même nom que le fichier .EXE. Les anti-virus qui ne cherchent que les modifications apportées aux fichiers existants (*vérificateurs d'intégrité*) ne détecteront pas ce type de virus.

- **Les virus blindés** - Un virus blindé est un virus qui utilise des astuces spéciales pour que son dépistage, son désassemblage et la compréhension de son code soient plus durs. Ils utilisent certaines ruses techniques pour mieux résister au désassemblage et à la détection et rendre leur fonctionnement quasiment incompréhensible.

- **Les virus souterrains** - Les virus souterrains sont des virus qui appellent directement les vecteurs d'interruption du DOS et du BIOS, contournant ainsi tout

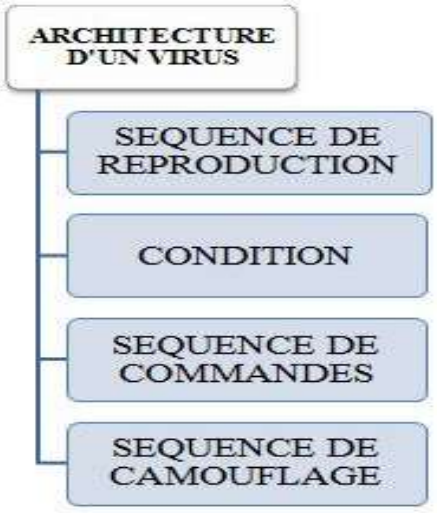
programme de contrôle qui pourrait être chargé et avoir intercepté ces mêmes vecteurs dans le but de détecter l'activité d'un virus. Certains anti-virus utilisent cette même technique pour contourner un virus inconnu ou non détecté.

- **Les virus compte-gouttes** - Un virus compte-gouttes est un programme conçu pour installer un virus sur le système visé. Le code du virus est en règle générale contenu dans ce programme de telle manière qu'il ne sera pas détecté par un anti-virus qui, dans d'autres circonstances, détecte ce virus (le compte-gouttes n'est pas infecté par ce virus). Bien qu'assez rare, ce type de virus a été signalé à plusieurs reprises. Un compte-gouttes est en fait un cheval de Troie dont le but est d'installer le virus. Un compte-gouttes qui installe le virus seulement en mémoire (donc sans infecter de fichiers sur le disque) est parfois appelé un *injecteur*.

- **Les bombes ANSI** - Une bombe ANSI est une séquence de caractères, généralement incluse dans un fichier texte, qui reprogramme certaines fonctions du clavier d'ordinateurs ayant une console ANSI (écran + clavier). On peut ainsi reprogrammer la touche Enter d'un clavier pour qu'elle exécute l'instruction format c : suivi de la fonction Enter. Néanmoins, cette possibilité ne constitue pas une grande menace. En effet, il est rare pour un logiciel moderne d'exiger un ordinateur tournant sur une console ANSI. De même, peu de gens utilisent des logiciels qui envoient simplement la sortie sur le terminal, donc une bombe ANSI dans un émail ne reprogrammerait pas votre clavier.

## II.2.2.4. ARCHITECTURE D'UN VIRUS

Un virus se compose de 3 fonctionnalités principales et d'une quatrième optionnelle (mais de plus en plus présente dans les virus afin d'en améliorer l'efficacité), comme le montre la *Figure ci-dessous :*



- **Séquence de reproduction** - C'est l'objectif premier du virus. Elle inclut une fonctionnalité de recherche, qui permet de rechercher des fichiers à infecter. Elle permet aussi au virus de vérifier d'abord que le fichier n'est pas déjà infecté, pour ne l'infecter que le cas échéant. En effet, un virus ne doit pas se reproduire deux fois dans un fichier, car son comportement serait alors faussé.

- **Condition** - Il s'agit tout simplement de la partie qui va cordonner le lancement de l'action qu'est censé accomplir le virus. En effet, le virus a toujours un objectif précis. C'est la séquence de commande (*ou de destruction*) qui est chargée de cette action. Elle est déclenchée lorsque la condition est satisfaite. Cette dernière peut-être de diverses forme (*une date, une action particulière de l'utilisateur, une réaction spécifique de l'ordinateur...).* Les développeurs de virus font preuve de toujours plus d'imagination pour trouver des conditions de déclenchement de plus en plus originales et spécifiques. Cette condition peut aussi être à l'origine du bon fonctionnement ou non du virus.

- **Séquence de commandes** - Comme nous venons de le dire, c'est elle qui effectue l'action du virus. Cela peut être détruire des fichiers, formater une partition...

- **Séquence de camouflage** - Malgré leur petite taille, les virus peut être vite repérés (pour certains). Les développeurs de virus ont donc élaboré plusieurs techniques pour cacher le virus. Il existe plusieurs techniques. Nous les aborderons en parlant des virus polymorphes et furtifs par la suite.

Il s'avère qu'en dépit de leur finalité, tous les virus fonctionnent selon un modèle unifié qui les décrit tous: le modèle de *virus générique*. *Eric Filiol* et *Jean-Paul* Fizaine estiment que le virus est une sorte de machine de Turing se décomposant en sous autres machines de Turing connectées entre elles dont chacune représente un élément fonctionnel. Un virus **V** est un quadruplet :

$$\mathbf{V_{Pr} = (Rech, Inf, Cf, Tr)}.$$

**Rech** c'est l'ensemble des fonctions de recherche de cibles; **Inf** est l'ensemble des fonctions d'infection tel que :

$$\mathbf{Inf=\{E, R, A, T\}}$$

Où **E, R, A, T** sont des ensembles de machines de Turing tels que **E** la classe des fonctions par *écrasement*, **R** la classe des fonctions par *recouvrement*, **A** la classe des fonctions par *accompagnement*, **T** est la classe des fonctions par *entrelacement.* Tandis que **Cf** est l'ensemble des fonctions des *charges finales* et **Tr** l'ensemble des fonctions de *transfert d'exécution*.

## II.2.2.5. MODE DE CONTAMINATION D'UN VIRUS

Le virus informatique utilise de même 4 modes de contamination pour se propager dans les systèmes informatiques :

- **Contamination par recouvrement** - le virus écrase les premières instructions du fichier par ses propres instructions. L'avantage est que la taille du fichier n'est pas modifiée. Cependant, il n'est plus utilisable par la suite, le début de ses instructions ayant été supprimé.
- **Contamination par ajout** - le virus s'exécute avant le code original du fichier infecté, mais repasse la main à ce dernier à la suite de son exécution. Dans ce cas, la taille du fichier est modifiée.
- **Contamination par entrelacement** - il s'agit ici d'insérer du code entre les blocs valides du programme. Elle est plus difficile à mettre en place, mais est moins facilement détectée.
- **Contamination par accompagnement** - ici, il s'agit tout simplement de surcharger les fichiers infectés en les rendant plus lourd lors de leurs exécutions.

## II.2.2.5. CYCLE DE VIE D'UN VIRUS

Les virus informatiques suivent un cycle de vie, qui recense 7 grandes étapes :

- **Création** : c'est la période durant laquelle un programmeur développe un virus aussi féroce que possible (dans la majeure partie des cas). La programmation se fait généralement en code assembleur ou Visual Basic, ou encore parfois en C ou C++.
- **Gestation** : C'est le temps pendant lequel le virus s'introduit dans le système qu'il souhaiter infecter. Il y reste en sommeil.
- **Reproduction (infection):** comme nous l'avons dit, le virus doit se reproduire .Un virus correctement conçu se reproduira un nombre important de fois avant de s'active r. C'est là le meilleur moyen de s'assurer de la pérennité d'un virus.
- **Activation** : Les virus possédant une routine de destruction (portions de code destinées à causer des dégâts sur l'hôte) ne s'activent que lorsque certaines conditions sont réunies. Certains s'activent à une date précise (fixée par le développeur), d'autres possèdent un système de compte à rebours interne. L'activation peut aussi avoir lieu à distance, par le développeur. Même les virus ne possédant pas de telles routines et ne nécessitant pas de procédure

d'activation spécifique peuvent causer des dommages aux systèmes en s'appropriant petit à petit l'ensemble des ressources.

- **Découverte** : C'est le moment où l'utilisateur s'aperçoit que son système a des comportements étranges et soupçonne la présence de virus. Ou alors, les anti-virus performants découvrent certains virus avant qu'ils aient eu le temps de faire des ravages.
- **Assimilation** : Une fois la découverte faite, les développeurs de logiciels anti-virus mettent à jour leur base de donnée vira le (nous reviendrons sur cette notion) afin que les utilisateurs puissent détecter la présence de virus sur leur ordinateur. Ils développent également le correctif (ou antidote) permettant d'éradiquer le virus (si cela est possible).
- **Elimination** : C'est la mort du virus. Tout au moins, c'est la mort de l'exemplaire du virus sur un poste utilisateur. C'est le moment où l'anti-virus ayant découvert le virus propose à l'utilisateur de le supprimer. Même si de nombreux virus connus depuis des années ne sont pas complètement annihilés, ils ont cessé de constituer une menace sérieuse car ils sont découverts très rapidement. Dans les faits, rares sont les virus ayant complètement disparu.

## II.2.3. LES VERS INFORMATIQUES

Un *ver informatique* est un logiciel malveillant qui se reproduit sur plusieurs ordinateurs en utilisant un réseau informatique comme Internet. Il a la capacité de se dupliquer une fois qu'il a été exécuté. Contrairement au virus, le ver se propage sans avoir besoin de se lier à d'autres programmes exécutables. Le ver appartient à la famille des programmes malveillants ou nuisibles, « *les malware* ».

Le concept de ver vient des programmes autoreproducteurs qui eux-mêmes sont issus d'idées des logiciens *John von Neumann* et *W. V. Quine*. Un virus, contrairement à un virus informatique, n'a pas besoin d'un programme hôte pour se reproduire. Il exploite les différentes ressources de l'ordinateur qui l'héberge pour assurer sa reproduction. Il faut noter également que les données qui sont corrompues ou détruites par un ver informatique sont généralement irrécupérables.

La plupart du temps, les vers n'ont d'autres utilités que la destruction (s'ils s'accompagnent d'une bombe logique), et la congestion du réseau. Malgré tout, le ver a parfois d'autres utilisations. Certaines entreprises les utilisent pour tester la sécurité de leur réseau intranet. L'objectif des vers n'est pas seulement de se reproduire mais habituellement, ils sont un objectif malfaisant, par exemple :

- Espionner l'ordinateur où il se trouve ;
- Offrir une porte dérobée à des pirates informatiques ;
- Détruire des données sur l'ordinateur où il se trouve ou y faire d'autres dégâts ;
- Envoyer de multiples requêtes vers un site Internet dans le but de le saturer (attaque par déni de service).

Les conséquences des effets secondaires de l'activité d'un ver sur un ordinateur sont souvent :

- Le ralentissement par saturation de la machine infectée ;
- Le ralentissement par saturation du réseau utilisé par la machine infectée ;
- Le plantage de services ou du système d'exploitation de la machine infectée.

En majorité, des vers écrits sous forme de scripts peuvent être intégrés dans un *courriel* ou sur une page *HTML* (*chevaux de Troie*). Ces vers sont activés par les actions de l'utilisateur qui croit accéder à des informations lui étant destinées. Un ver peut aussi être programmé en C, C++, Delphi, assembleur, ou dans un autre langage de programmation. La plupart du temps, les vers utilisent des failles de logiciels pour se propager.

Voici les quelques exemples des vers sur Internet les plus célèbres :

- Félicitations vous venez d'être tirer au sort pour un séjour aux Etats-Unis.
- Vous avez reçu un bonus sur votre carte Visa.
- Vous êtes les 1.000.000 visiteurs ; Vous venez de gagner un smartphone.
- Le ver informatique le plus populaire est le ver crée par Samy Kankar qui lui a donné son nom : le **ver SAMY**. Il a infecté plus d'un million d'utilisateur *MySpace* en seulement 20 heures. Voici ce qu'affichait le ver à l'écran des utilisateurs infectés : "*mais par dessus tout, Samy est mon héros*". Chaque personne visitant un profil infecté se faisait infecter à son tour.
- **Le ver Facebook** qui se propage à travers Facebook Messenger, qui est un système de messagerie instantanée incorporé au réseau social Facebook. Une personne infectée va automatiquement envoyer un message à tous ses contacts avec un lien vers un site permettant de télécharger le ver.

## II.2.3.1. BREF HISTORIQUE

Le terme « ver » (en anglais « *worm* ») a été utilisé pour la première fois par le romancier britannique John Brunner dans son roman *Sur l'onde de choc*. Le 2 novembre 1988, Robert Tappan Morris, étudiant en informatique, mit en circulation ce qui a été appelé plus tard le ver de Morris et qui causa le krach d'un très grand nombre d'ordinateurs sur Internet. Durant le procès de Morris, la cour a estimé que le coût de l'élimination du virus peut être évalué entre 200 et 53 000 dollars. Quant à Morris, il est la première personne condamnée en vertu de la loi américaine sur les fraudes et les abus (*Computer Fraud and Abuse Act*)[4].

## II.2.3.2. CARACTERISTIQUE D'UN VER INFORMATIQUE

- Le ver est souvent transmis aux ordinateurs de différentes manières, telles que le courrier électronique, des programmes source obscurs, des sites de forum, des DVD et des CD de jeux piratés.
- Le ver est conçu pour se copier d'un ordinateur à un autre automatiquement. Tout d'abord, il prend le contrôle des propriétés qui transmettent des fichiers ou des informations sur l'ordinateur. Cela peut entraîner un trafic réseau important en raison de l'effet domino, ce qui ralentit les réseaux des lieux de travail et l'ensemble de l'Internet.
- Le ver est une sous-classe de virus et se propage généralement sans action de l'utilisateur et distribue des copies complètes de lui-même (*éventuellement modifiées)* de réseaux en réseaux.
- Un ver peut consommer de la mémoire ou de la bande passante réseau, ce qui peut entraîner une panne d'ordinateur.
- Comme les vers n'ont pas besoin d'un programme ou d'un fichier "support" pour se répandre, ils peuvent ouvrir un tunnel dans votre système et permettre à une autre personne de contrôler votre ordinateur à distance. Des exemples de vers récents incluent le **ver Sasser** et le **ver Blaster**.

## II.2.3.3. CLASSIFICATION D'UN INFORMATIQUE

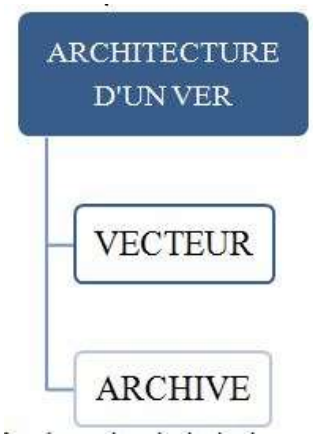
Il existe actuellement 4 Classes des vers informatiques :

- Vers de réseau (*Vers de réseaux de partage de fichiers*) ;
- Vers de courrier électronique ;
- Vers de messagerie instantanée (*Vers IRC : Internet Relay Chat*);
- Vers Internet ;

[4] J. Dressler, *Cases and Materials on Criminal Law*, Thomson/West, 2007 « United States v. Morris »

## V.2.3.4. ARCHITECTURE D'UN VER INFORMATIQUE

Il est composé de deux parties :




- **Le vecteur** - c'est la partie principale du ver, qui s'occupe de la recherche de failles et est responsable de transmettre la seconde partie du ver.

- **L'archive** - c'est la partie « *morte* » du ver, qui sera envoyée sur le système distant et l'infectera.

Cette archive peut exister sous deux formes distinctes :

- **Sous forme de sources** - la reproduction est beaucoup plus aisée, car le nombre de machines potentiellement ciblées est beaucoup plus grand. L'inconvénient majeur de ce procédé est qu'un compilateur est nécessaire sur la machine à infecter. Les données nécessaires à l'infection sont plus volumineuses et une compression des sources sera parfois opérée préalablement.

- **Sous forme binaire** - la reproduction de ce type de ver n'est possible que sur une architecture compatible avec les données binaires. Cependant, ce format ne nécessite pas de compilateur sur la machine hôte et est souvent plus compacte que la version source. Il est à remarquer qu'il existe des vers multiformes, c'est-à-dire utilisant une archive en sources ou binaire selon le système à infecter.

## II.2.3.5. MODE DE REPRODUCTION D'UN VER

Avant d'infecter un système, le ver doit procéder dans l'ordre à une série d'étapes. Ce n'est qu'à la suite de celles-ci qu'il pourra attaquer la machine cible. La succession d'étapes est la suivante (et ne concerne que la partie « *vecteur* » jusqu'à la phase d'invasion) :

- **Initialisation** - ce sont les premières instructions du ver. Il peut s'agir de la création de fichiers temporaires, ou la compilation d'une partie de l'archive. A cet instant, le ver est toujours sur la machine mère.
- **Recherche de l'hôte** - Cette étape procède à un scan d'IP. Ce scan peut être aléatoire ou incrémental (*on passe en revue des séries d'adresses IP*). Pour chacune d'entre elles, on teste si le système répond ou non ("up" ou "down").
- **Identification de l'hôte** - une fois une victime potentielle détectée, le ver (la partie vecteur) va tester le système en place sur la machine distante. Il s'agira de vérifier si le système possède des failles pouvant être exploitées par le ver, s'il possède un compilateur approprié, ... Si ce n'est pas le cas, on retourne à l'étape précédente.
- **Attaque** - c'est ici que le ver exploite les vulnérabilités mises au jour dans la phase précédente. Le vecteur obtient alors un accès sur la machine cible.
- **Invasion** - Le vecteur est maintenant présent sur la machine cible, mais n'est pas encore actif. Les instructions sont toujours données par la machine mère. Dès cet instant, le vecteur va rapatrier la partie archive sur le système à infecter. Deux techniques sont possibles :

  → *Soit le vecteur est toujours accompagné de sa partie archive*, auquel cas l'archive est présente sur la machine à tout moment, le vecteur uploadant son archive de machines en machines. Cette technique est un peu plus complexe à réaliser car le vecteur doit avoir été programmé pour transmettre cette archive.

  → *Soit le vecteur doit rapatrier la partie archive.* Ce rapatriement se fait à partir de la machine mère, ou depuis un serveur fixe (de type FTP) et unique pour toutes les machines (mais dans ce cas, l'attaque se terminera si le serveur tombe en panne). C'est beaucoup plus facile à réaliser, mais également plus dangereux car l'utilisateur piraté pourra plus facilement tracer le ver. Si l'archive est sous forme de sources, le vecteur devra également procéder à la décompression et à la compilation de ces sources avant de passer à la phase suivante.

- **Reproduction** - ici aussi, deux possibilités existent pour permettre la reproduction du ver :

  → *Soit on tue le ver présent sur la machine mère*. Le ver se déplacera alors de stations en stations.

  → *Soit on ne le tue pas et chacun des deux vers continue à se propager selon la méthode décrite ci-dessus*. Ce choix provoque une étendue exponentielle du ver et est logiquement plus dangereuse que la précédente.

Il faut également noter que des mesures simples permettent de limiter le risque d'attaque par un ver :

- Analyse régulière de tous les fichiers suspects à l'aide d'un antivirus ;
- Mises à jour régulières des logiciels installés pour s'assurer d'avoir les dernières versions ;
- Évitement des sites Internet à risque (*sites de hackers ou de téléchargement de matériel piraté par exemple)* ;
- Scannage des pièces jointes d'un email par un antivirus à jour.

## II.2.4. LE CHEVAL DE TROIE

Un *cheval de Troie* (*Trojan horse* en anglais) est un type de logiciel malveillant, qui ne doit pas être confondu avec les virus ou autres parasites. Le cheval de Troie est un logiciel en apparence légitime, mais qui contient une fonctionnalité malveillante. Le rôle du cheval de Troie est de faire entrer ce parasite sur l'ordinateur et de l'installer à l'insu de l'utilisateur.

Un cheval de Troie informatique est un programme d'apparence inoffensive, mais qui en contient un autre, malveillant celui-là et qui est installé par l'utilisateur lui-même, ignorant qu'il fait pénétrer un intrus malveillant sur son ordinateur. C'est par analogie, que ce type de programme a été baptisé « *cheval de Troie* », en référence à la ruse qu'*Ulysse* utilisa pour contourner les défenses adverses. Le cheval de Troie prend l'apparence d'un logiciel existant, légitime et parfois même réputé, mais qui aura été modifié pour y dissimuler un parasite. La subtilité avec laquelle l'installation est faite est expliquée par *Ken Thompson* dans sa conférence Turing. Berné, l'utilisateur va télécharger et installer le programme, pensant avoir affaire à une version saine.

En réalité, le logiciel véhicule un parasite qui va pouvoir s'exécuter sur son ordinateur. Les logiciels crackés peuvent être des chevaux de Troie qui vont allécher l'internaute qui cherche à obtenir gratuitement un logiciel normalement payant *(Adobe Acrobat pro, Photoshop, Microsoft Office..).* En 2014, une étude de l'Association of Internet Security Professionnals centrée sur les dangers du live streaming illégal révèle qu'un ordinateur sur trois est infecté par un logiciel malveillant et que 73 % de ces infections proviennent d'un cheval de Troie[5].

Le cheval de Troie ne doit pas être confondu avec d'autres notions proches :

- L'**injecteur** (ou *dropper*, en anglais) - est quasiment identique au cheval, car il sert lui aussi de véhicule pour une malveillance. Mais l'injecteur est un programme spécialement fabriqué pour propager des parasites, alors que le cheval est une version modifiée d'un programme existant et légitime.
- La **porte dérobée** (*backdoor*) - est un programme qui va s'exécuter discrètement sur l'ordinateur où il est installé pour y créer une faille de sécurité. Le backdoor ouvre un ou plusieurs ports sur la machine, ce qui lui permet d'accéder à internet librement et de télécharger, à l'insu de l'utilisateur, un parasite. Le backdoor n'est donc pas un cheval de Troie : il ne véhicule pas le parasite en lui, il va simplement ouvrir l'accès et récupérer, via internet, le programme malveillant qui se trouve sur un serveur distant.
- Le **RAT (Remote administration tool) -** est un logiciel de prise de contrôle à distance d'un ordinateur. Un RAT peut être un outil légitime (par exemple pour le dépannage à distance), mais il peut aussi être utilisé par un pirate pour s'emparer d'une machine. Dans ce cas, l'introduction du RAT sur la machine à contrôler se fait à l'insu de l'utilisateur. Par exemple, par un cheval de Troie qui contient le RAT, mais le RAT n'est pas le cheval. Contrairement à ce qu'on lit parfois, le T de RAT ne signifie pas *Trojan* mais *Tool* (outil).
- Les **bombes de décompression -** ne transportent pas de parasite, mais elles peuvent être confondues avec les chevaux de Troie car la notion de conteneur entre aussi en jeu. Il s'agit d'un fichier compressé, par exemple un fichier zip, de taille raisonnable tant qu'il n'est pas ouvert. Mais lorsque l'utilisateur va tenter de la décompresser, elle va générer un fichier d'une taille gigantesque. Cette « explosion » entraîne le ralentissement ou le plantage de l'ordinateur, et sature le disque dur avec des données inutiles. Bien qu'il s'agisse de conteneurs malveillants, le fonctionnement des bombes de décompression n'a donc rien à voir avec celui des chevaux de Troie. En effet, elles ne transportent aucun parasite indépendant, elles saturent la machine de données aléatoires.

[5] Karger et Schell écrivent même que Thompson a ajouté cette référence dans une version ultérieure de sa conférence Turing : Ken Thompson, « On Trusting Trust. », *Unix Review*, vol. 7, n° 11, novembre 1989, p. 70-74

- **Le Trojan Stealer** - plutôt spécialisé dans le vol de données et notamment les comptes en ligne (*Mail, Réseaux sociaux ou encore bancaire*).

## II.2.4.1. BREF HISTORIQUE

Les chevaux de Troie informatiques (ou *Trojan horses* en anglais) tirent leur nom d'une célèbre légende de la Grèce antique, racontée par Homère dans l'Iliade et reprise par Virgile dans l'Énéide. Le cheval de Troie est la méthode utilisée par les Grecs pour conquérir la ville de Troie : *le héros Ulysse* fit construire un immense étalon de bois qu'il plaça devant les portes de Troie et dans les flancs duquel il se cacha avec ses compagnons. Lorsque les Troyens découvrirent ce cheval, ils le firent entrer eux-mêmes dans leur cité. Ils s'endormirent sans méfiance tandis que le cheval se trouvait dans leurs murs. À la nuit tombée, Ulysse et ses compagnons sortirent de leur cachette et ouvrirent les portes de la ville au reste de l'armée, qui la détruisit et massacra ses habitants.

Le terme « *cheval de Troie* » a été inventé en 1970 par Daniel J. Edwards, chercheur à la NSA. La terminologie a été par la suite utilisée en 1974 dans un rapport de l'US Air Force sur l'analyse de la vulnérabilité des systèmes informatiques, puis présentée en 1981 par David Jordan et enfin vraiment popularisée par Ken Thompson dans la conférence Turing qu'il donna à la réception du prix Turing en 1983, prix qu'il avait reçu pour avoir créé UNIX. Symptôme

## II.2.4.2. MANIFESTIONS D'UNE INFECTION PAR UN CHEVAL DE TROIE

- Activité anormale de la carte réseau ou du disque dur (des données sont chargées en l'absence d'activité de la part de l'utilisateur) ou du modem ;
- Réactions curieuses de la souris ;
- Ouvertures impromptues de programmes, du lecteur CD/DVD ;
- Plantages répétés ;
- Redémarrage répété du système ;
- Écran ou fenêtres avec des messages inhabituels ;
- Un comportement inhabituel dans le fonctionnement de l'ordinateur, tels que: changements d'économiseur d'écran de bureau, modification du rôle des boutons de la souris, modification du volume du lecteur audio ;

- Ouverture/Fermeture intempestive de fenêtres ;
- Les programmes commencent ou terminent leur exécution de manière inattendue ;
- Le navigateur accède tout seul à certains sites Internet ;
- Présence d'autres programmes qui n'ont pas été volontairement installés (y compris des logiciels malveillants) ;
- Vol de renseignements personnels : informations bancaires, mots de passe, codes de sécurité...
- Suppression, modification ou transfert de fichiers (*téléchargement ou upload*) ;
- Exécution ou arrêt de processus ;
- Arrêt ou redémarrage impromptus de l'ordinateur ;
- Surveillance des frappes (voir Enregistreur de frappe) ;
- Captures d'écran impromptues ;
- Espace libre du disque dur occupé par des fichiers inutiles.

## II.2.5. LES ROOTKITS

Un *rootkit* (Aussi appelé « *outil de dissimulation d'activité* », « *maliciel furtif* », « *trousse administrateur pirate* », parfois simplement « *kit* »), est un ensemble de techniques mises en œuvre par un ou plusieurs logiciels, dont le but est d'obtenir et de pérenniser un accès (généralement non autorisé) à un ordinateur de la manière la plus furtive possible, à la différence d'autres logiciels malveillants. Le terme peut désigner la technique de dissimulation ou plus généralement un ensemble particulier d'objets informatiques mettant en œuvre cette technique.

Pour l'« *attaquant* », l'utilité d'un « *rootkit »* est soit de mettre à disposition des ressources système (temps processeur, connexions réseaux, etc.) sur une, voire plusieurs machines, parfois en utilisant la « cible » comme intermédiaire pour une autre attaque ; soit d'espionner, d'accéder aux données stockées ou en transit sur la machine cible[6]. Ils sont généralement classés parmi les logiciels malveillants, mais pas toujours ; ils peuvent utiliser des « *techniques virales* » pour se transmettre (par exemple, en utilisant un virus ou un cheval de Troie). Il existe des outils de détection et des méthodes de protection pour les contrer mais elles ne sont pas totalement efficaces.

[6] Greg Hoglund et James Butler (trad. Freenet Sofor ltd.), *Rootkits : infiltrations du noyau Windows* [« Rootkits: Subverting the Windows »], Paris, Campus Press, 2006 (1re éd. 2005), 338 p.

## II.2.5.1. TYPES DES ROOTKITS

Un rootkit peut intervenir à un ou plusieurs niveaux du système parmi les cinq suivants : *micrologiciel, hyperviseur, noyau, bibliothèque* ou *applicatif* :

- **Niveau micrologiciel / matériel** - Il est possible d'installer des rootkits directement au niveau du micrologiciel (ou *firmware*). De nombreux produits proposent désormais des mémoires flash qui peuvent être utilisées pour injecter durablement du code en détournant par exemple l'usage d'un module de persistance souvent implanté dans le BIOS de certains systèmes. C'est par exemple, « *LoJack, d'Absolute Software* »[7] Un outil légitime qui utilise cette technique pouvant permet de suivre un ordinateur à l'insu de l'utilisateur pour retrouver un ordinateur portable en cas de vol. Ce code reste en place après un formatage du disque dur, voire après un flashage du BIOS si le module de persistance est présent et actif. Tout périphérique disposant d'un tel type de mémoire est donc potentiellement vulnérable.

- **Niveau hyperviseur** - Ce type de rootkit se comporte comme un hyperviseur natif, après s'être installé et avoir modifié la séquence de démarrage, pour être lancé en tant qu'hyperviseur à l'initialisation de la machine infectée. Le système d'exploitation original devient alors un hôte (invité) du rootkit, lequel peut intercepter tout appel au matériel. Il devient quasiment impossible à détecter depuis le système original. *Blue Pill* est un autre exemple de rootkit utilisant cette technique.

- **Niveau noyau** - Certains rootkits s'implantent dans les couches du noyau du système d'exploitation : soit dans le noyau lui-même, soit dans des objets exécutés avec un niveau de privilèges équivalent à celui du noyau. Sous GNU/Linux, il s'agit souvent de modules pouvant être chargés par le noyau, et sous Windows de pilotes. Avec un tel niveau de privilèges, la détection et l'éradication du rootkit n'est souvent possible que de manière externe au système en redémarrant depuis un système sain, installé sur CD, sur une clé USB ou par réseau.

- **Niveau bibliothèque** - À ce niveau, le rootkit détourne l'utilisation de bibliothèques légitimes du système d'exploitation. Plusieurs techniques peuvent être utilisées. On peut *patcher* une bibliothèque, c'est-à-dire lui ajouter du code. On peut aussi détourner l'appel d'un objet par hooking, ce qui revient à appeler une « *autre fonction* » puis à revenir à la fonction initiale, pour que le

[7] Ric Vieler, *Professional rootkits*, Indianapolis, IN, Wiley/Wrox, coll. « Wrox professional guides », 2007, 334 p.

détournement soit transparent du point de vue fonctionnel. Enfin, on peut remplacer des appels système par du code malveillant. Ce type de rootkit est assez fréquent, mais il est aussi le plus facile à contrer, notamment par un contrôle d'intégrité des fichiers essentiels, en surveillant leur empreinte grâce à une fonction de hachage ; par une détection de signature du programme malveillant ; ou par exemple, par un examen des *hooks* au moyen d'outils comme unhide sous GNU/Linux ou HijackThis sous Windows.

- **Niveau applicatif** - Un rootkit applicatif implante des programmes malveillants de type cheval de Troie, au niveau utilisateur. Ces programmes prennent la place de programmes légitimes par usurpation d'identité ou en modifiant le comportement, afin de prendre le contrôle des ressources accessibles par ces programmes. Par exemple, une application de traitement de texte peut être remplacée par une version malicieuse et donner accès aux fonctions permettant de lire et d'écrire un fichier dans une partie de l'arborescence.

## II.2.5.2. MODE OPERATOIRE D'UN ROOTKIT

1. **Contamination** - Le mode de contamination par un rootkit se fait en deux phases :

   - La première phase d'action d'un *rootkit* sert généralement à trouver un hôte vulnérable par *balayage* d'un ensemble d'adresses IP ou grâce à une base de données d'IP vulnérables.

   - L'étape suivante consiste à chercher à obtenir un accès au système, sans forcément que celui-ci soit *un accès privilégié* (*ou en mode administrateur*). Il existe trois manières d'obtenir un accès au système, en suivant les techniques habituelles des programmes malveillants :

     → Mettre en œuvre un exploit, c'est-à-dire profiter d'une vulnérabilité de sécurité connue ou non, à n'importe quel niveau du système (application, système d'exploitation, BIOS, etc.). Cette mise en œuvre peut être le fait d'un virus, mais elle résulte aussi, souvent, de *botnets* qui réalisent des scans de machines afin d'identifier et d'exploiter les failles utiles à l'attaque.

     → Même s'il n'est pas un virus à proprement parler, un rootkit peut utiliser des *techniques virales* pour se transmettre, notamment via un cheval de Troie. Un virus peut avoir pour objet de répandre des rootkits sur les machines infectées. *A contrario*, un virus peut aussi utiliser les techniques utilisées par des rootkits pour parfaire sa dissimulation.

→ Enfin, *l'attaque par force brute* permet d'accéder au système, en profitant de la faiblesse des mots de passe mis en œuvre par certains utilisateurs. À ces fins, il suffit de tester les mots de passe les plus courants.

2. **La modification du système** - Une fois la contamination effectuée et l'accès obtenu, la phase suivante consiste à installer, au moyen de son script d'installation, les objets et outils nécessaires au rootkit; c'est-à-dire les objets (programmes, bibliothèques) permettant la mise en place de la *charge utile du rootkit*, s'ils n'ont pas pu être installés durant la phase de contamination, ainsi que les outils et les modifications nécessaires à la dissimulation. L'opération consiste en l'ouverture de portes dérobées, afin de permettre le contrôle des machines (pc, serveurs, etc.), et, d'y installer la charge utile. Le tout, afin de pérenniser l'accès au système, ceci constitue une technique très fréquemment usitée.

3. **La Dissimulation** - Le rootkit cherche à dissimuler son activité pour minimiser le risque qu'on le découvre, afin de profiter le plus longtemps possible de l'accès frauduleux, mais aussi pour rendre sa désinstallation difficile. Il va notamment dissimuler ses propres fichiers, les autres fichiers utilisés par l'attaquant, les processus qu'il exécute et les connexions qu'il va ouvrir. Cette faculté de dissimulation le différencie des virus, qui cherchent principalement à se répandre, bien que ces deux fonctions soient parfois jumelées pour une efficacité supérieure. Plusieurs méthodes de dissimulation peuvent être combinées.

4. **Le Maintien de l'accès** - Un *rootkit* doit pouvoir être manipulé à distance par un attaquant. Celui-ci cherche donc souvent à maintenir un shell (ou « *interpréteur de commandes* ») disponible idéalement à n'importe quel moment (*ou au moins durant l'installation du rootkit*), en remplaçant des commandes comme *ping* ou *xterm*. Généralement, l'attaquant installe plusieurs de ces portes dérobées au cas où l'une viendrait à être découverte et supprimée.

5. **La Mise en place de la charge utile** - La charge utile est la partie active du rootkit (*programme malveillant en général*), dont le rôle est d'accomplir les tâches assignées. Cette charge utile permet d'avoir accès aux ressources de la machine infectée, et notamment le processeur, pour décrypter des mots de passe, pour effectuer des calculs distribués à des fins malveillantes ou pour mettre en œuvre (ou détourner l'usage légitime) des applications comme un serveur de messagerie afin d'envoyer des mails (*pourriel* ou *spam*) en quantité. Les ressources réseaux intéressent également les attaquants, la machine pouvant alors servir de base pour d'autres attaques (*exploits*) ou pour inspecter, sniffer l'activité réseau. La machine infectée peut aussi devenir le point de départ pour d'autres attaques, sur internet, ou sur l'intranet, comme un déni de service. La prise de contrôle de la machine offre la

possibilité de constituer un réseau de type botnet (*la machine infectée devenant alors une machine zombie, comme dans le cas du botnet Srizbi*[8]), ou d'accéder à d'autres machines, par rebond.

6. **Elévation du niveau de privilège** - l'élévation de privilège est souvent nécessaire pour que le camouflage soit efficace : le rootkit peut utiliser certains exploits afin de parfaire sa dissimulation en opérant à un niveau de privilège très élevé, pour atteindre des bibliothèques du système, des éléments du noyau, pour désactiver les défenses du système, etc.

## II.2.6. LES PORTES DEROBEES

Dans un logiciel, une **porte dérobée** (de l'anglais *backdoor*, littéralement *porte de derrière*) est une fonctionnalité inconnue de l'utilisateur légitime, qui donne un accès secret au logiciel. L'introduction d'une porte dérobée dans un logiciel à l'insu de son utilisateur transforme le logiciel en cheval de Troie.

### II.2.6.1. TECHNIQUES UTILISEES

Une porte dérobée peut être introduite soit par le développeur du logiciel, soit par un tiers. La personne connaissant la porte dérobée peut l'utiliser pour surveiller les activités du logiciel, voire en prendre le contrôle (par contournement de l'authentification). Enfin, selon l'étendue des droits que le système d'exploitation donne au logiciel contenant la porte dérobée, le contrôle peut s'étendre à l'ensemble des opérations de l'ordinateur. La généralisation de la mise en réseau des ordinateurs rend les portes dérobées nettement plus utiles que du temps où un accès physique à l'ordinateur était la règle. Parmi les motivations amenant les développeurs de logiciel à créer des portes dérobées, il y a :

- l'intérêt pratique d'un accès facile et toujours ouvert au logiciel pour pouvoir mener efficacement les actions de maintenance ;
- la possibilité de désactiver subrepticement le logiciel en cas de désaccord avec son client (non-paiement de licence).

Parmi les motivations amenant les pirates informatiques à installer une porte dérobée :

- la possibilité de surveiller ce que fait l'utilisateur légitime et de copier ou détruire des données ayant une valeur (mots de passe, clé privée pour déchiffrer des messages privés, coordonnées bancaires, secrets commerciaux) ;

[8] Botnet Spams 60 Billion Emails A Day , CyberInsecure.com, 9 mai 2008

- la possibilité de prendre le contrôle d'un ordinateur et de pouvoir l'utiliser pour mener des actions malfaisantes (envoi de pourriels notamment pour l'hameçonnage, de virus informatiques, déni de service) ;

- le contrôle d'un vaste réseau d'ordinateurs (voire *botnet*), qui peut être utilisé pour du chantage au déni de service distribué (DDoS), ou revendu à des criminels.

Pour installer des portes dérobées en masse, les pirates utilisent des vers. Ceux-ci se répandent automatiquement et installent un serveur informatique sur chaque ordinateur infecté. Ensuite le pirate peut se connecter à travers Internet au serveur.

# II.3. ESPIONNAGE INFORMATIQUE

L'espionnage informatique est une surveillance secrète et désobligeante accomplis par un attaquant au moyen de l'outil informatique afin de s'acquérir des informations dont il n'est pas censé obtenir. L'espionnage informatique peut s'effectuer de plusieurs manières, les plus usuelles sont :

- l'homme du milieu (Environnement informatique) ;
- les espiogiciels ;
- les cookies.

## II.3.1. L'HOMME DU MILIEU

Lorsqu'un pirate, prenant le contrôle d'un équipement du réseau, se place au milieu d'une communication il peut écouter ou modifier celle-ci. On parle alors de « l'homme du milieu » *(man in the middle)*. Les points sensibles permettant cette technique sont les plus souvent les protocoles :

- **DHCP** : ce protocole n'est pas sécurisé et un pirate peut fournir à une victime des paramètres réseau qu'il contrôle. Solution : IP fixe.
- **ARP** : si le pirate est dans le même sous réseau que la victime et le serveur (même si commutateur), il peut envoyer régulièrement des paquets ARP signalant un changement d'adresse MAC aux deux extrémités. Solution : ARP statique.
- **ICMP** : Un routeur peut émettre un ICMP-redirect pour signaler un raccourci, le pirate peut alors demander de passer par lui. Solution : refuser ICMP-redirect ou seulement vers des routeurs identifiés.
- **RIP** : Le pirate envoie une table de routage à un routeur indiquant un chemin à moindre coût et passant par un routeur dont il a le contrôle. Solution : nouvelle version de RIP qui intègre une identification des routeurs de confiance.

- **DNS** : par « ID spoofing » un pirate peut répondre le premier à la requête de la victime et par « cache poisoning » il corrompt le cache d'un serveur DNS. Solution : proxy dans un réseau différent des clients, désactivation de la récursivité, vidage du cache DNS régulier.
- **Proxy HTTP** : Par définition un proxy est en situation d'homme du milieu. Une confiance dans son administrateur est nécessaire de même qu'un contrôle du proxy lors de son départ.
- **Virus** : un virus, éventuellement spécifique à la victime et donc indétectable, peut écrire dans le fichier « *hosts* »… Solution : bloquer les *.vbs et .exe.*

## II.3.2. LES ESPIOGICIELS

Un *logiciel espion* (aussi appelé *mouchard* ou *espiogiciel* ; en anglais ***spyware***) est un logiciel malveillant qui s'installe dans un ordinateur ou autre appareil mobile, dans le but de collecter et transférer des informations sur l'environnement dans lequel il s'est installé, très souvent sans que l'utilisateur en ait connaissance. L'essor de ce type de logiciel est associé à celui d'Internet qui lui sert de moyen de transmission de données. Le terme de *logiciel espion*, est une traduction du mot anglais *spyware*, qui est une contraction de *spy* (espion) et *software* (logiciel).

### II.3.2.1. VECTEURS D'INFECTION

Les logiciels espions sont souvent inclus dans des logiciels gratuits et s'installent généralement à l'insu de l'utilisateur. Ils ne sont généralement actifs qu'après redémarrage de l'ordinateur. Certains, comme *Gator*, sont furtifs et ne se retrouvent donc pas dans la table des processus (*accès : {Ctrl+alt+suppr} pour Windows, {ps} pour Unix).* Un logiciel anti-espion performant peut toutefois les détecter et envoie une alerte avant leur installation.

Les logiciels espions sont développés principalement par des sociétés proposant de la publicité sur Internet. Pour permettre l'envoi de publicité ciblée, il est nécessaire de bien connaître sa cible. Cette connaissance peut être facilement obtenue par des techniques de profilage dont le logiciel espion fait partie. Le logiciel espion attaque très souvent les *systèmes Microsoft Windows* du fait de leur popularité et surtout du bureau lancé avec la totalité des droits la plupart du temps. Certaines pages Web peuvent, lorsqu'elles sont chargées, installer à l'insu de l'utilisateur un logiciel espion, généralement en utilisant des *failles de sécurité* du navigateur de la victime.

Les logiciels espions sont souvent présents dans des *gratuiciels*, ou des *partagiciels*, afin de rentabiliser leur développement. Certains gratuiciels cessent de fonctionner après la suppression de l'espiogiciel associé. On ne connaît pas *de logiciels libres* comme *Mozilla Firefox* qui contiennent des logiciels espions. Enfin, certains administrateurs systèmes ou administrateurs réseaux installent ce type de logiciel pour surveiller à distance l'activité de leurs ordinateurs, sans avoir à se connecter dessus.

Pour rester dans la légalité, il est obligatoire de signaler la présence d'un logiciel espion sur le téléphone ou l'ordinateur. Les principaux vecteurs d'infections sont :

- les logiciels de cassage de protection (*type cracks et keygens*) ;
- les faux codecs ;
- certains logiciels gratuits (*certaines barres d'outils ou utilitaires par exemple*) ;
- les faux logiciels de sécurité (*rogues*) ;
- la navigation sur des sites douteux, notamment ceux au contenu illégal ;
- les pièces jointes et les vers par messagerie instantanée.

## II.3.2.2. MODE OPÉRATOIRE DES LOGICIELS ESPIONS

Un logiciel espion est composé de trois mécanismes distincts :

- Le mécanisme d'infection, qui installe le logiciel. Ce mécanisme est identique à celui utilisé par les *virus,* les *vers* ou les *chevaux de Troie.* Par exemple, l'espiogiciel *Cydoor* utilise le logiciel grand public *Kazaa* comme vecteur d'infection ;
- Le mécanisme assurant la collecte d'information. Pour l'espiogiciel Cydoor, la collecte consiste à enregistrer tout ce que l'utilisateur recherche et télécharge via le logiciel Kazaa ;
- Le mécanisme assurant la transmission à un tiers. Ce mécanisme est généralement assuré via le réseau Internet. Le tiers peut être le concepteur du programme ou une entreprise.

Le logiciel espion peut afficher des offres publicitaires, télécharger un virus, installer un cheval de troie (ce que fait *WhenU. SaveNow*, par exemple), capturer des mots de passe en enregistrant les touches pressées au clavier (*keyloggers*), espionner les programmes exécutés à telle ou telle heure, ou encore espionner les sites Internet visités.

## II.3.2.3. PRÉVENTION CONTRE LES LOGICIELS ESPIONS

De manière générale, avant d'installer un logiciel, l'utilisateur devrait être sûr de sa provenance, qu'il s'agisse d'un téléchargement sur internet ou d'un cédérom. Pour limiter les risques, l'internaute devrait privilégier les sites de téléchargement connus ou le site de l'éditeur, et prendre des renseignements complémentaires sur ces sites ou sur des forums spécialisés.

Pour les utilisateurs non-néophytes, l'utilisation des logiciels libres peut être un moyen de lutter contre les logiciels espions. En effet, les sources de ces logiciels sont disponibles, vérifiables et modifiables, ce qui permet la détection et l'élimination de logiciels espions de ces programmes s'ils en contiennent. Dans les logiciels non libres les sources ne sont pas disponibles, il est donc plus difficile de détecter la présence de ce genre de menace et impossible de l'éliminer. Certains programmes soi-disant destinés à lutter contre les logiciels espions contiennent eux-mêmes ce type de menace, ou se révèlent totalement inefficaces avec pour seul but de facturer une licence d'utilisation (cas de Spyware Assassin par exemple)[9].

Le contrôle des flux sortants est la plupart du temps réalisé par l'administrateur réseau. Par l'intermédiaire d'un *pare-feu*, le contrôle des flux sortants bloque toute connexion qui tente de s'effectuer à partir de l'ordinateur (ou du réseau interne) vers l'extérieur (généralement Internet), sauf les connexions autorisées préalablement (on autorise généralement les connexions vers des sites Web, mais on autorise moins souvent le poste-à-poste). Même si le contrôle des flux sortants est encore peu mis en place à l'heure actuelle, il est primordial dans la compréhension et le blocage de certains problèmes, comme la présence de logiciels espions, car ils vont être amenés à se connecter à l'extérieur pour envoyer les informations qu'ils auront recueillies.

## II.3.2.4. LOGICIELS ANTI-ESPIONS

Il existe plusieurs logiciels spécialisés dans la détection et la suppression de spywares, mais leur utilisation tend à être désuète, car la plupart des logiciels antivirus et des anti-malwares (comme *Malwarebytes' Anti-Malware*) proposent de traiter ce type de programme indésirable. À noter que certains programmes malveillants, appelés *rogues*, sont de faux anti-espions qui installent en fait des spywares.

La plupart des anti-spywares gratuits (comme *A-squared* ou *Spybot - Search & Destroy*) sont bridés dans leur version gratuite (pas de protection en temps réel par exemple). Certains anti-spywares payants (comme *Terminator, Spyware Doctor, Webroot, etc*.), sont aussi complets que l'antivirus classique. À l'instar des antivirus,

[9] « *Spyware Assassin, une belle arnaque arrêtée* », sur infos-du-net.com, 14 mars 2005.

les logiciels anti-espions utilisent des bases de données fréquemment mises à jour (*certaines mises à jour sont manuelles*). En revanche, contrairement à la croyance populaire, il n'est pas recommandé d'utiliser plusieurs logiciels de détection ou de désinfection (*cela augmente les risques de plantage et de ralentissement de l'ordinateur*).

## II.3.3. LES COOKIES

Un *cookie*[10] (ou **témoin de connexion**[11]) est défini par le protocole de communication HTTP comme étant une suite d'informations envoyée par un serveur HTTP à un client HTTP, que ce dernier retourne lors de chaque interrogation du même serveur HTTP sous certaines conditions.

Le cookie est l'équivalent d'un fichier texte de petite taille, stocké sur le terminal de l'internaute. Existant depuis les années 1990, ils permettent aux développeurs de sites web de conserver des données utilisateur afin de faciliter la navigation et de permettre certaines fonctionnalités. Les cookies ont toujours été plus ou moins controversés car elles contiennent des informations personnelles résiduelles pouvant potentiellement être exploitées par des tiers. Ces informations censées être privées ne le sont pas vraiment, puisqu'elles sont accessibles à un certain point. Il est envoyé en tant qu'en-tête HTTP par le serveur web au navigateur web qui le renvoie inchangé à chaque fois qu'il accède au serveur. Un cookie peut être utilisé pour une authentification, une session (maintenance d'état), et pour stocker une information spécifique sur l'utilisateur, comme les préférences d'un site ou le contenu d'un panier d'achat électronique. Le terme cookie est dérivé de *magic cookie*[12], un concept bien connu dans l'informatique d'UNIX, qui a inspiré l'idée et le nom des cookies de navigation. Quelques alternatives aux cookies existent, chacune a ses propres utilisations, avantages et inconvénients.

Étant généralement stockés sous forme de simples fichiers texte, les cookies ne sont pas exécutables. Ils ne sont ni des logiciels espions ni des virus, bien que des cookies provenant de certains sites soient détectés par plusieurs logiciels antivirus parce qu'ils peuvent permettre de suivre les utilisateurs ayant visité certains sites web. La plupart des navigateurs récents permettent aux utilisateurs de décider s'ils acceptent ou rejettent les cookies. Les utilisateurs peuvent aussi choisir la durée de stockage des cookies. Toutefois, le rejet complet des cookies rend certains sites inutilisables. Par exemple, les paniers d'achat de magasins ou les sites qui exigent une connexion à l'aide d'identifiants (utilisateur et mot de passe).

[10] Un « cookie » est une chaîne de caractère qu'un serveur dépose sur votre disque dur, via votre navigateur, afin normalement d'accélérer ou d'autoriser votre prochaine visite.
[11] Ministère de l'Éducation: Bulletin Officiel de l'Éducation Nationale BO N° 14 du 8 avril 1999
[12] http://www.granddictionnaire.com/ficheOqlf.aspx?Id_Fiche=2075216

## II.3.3.1. UTILISATIONS DES COOKIES

- **Gestion des sessions** - Les cookies peuvent être utilisés pour maintenir les données relatives à l'utilisateur durant sa navigation, mais aussi à travers plusieurs visites. Les cookies ont été introduits pour donner un moyen d'implémenter les paniers d'achat électronique, un dispositif virtuel dans lequel l'utilisateur peut accumuler les articles qu'il veut acheter durant sa navigation sur le site. Le navigateur web renvoie alors cet identifiant de session à chaque requête suivante et les articles du panier sont enregistrés et associés avec ce même identifiant unique de session. Une utilisation fréquente des cookies est utile pour la connexion à un site à l'aide d'identifiants.
- **Personnalisation** - Les cookies peuvent être utilisés pour mémoriser l'information sur l'utilisateur d'un site, dans le but de lui montrer un contenu approprié dans le futur. Beaucoup de sites web utilisent les cookies pour la personnalisation basée sur les préférences des utilisateurs. Les utilisateurs sélectionnent leurs préférences dans un formulaire et envoient celles-ci au serveur. Le serveur encode les préférences dans un cookie et renvoie celui-ci au navigateur. Par la suite, à chaque fois que l'utilisateur accède à une page de ce site, le navigateur renvoie le cookie et donc la liste des préférences ; le serveur peut alors personnaliser la page d'après les préférences de l'utilisateur. Par exemple,. Le moteur de recherche Google permet à ses utilisateurs (*même s'ils ne sont pas enregistrés*) de choisir le nombre de résultats qu'ils veulent voir sur chaque page de résultats.
- **Pistage** - Les cookies de pistage sont utilisés pour suivre les habitudes de navigation des utilisateurs d'internet. Cela peut être fait aussi en partie en utilisant l'adresse IP de l'ordinateur faisant une requête d'une page ou à l'aide de l'en-tête HTTP « *référant* » que le client envoie à chaque requête, mais les cookies permettent une plus grande précision. Cela peut être fait comme dans l'exemple suivant :

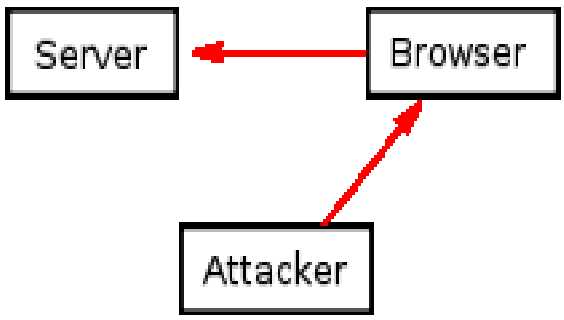
  → Si l'utilisateur fait appel à une page d'un site, et que la requête ne contient pas de cookie, le serveur présume que c'est la première page visitée par l'utilisateur. Le serveur crée alors une chaîne aléatoire et l'envoie au navigateur en même temps que la page demandée.

  → À partir de ce moment, le cookie sera automatiquement envoyé par le navigateur à chaque fois qu'une nouvelle page du site sera appelée. Le serveur enverra la page comme d'habitude, mais enregistrera aussi l'URL de la page appelée, la date, l'heure de la requête et le cookie dans un fichier de journalisation.

## II.3.3.2. INCONVENIENTS DES COOKIES

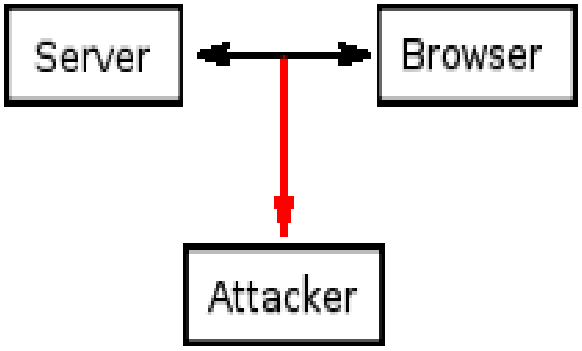
En plus des problèmes d'atteinte à la vie privée, les cookies ont aussi quelques inconvénients techniques. En particulier, ils n'identifient pas toujours exactement les utilisateurs, ils peuvent ralentir la performance des sites lorsqu'en grand nombre, ils peuvent être utilisés pour des attaques de sécurité et ils sont en oppositions avec le transfert représentatif d'état, style architectural du logiciel. On peut citer :

- **Identification imprécise** - Si plus d'un navigateur est utilisé sur un ordinateur, dans chacun d'eux, il y a toujours une unité de stockage séparée pour les cookies. Par conséquent les cookies n'identifient pas une personne, mais la combinaison d'un compte utilisateur, d'un ordinateur, et d'un navigateur web. Ainsi, n'importe qui peut utiliser ces comptes, les ordinateurs, ou les navigateurs qui ont la panoplie des cookies. De même, les cookies ne font pas la différence entre les multiples utilisateurs qui partagent le même compte d'utilisateur, l'ordinateur, et le navigateur comme dans les « *internet cafés* » ou tous lieux donnant accès libre à des ressources informatiques.

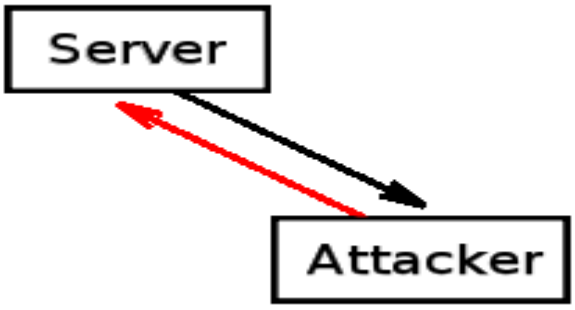



- **Vol de cookie** - Durant l'opération normale, les cookies sont renvoyés entre le serveur (*ou un groupe de serveurs dans le même domaine*) et le navigateur de l'ordinateur de l'utilisateur. Puisque les cookies peuvent contenir des informations sensibles (*nom de l'utilisateur, un mot de passe utilisé pour une authentification, etc.*), leurs valeurs ne devraient pas être accessibles aux autres ordinateurs. Le vol de cookie est un acte d'interception des cookies par un tiers non autorisé. Les cookies peuvent être volés via un renifleur de paquets dans une attaque appelée détournement de session. Le trafic sur le net peut être intercepté et lu par les ordinateurs autres que ceux qui envoient et qui reçoivent (*particulièrement sur l'espace public Wi-Fi non-chiffré*). Ce trafic inclut les cookies envoyés sur des sessions utilisant le protocole HTTP ordinaire. Quand le trafic réseau n'est pas chiffré, des utilisateurs malveillants peuvent ainsi lire les communications d'autres utilisateurs sur le réseau en utilisant des « renifleurs de paquets ».

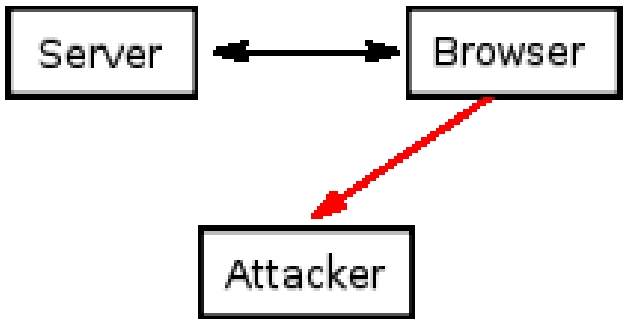
Ce problème peut être surmonté en chiffrant la connexion entre l'ordinateur de l'utilisateur et le serveur par l'emploi du protocole HTTPS. Un serveur peut spécifier un *drapeau sécurisé* tout en mettant en place un cookie ; le navigateur l'enverra seulement sur une ligne sécurisée, comme une connexion en SSL.




- **Modification de cookie** - Dès que les cookies ont besoin d'être stockés et renvoyés inchangés au serveur, un attaquant peut modifier la valeur des cookies avant leur renvoi au serveur. Par exemple, si un cookie contient la valeur totale que l'utilisateur doit payer pour les articles mis dans le panier du magasin, en changeant cette valeur le serveur est exposé au risque de faire payer moins l'attaquant que le prix de départ. Le procédé de modifier la valeur des cookies est appelé *cookie poisoning* et peut être utilisé après un vol de cookie pour rendre l'attaque persistante.




- **Manipulation de cookie entre sites web** - Chaque site est supposé avoir ses propres cookies, donc un site ne devrait pas être capable de modifier ou créer des cookies associés à un autre site. Une faille de sécurité d'un navigateur web peut permettre à des sites malveillants d'enfreindre cette règle. L'exploitation d'une telle faille est couramment appelée *cross-site cooking*. Le but de telles attaques peut être le vol de l'identifiant de session. Les utilisateurs devraient utiliser les versions les plus récentes des navigateurs web dans lesquels ces vulnérabilités sont pratiquement éliminées.




- **État contradictoire entre le client et le serveur -** L'utilisation des cookies peut générer une contradiction entre l'état du client et l'état stocké dans le cookie. Si l'utilisateur acquiert un cookie et clique sur le bouton « Retour » du navigateur, l'état du navigateur n'est généralement pas le même qu'avant cette acquisition. Par exemple, si le panier d'une boutique en ligne est réalisé en utilisant les cookies, le contenu du panier ne peut pas changer quand l'utilisateur revient dans l'historique du navigateur : si l'utilisateur presse sur un bouton pour ajouter un article dans son panier et clique sur le bouton « Retour », l'article reste dans celui-ci. Cela n'est peut-être pas l'intention de l'utilisateur, qui veut certainement annuler l'ajout de l'article. Cela peut mener à un manque de fiabilité, de la confusion, et des bugs. Les développeurs web doivent donc être conscients de ce problème et mettre en œuvre des mesures visant à gérer des situations comme celle-ci.

- **Échéance de cookie -** Les cookies permanents ont été critiqués par les experts de la sécurité de la vie privée pour ne pas être prévus pour expirer assez tôt, et de ce fait permettre aux sites web de suivre les utilisateurs et de construire au fur et à mesure leur profil. Cet aspect des cookies fait partie aussi du problème de détournement de session, parce qu'un cookie permanent volé peut être utilisé pour se faire passer pour un utilisateur pour une période de temps considérable.

# TROISIEME CHAPITRE – SYSTEMES DE PROTECTION INFORMATIQUE

*Un système de protection informatique* est un ensemble des techniques permettant de se prémunir contre les attaques et piraterie informatique, en interdisant la copie de contenus d'un support (logiciel) ou en rendant inutilisable toute intrusion dans le système.

Les systèmes de protection informatique les plus connus sont :

- Les anti-virus ;
- Les systèmes de détection (et prévention) d'intrusion (IDS) ;
- Les firewalls ;

## III.1. LES ANTI-VIRUS

*Les antivirus* sont des logiciels conçus pour identifier, neutraliser et éliminer des logiciels malveillants (*dont les virus informatique ne sont qu'une catégorie).* Ces derniers peuvent se baser sur l'exploitation de failles de sécurité, mais il peut également s'agir de logiciels modifiant ou supprimant des fichiers, que ce soit des documents de l'utilisateur stockés sur l'ordinateur infecté, ou des fichiers nécessaires au bon fonctionnement de l'ordinateur (*le plus souvent ceux du système d'exploitation).*

Il est intéressant de noter qu'une fois un fichier infecté, il ne l'est jamais deux fois. En effet, un virus est programmé de telle sorte qu'il signe le fichier dès qu'il est contaminé. On parle ainsi de signature de virus. Cette signature consiste en une suite de bits apposée au fichier. Cette suite, une fois décelée, permettra de reconnaitre le virus. Lorsque le virus est détecté par l'antivirus, plusieurs possibilités sont offertes pour l'éradiquer :

- Supprimer le fichier infecté ;
- Supprimer le code malicieux du fichier infecté ;
- Placer le ou les fichiers infectés en "quarantaine" pour un traitement futur.

Voici les anti-virus les plus populaires selon leurs finalités :

- **Les principaux anti-virus des PC and Serveurs** : AhnLab V3 Internet Security - Avast Antivirus - AVG - Avira AntiVirus - Bitdefender - ClamWin - ClamAV - Comodo Antivirus - Comodo Internet Security - Dr. Web - NOD32 - F-Secure - F-PROT - Fortinet - G Data Software - Advanced SystemCare - iolo System Shield - Kaspersky AntiVirus - Kaspersky Internet Security -

KingSoft - Mac Internet Security - McAfee VirusScan - Microsoft Security Essentials - Windows Defender - Panda - 360 Safeguard - Outpost Security Suite - Sophos - Symantec Endpoint Protection - Immunet - Element Anti-Virus - Norton AntiVirus - Norton Internet Security - Spyware Doctor - VirusBarrier - Trend Micro Internet Security - TrustPort - Vba32 AntiVirus - Zone Alarm.

- **Les principaux anti-virus des mobiles et tablettes :** AhnLab Mobile Security (en) - Avast Antivirus - AVG - Avira Free Android Security - Bitdefender Mobile Security - CM Security - Comodo Mobile Security - Dr. Web Mobile Security Suite - ESET Mobile Security - F-Secure Mobile Security - G Data MobileSecurity - Lookout Mobile Security - McAfee Mobile Security - FireAMP Mobile - Trend Micro Mobile Security - TrustPort Mobile Security - VirusBarrier.

## III.1.1. FONCTIONNEMENT DE L'ANTI-VIRUS

Un logiciel antivirus vérifie les fichiers et courriers électroniques, les secteurs de démarrage (*afin de détecter les virus de boot*), mais aussi la mémoire vive de l'ordinateur, les médias amovibles (*clefs USB, CD, DVD, etc*.), les données qui transitent sur les éventuels réseaux (*dont internet*) …. Différentes méthodes sont possibles :

- Les principaux antivirus du marché se concentrent sur des fichiers et comparent alors *la signature virale* du virus aux codes à vérifier ;
- La *méthode heuristique* est la méthode la plus puissante, tendant à découvrir un code malveillant par son comportement. Elle essaie de le détecter en analysant le code d'un programme inconnu. Parfois de fausses alertes peuvent être provoquées ;
- L'*analyse de forme* repose sur du filtrage basé entre des règles *rege-xp* ou autres, mises dans un fichier *junk*. Cette dernière méthode peut être très efficace pour les serveurs de messagerie électronique supportant les rege-xp type postfix puisqu'elle ne repose pas sur un fichier de signatures.

Les antivirus peuvent balayer le contenu d'un disque dur, mais également la mémoire vive de l'ordinateur. Pour les anti-virus les plus modernes, ils agissent en amont de la machine en scrutant les échanges de fichiers avec l'extérieur, aussi bien en flux descendant (téléchargement) que montant (*téléchargementt ou upload*). Ainsi, les courriels sont examinés, mais aussi les fichiers copiés sur ou à partir de supports amovibles tels que cédéroms, disquettes, connexions réseau, clefs USB…

## III.1.2. TECHNIQUES DE DETECTION DES ANTI-VIRUS

En général, la guerre entre virus et antivirus est bien réelle. Dès qu'un groupe agit, le camp opposé tente de trouver la parade. Pour détecter les virus, les antivirus doivent user de plusieurs techniques spécialement :

- **Le scanning des signatures (Dictionnaire)** - La détection des virus consiste à la recherche de ces signatures à partir d'une base de données de signatures (*on parle également de définitions de virus*). Le principal avantage de cette technique est qu'il est possible de détecter le virus avant qu'il ne soit en action. Cependant, il est nécessaire que sa signature soit présente dans la base de données afin qu'il soit détecté. De plus, il est nécessaire de tenir la base régulièrement à jour afin de pouvoir détecter les nouveaux virus.

- **Le moniteur de comportement** : Il s'agit ici de contrôler en continu toute activité suspecte telles que les lectures et écritures dans des fichiers exécutables, les tentatives d'écriture dans les secteurs de partitions et de boot du disque.

- **Liste blanche** - est une technique de plus en plus utilisée pour lutter contre les logiciels malveillants. Au lieu de rechercher les logiciels connus comme malveillants, on empêche l'exécution de tout logiciel à l'exception de ceux qui sont considérés comme fiables par l'administrateur système. En adoptant cette méthode de blocage par défaut, on évite les problèmes inhérents à la mise à jour du fichier de signatures virales. De plus, elle permet d'empêcher l'exécution de logiciels indésirables. Étant donné que les entreprises modernes possèdent de nombreuses applications considérées comme fiables, l'efficacité de cette technique dépend de la capacité de l'administrateur à établir et mettre à jour la liste blanche. Cette tâche peut être facilitée par l'utilisation d'outils d'automatisation des processus d'inventaire et de maintenance.

- **Le contrôleur d'intégrité** : Le principe est que l'antivirus maintienne une liste des fichiers exécutables associés à leur taille, leur date de création, de modification, voire un **CRC** (*Contrôleur Redondance Cyclique*). L'utilisation du CRC permet de vérifier qu'un exécutable n'a pas été modifié en comparant sa somme de contrôle avant et après son exécution. En effet, en dehors d'une mise à jour explicite du fichier, un fichier exécutable n'est pas sensé être modifié. Le même type de vérifications peut être instauré avec la date et l'heure de modification. Cependant, il suffira aux virus de mémoriser ces valeurs afin de pouvoir les restaurer par la suite.

- **L'analyse heuristique** : A la différence du moniteur de comportement qui détecte les modifications causées par les virus, l'analyse heuristique tente de détecter les virus avant leur exécution, en cherchant des portions de code suspectes. Il pourrait par exemple chercher des séquences de lecture suivies de séquences d'écriture sur un même fichier exécutable. Cette technique permet donc de détecter des virus même s'ils ne sont pas présents dans la base de données, puisque l'analyseur teste des séquences d'instructions communes à de nombreux virus.

## III.2. LES SYSTEMES DE DETECTION D'INTRUSION

Un *système de détection d'intrusion* (ou IDS : *Intrusion Detection System*) est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (*un réseau ou un hôte*). Il permet ainsi d'avoir une connaissance sur les tentatives réussies comme échouées des intrusions. Les IDS, les plus connus selon leurs différentes catégories sont :

- **Les IDS réseau (NIDS)** - Snort ; Bro ; Suricata ; Enterasys ; Check Point ; Tipping point ; etc.
- Les IDS système (HIDS) - AIDE ; Chkrootkit ; DarkSpy ; Fail2ban ; IceSword ; OSSEC ; Rkhunter ; Rootkit Unhooker; Tripwire ; etc.
- Les IDS hybride - Prelude; OSSIM ; etc.

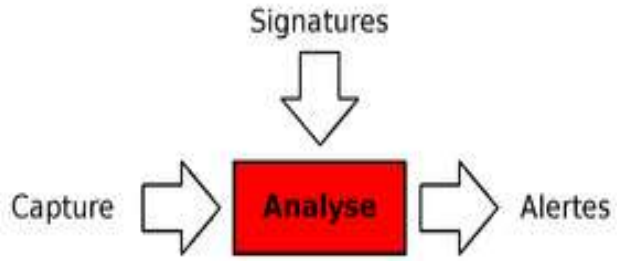
### III.2.1. TYPOLOGIE DE SYSTÈMES DE DÉTECTION D'INTRUSION

Il existe trois grandes familles distinctes d'IDS :

- **les NIDS** (*Network Based Intrusion Detection System*), qui surveillent l'état de la sécurité au niveau du réseau ;
- **les HIDS** (*HostBased Intrusion Detection System*), qui surveillent l'état de la sécurité au niveau des hôtes. Les HIDS sont particulièrement efficaces pour déterminer si un hôte est contaminé et les NIDS permettent de surveiller l'ensemble d'un réseau contrairement à un HIDS qui est restreint à un hôte.
- **les IDS hybrides**, qui utilisent les NIDS et HIDS pour avoir des alertes plus pertinentes.

## III.2.1.1. LES NIDS (IDS réseau)

Un NIDS se découpe en trois grandes parties : *la capture, les signatures* et *les alertes.* Cependant à ces trois principales parties, s'ajoute de même 2 principales techniques :




- **La Capture** - La capture sert à la récupération de trafic réseau. En général cela se fait en temps réel, bien que certains NIDS permettent l'analyse de trafic capturé précédemment. La plupart des NIDS utilisent la bibliothèque standard de capture de paquets *libpcap*. La bibliothèque de capture de paquets *Packet Capture Library* est portée sur quasiment toutes les plates-formes, ce qui permet en général aux IDS réseau de suivre. Le fonctionnement de la capture d'un NIDS est donc en général fortement lié à cette *libpcap*. Son mode de fonctionnement est de copier (*sous Linux*) tout paquet arrivant au niveau de la couche liaison de données du système d'exploitation. Une fois ce paquet copié, il lui est appliqué un *filtre BPF* (*Berkeley Packet Filter*), correspondant à l'affinage de ce que l'IDS cherche à récupérer comme information. Il se peut que certains paquets soient ignorés car sous une forte charge, le système d'exploitation ne les copiera pas. Le comportement de la *libpcap* est différent dans le monde *BSD*, puisqu'il lui attache le fichier périphérique « */dev/bpf* », permettant ainsi aux NIDS de ne pas avoir besoin des droits super utilisateur pour capturer le trafic mais simplement de pouvoir lire sur ce fichier sur lequel les filtres sont directement compilés. Aussi, le trafic analysé n'est pas forcément égal à celui du trafic entrant, étant donné que la *libpcap* agit à une couche en dessous du pare-feu (*qui agit au niveau réseau*).

- **Les Signatures -** Les bibliothèques de signatures (approche par scénario) rendent la démarche d'analyse similaire à celle des anti-virus quand ceux-ci s'appuient sur des signatures d'attaques. Ainsi, le NIDS est efficace s'il connaît l'attaque, mais inefficace dans le cas contraire. Les outils commerciaux ou libres ont évolué pour proposer une personnalisation de la signature afin de faire face à des attaques dont on ne connaît qu'une partie des éléments. Les outils à base de signatures requièrent des mises à jour très régulières.

Les NIDS ont pour avantage d'être des systèmes temps réel et ont la possibilité de découvrir des attaques ciblant plusieurs machines à la fois. Leurs inconvénients sont le taux élevé de faux positifs qu'ils génèrent, le fait que les signatures aient toujours du retard sur les attaques de type « *0day* » et qu'ils puissent être la cible d'une attaque.

- **Les Alertes -** Les alertes sont généralement stockées dans les journaux du système. Cependant il existe une norme qui permet d'en formaliser le contenu, afin de permettre à différents éléments de sécurité d'interopérer. Ce format s'appelle IDMEF (pour *Intrusion Detection Message Exchange Format*) décrit dans la RFC 4765. Le format IDMEF est popularisé par le projet Prelude, qui offre une infrastructure permettant aux IDS de ne pas avoir à s'occuper de l'envoi des alertes. Cela permet aux IDS de n'avoir qu'à décrire les informations qu'ils connaissent et « *Prelude* » se charge de les stocker pour permettre une visualisation humaine ultérieure.

## DIFFERENTES TECHNIQUES DES NIDS

- **La recherche de motif (pattern matching)** - La recherche de motif est ce qui permet à un NIDS de trouver le plus rapidement possible les informations dans un paquet réseau.

- **L'Analyse** - À partir des éléments donnés dans l'introduction, le moteur d'analyse met ces éléments de relation en employant plusieurs techniques : la **défragmentation**, la **dissection** protocolaire ou encore l'analyse comportementale :

  → **La défragmentation** - Les paquets dépassant une certaine taille (qui en général est de 1 500 octets) sont fragmentés. La fragmentation de l'en-tête de la couche transport étant aussi possible, cela rendait les NIDS vulnérables aux attaques de Stick et de Snot car les paquets fragmentés n'étaient pas analysés. Les NIDS ont le devoir de défragmenter les paquets avant analyse, afin de ne pas manquer une attaque. Il s'agit d'une opération relativement complexe, étant donné que chaque hôte de destination ne défragmente pas de la même façon, selon le système d'exploitation sur lequel l'attaque est visée. Il s'agit encore d'une technique d'évasion utilisable aujourd'hui car les NIDS ne sont pas forcément configurés correctement pour gérer un cas précis.

  → **La dissection** - La dissection permet de comprendre un protocole donné, de le décoder pour l'analyser. Il s'agit de la partie la plus sensible des NIDS car c'est elle qui est le plus grand vecteur d'attaques.

## III.2.1.2. LES HIDS (IDS machine)

Les HIDS, pour Host based IDS, signifiant "Système de détection d'intrusion machine" sont des IDS dédiés à un matériel ou système d'exploitation. Généralement, contrairement à un NIDS, le HIDS récupère les informations qui lui sont données par le matériel ou le système d'exploitation. Il y a pour cela plusieurs approches : signatures, comportement (statistiques) ou délimitation du périmètre avec un système d'ACL (Access Control List). Un HIDS se comporte comme un *daemon* ou un service standard sur un système hôte qui détecte une activité suspecte en s'appuyant sur une norme. Si les activités s'éloignent de la norme, une alerte est générée. La machine peut être surveillée sur plusieurs points :

- *Activité de la machine* : nombre et listes de processus ainsi que d'utilisateurs, ressources consommées, ...
- *Activité de l'utilisateur* : horaires et durée des connexions, commandes utilisées, messages envoyés, programmes activés, dépassement du périmètre défini...
- *Activité malicieuse* d'un ver, virus ou cheval de Troie.

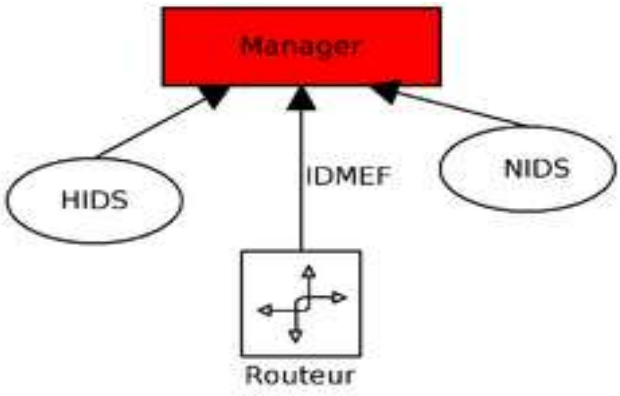
Un autre type d'HIDS cherche les intrusions dans le « noyau » (kernel) du système, et les modifications qui y sont apportées. Certains appellent cette technique « analyse protocolaire ». Très rapide, elle ne nécessite pas de recherche dans une base de signature. Exemples de contrôles pour Windows :

- *EPROCESS* (structure de données en mode noyau contenant des informations qui peuvent permettre de cacher un processus) ;
- Les processus fonctionnant en mode « noyau »
- Les fonctions logicielles système ou de gestion de périphérique présentes dans l'ordinateur.
- La SSDT (System Service Dispatch Table) table utilisée par Windows pour diriger des appels de système vers un traitement approprié : table d'adressage des interruptions ; etc.

Le HIDS a pour avantage de n'avoir que peu de faux positifs, permettant d'avoir des alertes pertinentes. Quant à ses inconvénients il faut configurer un HIDS par poste et il demande une configuration de chaque système.

### III.2.1.3. LES IDS hybride

Les IDS hybrides sont basés sur une architecture distribuée, où chaque composant unifie son format d'envoi d'alerte (*typiquement IDMEF : Intrusion Detection Message Exchange Format*) permettant à des composants divers de communiquer et d'extraire des alertes plus pertinentes.




Les avantages des IDS hybrides sont multiples :

- Moins de faux positifs ;
- Meilleure corrélation ;
- Possibilité de réaction sur les analyseurs.

## III.2.2. TECHNIQUES D'ANALYSE DU TRAFIC DES IDS

Il existe principalement deux techniques d'analyse du trafic, chacune ayant des avantages et des inconvénients :

- **L'analyse comportementale (*AIDS : Anomaly Intrusion Detection System*)** - à partir d'un comportement normal déterminé, l'IDS analyse le comportement des machines. Si un ordinateur se connecte en pleine nuit alors que personne n'est présent, cela pourrait lever une alerte pour l'IDS. Ainsi, dans ce type d'analyse, un profil est dressé et lorsque la machine liée s'éloigne du profil type, l'IDS réagit.

  → **Avantage** : ce type d'analyse permet de détecter des attaques inconnues, elle ne nécessite pas de base de données.

  → **Inconvénient** : cette détection est assez aléatoire, elle peut produire de fausses alertes relativement facilement.

- **L'analyse par scénario (*MIDS : Misuse Intrusion Detection System*)** - l'IDS utilise ici une base de données de signatures d'attaques. Ces signatures peuvent être assimilées à des déroulements d'attaques. En effet, chaque attaque possède des caractéristiques propres (*numéro de port, taille de paquet, protocole employé...).*

Ces caractéristiques peuvent être collectées et placées dans une base de données qu'interrogera l'IDS. Ce type d'IDS utilise les fichiers journaux (log). Dès qu'il détectera des séquences suspectes (*relatives à une signature de sa base de données*), il déclenchera une alerte.

→ **Avantage** : on peut gérer les attaques de façon très précise.

→ **Inconvénient** : on doit maintenir une base de données à jour.

Avertissons que certains IDS analysent uniquement les fichiers systèmes (fichiers d'historique), et d'autres uniquement le trafic réseau. On les nomme respectivement HIDS pour "Host IDS" (protection des machines) et NIDS pour "Network IDS" (protection du réseau). Les IDS ne réagissent pas non plus de la même manière en présence d'une attaque. La plupart agissent passivement, c'est-à-dire qu'une fois l'attaque détectée, ils émettent simplement une alerte. D'autres, beaucoup moins répandus, tentent de contre-attaquer. On dit que ces derniers sont actifs.

## III.2.3. LA CORRÉLATION DES IDS HYBRIDE

La corrélation est une connexion entre deux ou plusieurs éléments, dont un de ces éléments crée ou influence un autre. Elle se traduit plus généralement par la transformation d'une ou plusieurs alertes en attaque. Cela permet de faciliter la compréhension sur les attaques au lieu de s'éparpiller parmi les alertes. Idéalement, elle nécessite un IDS Hybride car plus il y a d'informations hétérogènes sur un événement, plus la corrélation se fait d'une façon pertinente. Les formats ayant été normalisés *(IDMEF*), il ne reste plus qu'à faire des associations afin de détecter des alertes qui n'auraient jamais eu lieu sur un analyseur seul.

Si l'on prend l'exemple d'une authentification échouée, cela génère une alerte de faible intensité. Mais s'il y a une série d'authentifications échouées avec des utilisateurs différents, on peut conclure à une attaque de force brute. La corrélation permet de générer de nouvelles alertes à partir de celles existantes. C'est une étape préalable à une contre-mesure efficace. Il y a diverses façons de faire de la corrélation. Cependant on peut définir deux catégories :

- la **corrélation passive** - correspondant à une génération d'alerte basée sur celles existantes. Nous pouvons prendre par exemple les scans de force brute ssh ;
- la **corrélation active** - qui va chercher les informations correspondant à des alertes émises. Par exemple, lorsqu'une personne se connecte en dehors des heures de travail, cela a une influence importante qui n'aurait pas été en temps normal d'activité.

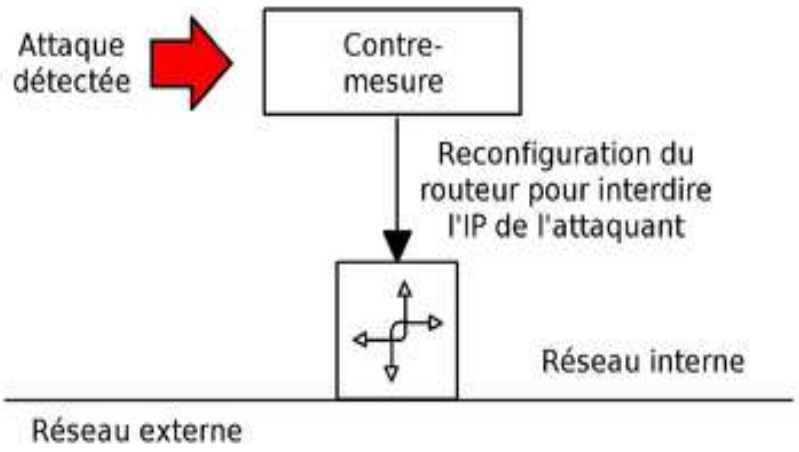
## III.2.4. L'HARMONISATION DES FORMATS

Le format IDMEF (*Intrusion Detection Message Exchange Format*) décrit une alerte de façon objet et exhaustive. Une alerte est le message qui est émis depuis un analyseur, qui est une sonde en langage IDMEF, vers un collecteur. Le but d'IDMEF est de proposer un standard permettant d'avoir une communication homogène quels que soient l'environnement ou les capacités d'un analyseur donné.

Ces alertes sont définies au format XML, offrant une possibilité de validation de chaque message. En général, les implémentations restent binaires, afin d'éviter les problèmes connus d'ajout d'information inutiles en dehors d'XML lorsque l'on envoie un message sur le réseau. Le format IDMEF offre aussi un vocabulaire précis, qu'il est courant d'utiliser dans le domaine de la détection d'intrusions. Par exemple, une classification correspond au nom d'une alerte ; une influence à celui d'un niveau d'attaque.

## III.2.5. LA CONTRE-MESURE DES IDS HYBRIDE

La contre-mesure est l'art de piloter les éléments réseau ou la machine cible, afin d'empêcher une attaque de se propager (*Islanding*) ou de perdurer. Il s'agit d'une procédure assez compliquée et souvent désactivée. Ce qui rend la contre-mesure difficile est la définition d'une attaque d'un point de vue formel. Il n'est pas possible de se baser sur des éléments qui génèrent des faux positifs. Et cela peut aussi engendrer un autre problème où l'attaquant se fait passer pour un client du réseau en générant des motifs d'attaque. Cela peut même bloquer le réseau interne si la contre-mesure est mal configurée. Un système de contre-mesure se configure en général avec une liste blanche, dans laquelle sont mises les IP du réseau interne.

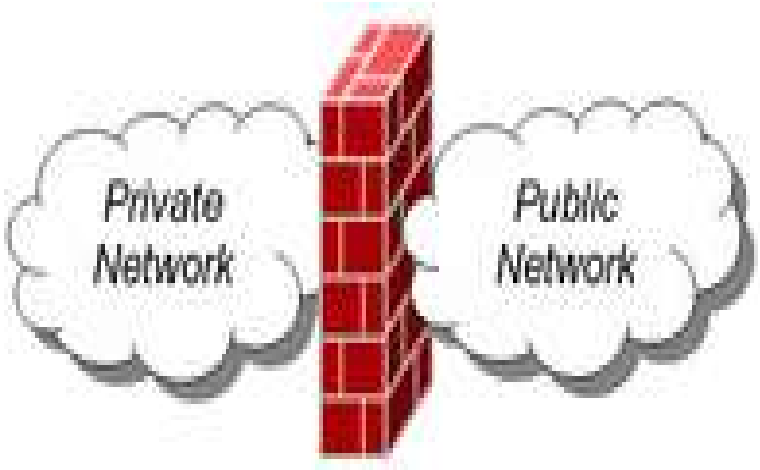


# III.3. LES FIREWALLS (PARE-FEU)

Un *pare-feu*[13] (parfois appelé *coupe-feu*, *garde-barrière*, *barrière de sécurité*, ou encore *firewall*. Dans un environnement Unix BSD (Berkeley Software Distribution), un pare-feu est aussi appelé *packet filter*. Traduction littérale *: mur de feu*) est un logiciel et/ou un matériel permettant de faire respecter la politique de sécurité du réseau, celle-ci définissant quels sont les types de communications autorisés sur ce réseau informatique. Le pare-feu a pour objectif principal de surveiller et contrôler les applications et les flux de données (paquets), en empêchant les connexions non-autorisées sur un réseau informatique ou autres.

En fait, un firewall peut être configuré à de nombreux niveaux :

- *Niveau des adresses IP* : on peut lui faire accepter les flux de données provenant d'une plage d'adresses, ou même d'une adresse uniquement.
- *Niveau des noms de domaine* : il est également possible d'empêcher l'accès à certaines adresses Internet.
- *Niveau des protocoles* : pour empêcher tout transfert FTP, tout accès Telnet, ou encore pour éviter le surf sur Internet (HTTP).
- *Niveau des ports* : pour supprimer le FTP, on peut refuser les connexions sur le port 21.
- *Niveau des mots ou phrases* : semblable aux expressions régulières, il est possible de refuser les paquets dont le contenu renferme des séquences de lettres données.




***Structure simple d'un pare-feu***

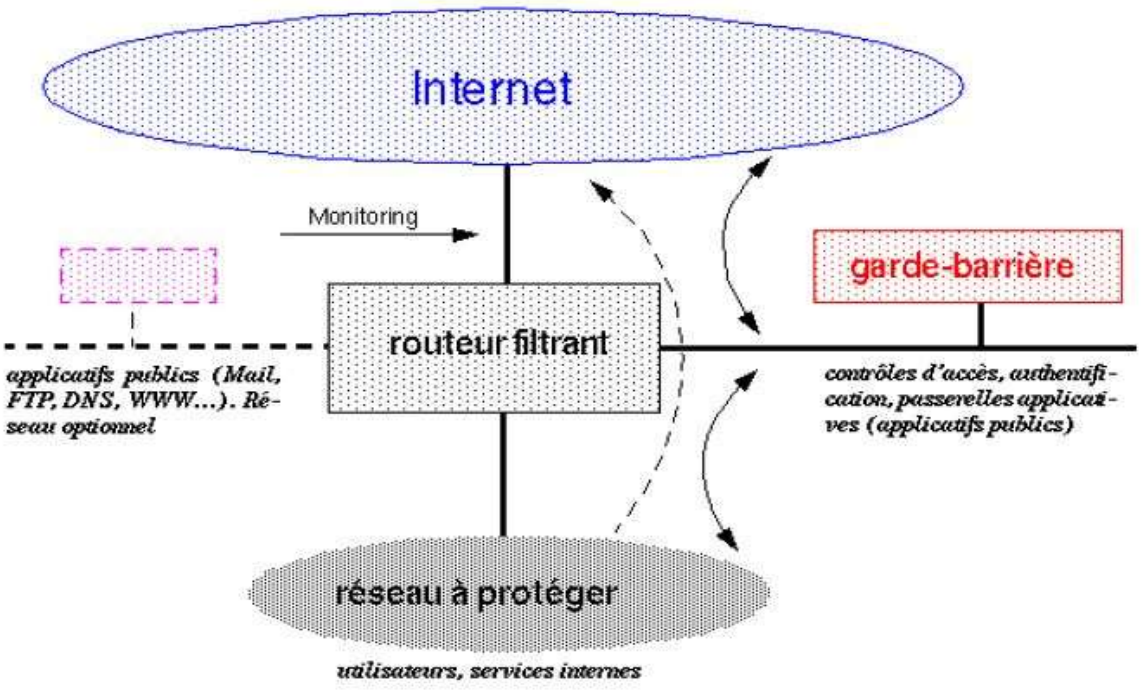
[13] Terme recommandé par la Commission générale de terminologie et de néologie, et couramment employé, chercher *firewall* dans France Terme.

Selon le contexte, le terme peut revêtir différentes significations :

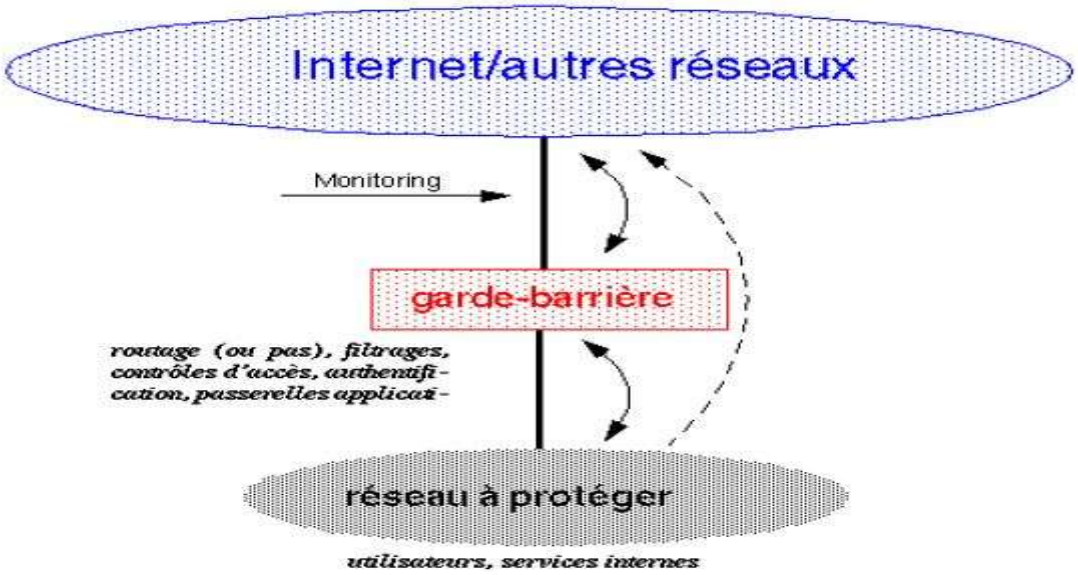
- Dans le domaine de la lutte contre les incendies de forêt, il se réfère aux allées pare-feu destinées à contenir l'extension des feux de forêts ;
- Au théâtre, le déclenchement d'un mécanisme « *pare-feu* » (ou « *coupe-feu* ») permet d'éviter la propagation du feu de la salle vers la scène ;
- Dans le domaine de l'architecture, il fait référence aux portes coupe-feu ou à tout autre dispositif constructif destiné à contenir l'extension d'un incendie ;
- En informatique, l'usage du terme « pare-feu » est donc métaphorique. Sa dénomination, reprend au sens figuré l'intention de "brûler par un mur de feu virtuel" tout ce qui tente d'entrer avec l'intention de nuire dans une machine ou un réseau. Il établit une barrière de protection contre les intrusions et les contaminations venant de l'extérieur.

D'une manière concrète, un pare-feu un système matérielle et immatérielle dédiée au routage entre LAN et Internet. Le trafic est analysé au niveau des datagrammes IP (*adresse, utilisateur, contenu...).* Un datagramme non autorisé sera simplement détruit, IP sachant gérer la perte d'information. Une translation d'adresse pourra éventuellement être effectuée pour plus de sécurité (protocole NAT *Network Address Translation*). Deux types d'architectures peuvent être exploités - *l'architecture classique et l'architecture concentrée* :

- L'architecture classique ;

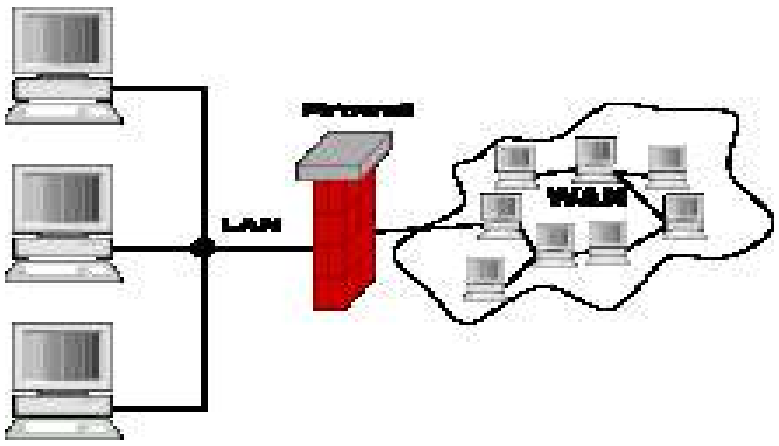


- L'architecture concentrée ;




## III.3.1. PRINCIPES DE FONCTIONNEMENT DES PARE-FEUX

Le pare-feu est jusqu'à ces dernières années est considéré comme une des pierres angulaires de la sécurité d'un réseau informatique (*il perd en importance au fur et à mesure que les communications basculent vers le HTTP sur TLS, court-circuitant tout filtrage*). Il permet d'appliquer une politique d'accès aux ressources réseau (serveurs). Il a pour principale tâche de contrôler le trafic entre différentes zones de confiance, en filtrant les flux de données qui y transitent. Généralement, les zones de confiance incluent Internet (*une zone dont la confiance est nulle*) et au moins un réseau interne (*une zone dont la confiance est plus importante*). Le but est de fournir une connectivité contrôlée et maîtrisée entre des zones de différents niveaux de confiance, grâce à l'application de la politique de sécurité et d'un modèle de connexion basé sur le principe du moindre privilège. On peut alors distinguer 3 types de principes de fonctionnement des pare-feux :



**1. Le filtrage de paquets (Packet Filtering)**

Les paquets sont analysés en les comparants à un ensemble de filtres (*c'est-à-dire à un ensemble de règles*). Les paquets seront alors soit rejetés, soit acceptés et transmis au réseau interne. Lorsque le paquet arrive au firewall, celui-ci analyse les champs IP et TCP/UDP. Ils sont confrontés à chacune des règles spécifiées dans la table des autorisations présente dans le firewall, et configurée par l'administrateur du système. Selon les règles qui autorisent ou refusent la transmission des paquets, le firewall obéira aux ordres. Si un paquet ne satisfait à aucune des règles, il est soit rejeté, soit accepté, suivant la philosophie choisie par l'administrateur réseau :

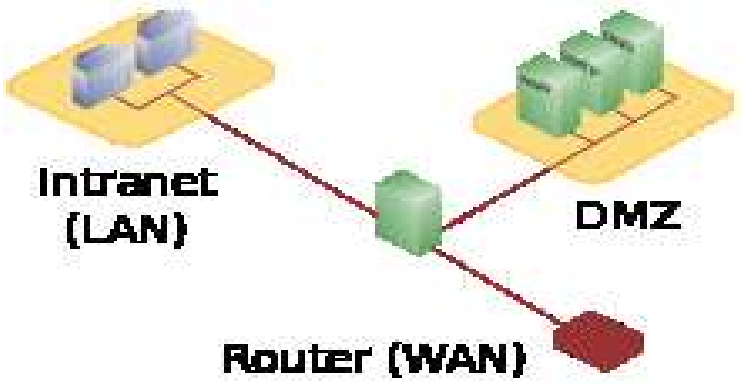
- Ce qui n'est pas expressément permis est interdit ;
- Ce qui n'est pas expressément interdit est permis.

La première de ces deux approches est beaucoup plus sûre. La seconde est plus risquée car elle suppose que l'administrateur est certain d'avoir envisagé tous les cas pouvant engendrer des problèmes. Le filtrage se fait selon divers critères. Les plus courants sont :

- l'origine ou la destination des paquets (adresse IP, ports TCP ou UDP, interface réseau, etc.) ;
- les options contenues dans les données (fragmentation, validité, etc.) ;
- les données elles-mêmes (taille, correspondance à un motif, etc.) ;
- les utilisateurs pour les plus récents.

L'avantage du filtrage par paquet est sa rapidité. Il est de plus relativement simple à implanter dans un réseau. Néanmoins, la sécurité ne peut se baser uniquement sur le filtrage par paquet. D'une part, il est difficile de maintenir un niveau suffisant de sécurité lorsque le nombre de règles augmente. D'autre part, le type d'informations accessibles à ce niveau est limité, en l'occurrence, il n'identifie que la machine et non son utilisateur.

Un pare-feu fait souvent office de routeur et permet ainsi d'isoler le réseau en plusieurs zones de sécurité appelées zones démilitarisées ou DMZ. Ces zones sont séparées suivant le niveau de confiance qu'on leur porte.




**2. Le filtrage du flux (Circuit Filtering)**

Le filtrage de flux ne prête pas attention au contenu des paquets transitant sur la connexion. De ce fait, ce type de filtrage ne peut être utilisé pour assurer l'authentification des parties, ou la sécurité du protocole par l'intermédiaire duquel a lieu la connexion. A la différence du filtrage de paquets, qui est considéré comme permissif, le filtrage de flux est restrictif. En effet, il n'autorisera le flux entre deux entités que si la connexion entre ces deux entités existe. On peut voir ce principe comme la création d'un tunnel entre deux machines. De ce fait, le circuit-filtering ne sera souvent utilisé qu'en complément de l'application gateway.

**3. La passerelle applicative (Application Gateway)**

À la différence du filtrage de paquets, qui analyse les paquets individuellement, l'application gateway permet de limiter les commandes à un service plutôt que de l'interdire. Ce principe de fonctionnement empêche le trafic direct entre le réseau protégé et l'Internet, et ce dans les deux sens. Le trafic interne n'atteindra jamais Internet, et inversement, aucun trafic Internet ne voyagera sur le réseau interne. En effet, chaque client interne se connectera sur un serveur proxy (qui est la base de ce principe). Toutes les communications se feront par l'intermédiaire de celui-ci. Il déterminera si le service demandé par l'utilisateur est permis et se connectera avec le destinataire en cas d'autorisation.

Le destinataire ne connaîtra pas l'adresse de son correspondant. Il ne communiquera qu'avec le serveur proxy, qui jouera en réalité le rôle d'un translateur d'adresse réseau (NAT). La sécurité peut être ici très élevée. Agissant au niveau applicatif, on peut notamment la retrouver l'authentification par mot de passe des utilisateurs. En tant qu'inconvénient, ce principe de fonctionnement supposera que toutes les machines sont configurées pour communiquer avec ce proxy, ce qui impliquera probablement une configuration individuelle de chaque machine ou l'installation de logiciels sur chacune d'elles.

Enfin, le pare-feu est également souvent situé à l'extrémité de tunnel IPsec ou TLS. L'intégration du filtrage de flux et de la gestion du tunnel est en effet nécessaire pour pouvoir à la fois protéger le trafic en confidentialité et intégrité et filtrer ce qui passe dans le tunnel. C'est le cas notamment de plusieurs produits du commerce nommés dans la liste ci-dessous.

Les pare-feux récents embarquent de plus en plus de fonctionnalités, parmi lesquelles on peut citer :

- Filtrage sur adresses IP / protocole ;
- Inspection *stateful* et applicative ;
- Intelligence artificielle pour détecter le trafic anormal ;
- Filtrage applicatif : HTTP (restriction des URL accessibles), Courriel (Anti-pourriel), Logiciel antivirus, anti-logiciel malveillant ;
- Traduction d'adresse réseau ;
- Tunnels IPsec, PPTP, L2TP ;
- Identification des connexions ;
- Serveurs de protocoles de connexion (telnet, SSH), de protocoles de transfert de fichier (SCP) ;
- Clients de protocoles de transfert de fichier (TFTP) ;
- Serveur Web pour offrir une interface de configuration agréable ;
- Serveur mandataire (« *proxy* » en anglais) ;
- Système de détection d'intrusion (« IDS » en anglais) ;
- Système de prévention d'intrusion (« IPS » en anglais).

## III.3.2. CATEGORIES DE PARE-FEU

Il existe 3 modèles de firewalls. Chacun possède des avantages et désagréments. Il faudra donc préalablement analyser les besoins réels en termes de sécurité, ainsi que les coûts engendrés avant toute utilisation :

- **Les firewalls Bridge** - Ce format de mur de feu a l'apparence d'un simple câble réseau, sans machine spécifique. Il est invisible et indétectable pour un pirate, son adresse MAC ne circulant jamais sur le réseau. Placé sur le réseau, le pirate devra donc automatiquement passer par lui pour transmettre des requêtes. On trouvera notamment ce type de firewalls dans des *Switch***.** Ces formats de pare-feu ont ***pour avantages*** : Ils sont relativement peu coûteux et transparent lors de leurs mises en place. Ils présentent comme ***Inconvénients*** *:* Pour les contourner, il suffit d'adapter l'attaque ; et ses fonctionnalités sont souvent restreintes.

- **Les firewalls hardware** - Ils sont souvent assimilés à des boites noires, l'accès à leur code étant difficile. Ce type de matériel propriétaire renferme d'ailleurs souvent un système de protection permettant d'authentifier le logiciel associé (*par signature RSA par exemple*), et ainsi rendre toute modification pratiquement impossible. Ils ont pour ***avantages*** **:** Ils sont facilement intégrables au réseau ; leur administration est souvent simplifiée et leur niveau de sécurité est assez élevé. Ils présentent comme **Inconvénients :** Ce type de firewall étant propriétaire, les mises à jour dépendent entièrement du constructeur et en raison de l'architecture hardware, peu de modifications sont autorisées.

- **Les firewalls logiciels** - Ces pare-feu existent autant sous forme commerciales que sous forme gratuites. Quelque soit leur origine, la sécurité pourra fortement varier. Un logiciel commercial pourra parfois mettre en avant sa facilité de mise en place et de configuration, mais ce sera souvent aux dépends de la sécurité. Au niveau des logiciels gratuits et/ou libres, ils seront souvent plus flexibles (*c'est-à-dire plus fournis en options*), mais nécessiteront la plupart du temps de bonnes connaissances en réseau afin de les configurer finement sans abaisser le niveau de sécurité.

Les pare-feux sont un des plus vieux équipements de sécurité informatique et, en tant que tel, ont subit à de nombreuses évolutions. Suivant la génération du pare-feu ou son rôle précis, on peut les classer en 6 différentes catégories :

1. **Pare-feu sans état (*stateless firewall*)** - C'est le plus vieux dispositif de filtrage réseau, introduit sur les routeurs. Il regarde chaque paquet indépendamment des autres et le compare à une liste de règles préconfigurées. Ces règles peuvent avoir des noms très différents en fonction du pare-feu : « ACL » pour *Access Control List* (*certains pare-feux Cisco*), politique ou *policy* (*pare-feu Juniper/Netscreen*), filtres, règles ou *rules*, etc. … La configuration de ces dispositifs est souvent complexe et l'absence de prise en compte des machines à états des protocoles réseaux ne permet pas d'obtenir une finesse du filtrage très évoluée. Ces pare-feux ont donc tendance à tomber en désuétude mais restent présents sur certains routeurs ou systèmes d'exploitation.

2. **Pare-feu à états (*stateful firewall*)** - Certains protocoles dits « à états » comme TCP introduisent une notion de connexion. Les pare-feux à états vérifient la conformité des paquets à une connexion en cours. C'est-à-dire qu'ils vérifient que chaque paquet d'une connexion est bien la suite du précédent paquet et la réponse à un paquet dans l'autre sens. Ils savent aussi filtrer intelligemment les paquets ICMP qui servent à la signalisation des flux IP. Enfin, si les ACL autorisent un paquet UDP caractérisé par un quadruplet (*ip_src, port_src, ip_dst, port_dst*) à passer, un tel pare-feu autorisera la réponse caractérisée par un quadruplet inversé, sans avoir à écrire une ACL inverse. Ceci est fondamental pour le bon fonctionnement de tous les protocoles fondés sur l'UDP, comme DNS par exemple. Ce mécanisme apporte en fiabilité puisqu'il est plus sélectif quant à la nature du trafic autorisé. Cependant dans le cas d'UDP, cette caractéristique peut être utilisée pour établir des connexions directes (P2P) entre deux machines (*comme le fait Skype par exemple*).

3. **Pare-feu applicatif** - Dernière génération de pare-feu, ils vérifient la complète conformité du paquet à un protocole attendu. Par exemple, ce type de pare-feu permet de vérifier que seul le protocole HTTP passe par le port TCP 80. Ce traitement est très gourmand en temps de calcul dès que le débit devient très important. Il est justifié par le fait que de plus en plus de protocoles réseaux utilisent un tunnel TCP afin de contourner le filtrage par ports. Une autre raison de l'inspection applicative est l'ouverture de ports dynamique. Certains protocoles comme FTP, en mode passif, échangent entre le client et le serveur des adresses IP ou des ports TCP/UDP. Ces protocoles sont dits « *à contenu sale* » ou « passant difficilement les pare-feux » car ils échangent au niveau applicatif (FTP) des informations du niveau IP (*échange d'adresses*) ou du niveau TCP (*échange de ports*). Ce qui transgresse le principe de la séparation des couches réseaux. Pour cette raison, les protocoles « à contenu sale » passent difficilement voire pas du tout les règles de NAT ...dynamiques, à moins qu'une inspection applicative ne soit faite sur ce protocole.

Chaque type de pare-feu sait inspecter un nombre limité d'applications. Chaque application est gérée par un module différent pour pouvoir les activer ou les désactiver. La terminologie pour le concept de module est différente pour chaque type de pare-feu : par exemple : Le protocole HTTP permet d'accéder en lecture sur un serveur par une commande GET, et en écriture par une commande PUT. Un pare-feu applicatif va être en mesure d'analyser une connexion HTTP et de n'autoriser les commandes PUT qu'à un nombre restreint de machines.

4. **Pare-feu identifiant** - Un pare-feu réalise l'identification des connexions passant à travers le filtre IP. L'administrateur peut ainsi définir les règles de filtrage par utilisateur et non plus par adresse IP ou adresse MAC, et ainsi suivre l'activité réseau par utilisateur. Plusieurs méthodes différentes existent qui reposent sur des associations entre IP et utilisateurs réalisées par des moyens variés. On peut par exemple citer authpf (*sous OpenBSD*) qui utilise « *ssh »* pour faire l'association. Une autre méthode est l'identification connexion par connexion (sans avoir cette association IP = utilisateur et donc sans compromis sur la sécurité), réalisée par exemple par la suite NuFW, qui permet d'identifier également sur des machines multi-utilisateurs. On pourra également citer Cyberoam qui fournit un pare-feu entièrement basé sur l'identité (*en réalité en réalisant des associations adresse MAC = utilisateur*) ou Check Point avec l'option NAC Blade qui permet de créer des règles dynamiques basée sur l'authentification « *Kerberos* » d'un utilisateur, l'identité de son poste ainsi que son niveau de sécurité (présence d'antivirus, de patchs particuliers).

5. **Pare-feu personnel** - Les pare-feux personnels, généralement installés sur une machine de travail, agissent comme un pare-feu à états. Bien souvent, ils vérifient aussi quel programme est à l'origine des données. Le but est de lutter contre les virus informatiques et les logiciels espions.

6. **Pare-feu Portail captif** - Les portails captifs sont des pare-feux dont le but est d'intercepter les usagers d'un réseau de consultation afin de leur présenter une page web spéciale (par exemple : avertissement, charte d'utilisation, demande d'authentification, etc.) avant de les laisser accéder à Internet. Ils sont utilisés pour assurer la traçabilité des connexions et/ou limiter l'utilisation abusive des moyens d'accès. On les déploie essentiellement dans le cadre de réseaux de consultation Internet mutualisés filaires ou Wifi.

# QUATRIEME CHAPITRE - ADMINISTRATION D'ACCES AUX DONNEES INFORMATIQUES

Dans ce chapitre, nous allons présenter d'une manière succincte les différentes techniques d'accès aux données dans un système informatique en général et dans un système d'exploitation en particulier. Étant donné que l'accès à un système exige une authentification, nous allons exposer les caractéristiques et la problématique d'un mot de passe.

## IV.1. MODÈLE DE LAMPSON

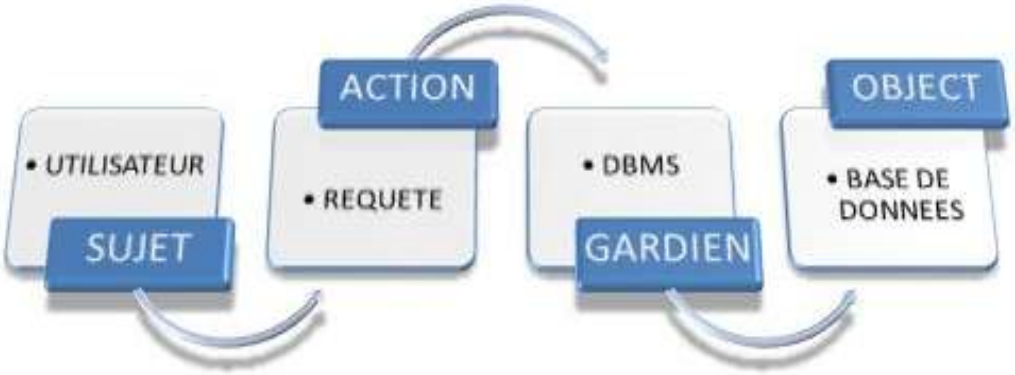
C'est en 2001 que Butler W. Lampson a présenté un modèle mettant en évidence les relations entre les différentes entités réelles d'un système d'accès aux données. Avant de voir un peu plus en détails les trois méthodes d'accès principales existantes, nous allons donc présenter son modèle. Lampson a défini les entités de son modèle en ceci :

- **Sujet** - Entité pouvant effectuer des actions (humain, processus, machine,. . .) ;
- **Action** - Opération effectuée par le sujet afin d'accéder à l'objet ;
- **Gardien** - Entité contrôlant l'accès proprement dit.
- **Objet** - Entité considérée comme ressource nécessitant un contrôle d'accès (fichier, répertoire, port,. . .) ;


*Modèle De B.W. Lampson*

L'exemple ci-après illustre, en pratique, le modèle de Lampson :

| SUJET | ACTION | GARDIEN | OBJET |
|---|---|---|---|
| UTILISATEUR | REQUETE | DBMS | BASE DE DONNEES |
| UTILISATEUR | AFFICHER UNE PAGE WEB | SERVEUR WEB | PAGE WEB |
| MACHINE | ENVOI DE PAQUET | FIREWALL | INTRANET |
| PROGRAMME | OUVRIR UN FICHIER | SECURITY MANAGEMENT JAVA | FICHIER |




Le modèle de Lampson est particulièrement adapté à la présentation pratique du protocole AAA (**Authentification**, **Autorisation** et **Audit**). En effet, par l'intermédiaire de son schéma, on représente facilement les endroits où ont lieu les différentes phases dudit protocole.

En revanche, la vérification de l'identité des parties communicantes est en réalité la base de tous les systèmes existants. Il fut complété au fil des années, mais reste historiquement la première modélisation de la sécurité d'un système. Le contrôle d'accès se fait en 4 étapes :

- **Identification** : Qui êtes-vous ?
- **Authentification** : Prouvez-le !
- **Autorisation** : Avez-vous les droits requis ?
- **Accounting/Audit** : Qu'avez-vous fait ?

Nous parlons du protocole AAA (*les deux premières étapes sont fusionnées*). Dans certaines situations, nous scinderons la dernière étape. Nous parlerons *d'Accounting* lorsque le fait de comptabiliser des faits sera demandé, et d'*Audit* lorsque des résultats plus globaux devront être étudiés. Notons également que l'authentification, visant à prouver l'identité d'un individu peut se faire de plusieurs manières :

- Ce que vous savez (mot de passe, code PIN, etc.) ;
- Ce que vous avez (carte magnétique, lecteur de carte, etc.) ;
- Ce que vous êtes (empreintes digitales, réseau rétinien, etc.).

L'authentification forte résultera de la combinaison de deux de ces facteurs.

# IV.2. MÉTHODES D'ACCÈS AUX DONNÉES

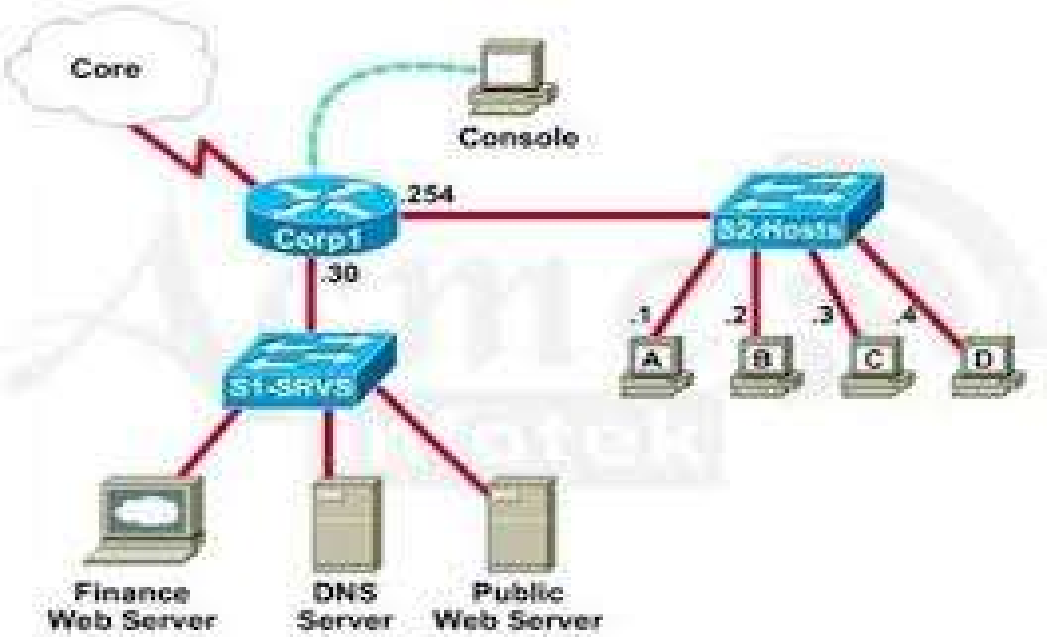
Quatre techniques principales existent. De nombreuses méthodes en sont dérivées mais ne seront qu'en partie évoquées, toutes pouvant être rapprochées de l'une ou l'autre des quatre méthodes évoquées ci-dessous :

## IV.2.1. Access List Control (ACL)

L'accès est déterminé par le propriétaire de l'objet. Celui-ci détermine qui (*sujet*) peut utiliser l'objet, et comment (*action*) il peut l'utiliser. On part du principe que tous les objets ont un propriétaire (*qui sera souvent le sujet qui aura créé cet objet*).

Quelques variantes de ce type d'accès permettent un transfert de propriété ou la délégation d'un droit entre utilisateurs non propriétaires. Cette méthode d'accès est celle rencontrée la plupart du temps dans les systèmes d'exploitation courants (*Linux, Windows).* Le principe est d'associer une liste de contrôle d'accès à chaque fichier. Cette ACL renferme un certain nombre d'entrées (*Access Control Entry - ACE*). Le détail des accès permis ou refusé est mentionné. Exemple d'ACL liée un objet T :

- ACE1 : L'utilisateur A peut le lire ;
- ACE2 : L'utilisateur B ne peut pas le modifier ;
- ACE3: . . .

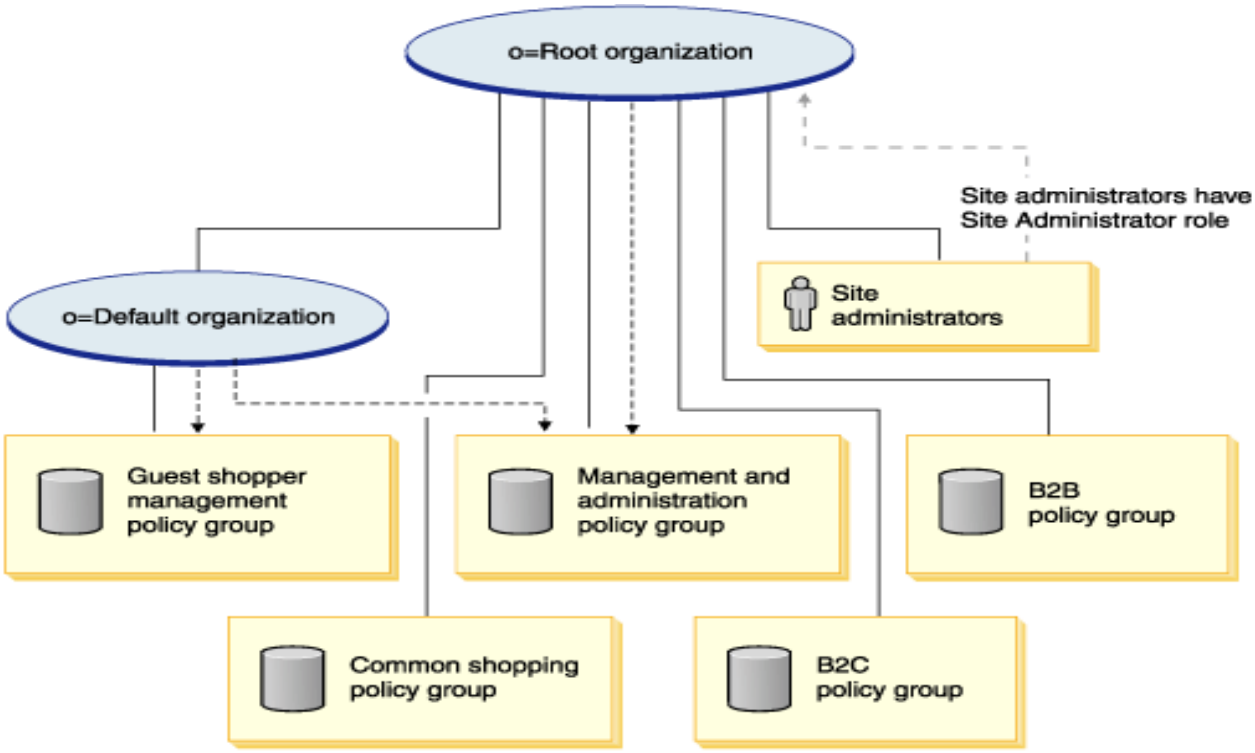


## IV.2.2. Mandatory Access Control (MAC)

Il s'agit ici de contrôler l'accès en se concentrant sur les flux de données. L'accès est déterminé par le système :

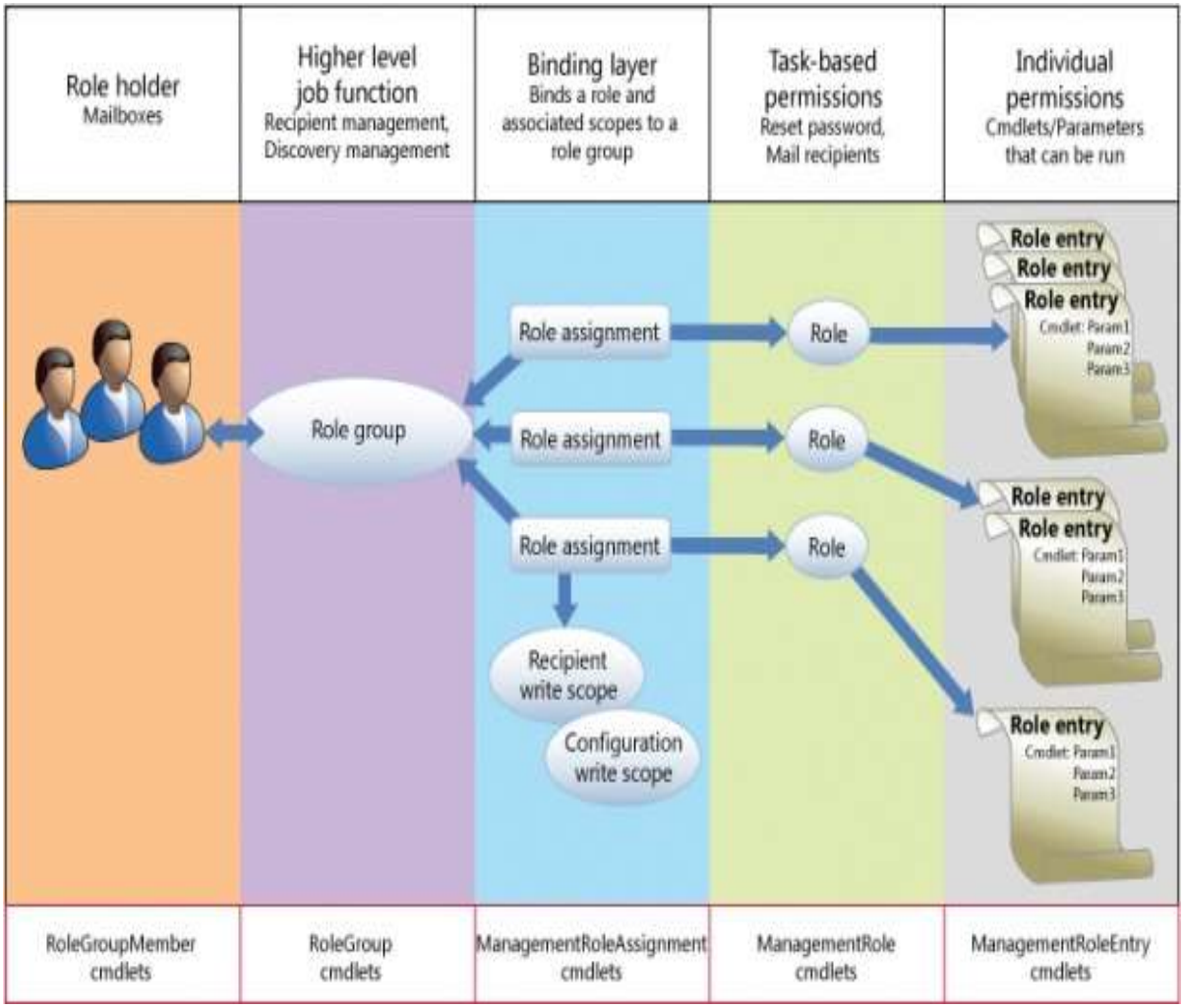
- Chaque sujet possède un label lui donnant un niveau de confiance ;
- Chaque objet possède un label permettant d'identifier le niveau de confiance requis pour l'utiliser ;
- Le sujet doit avoir un label supérieur ou égal à celui de l'objet.

Bref, cette méthode est beaucoup plus sûre, car elle ne dépend plus du propriétaire. Imaginez les pertes potentielles si le propriétaire d'un objet "sensible" venait à donner l'accès (volontairement ou non) à tous les utilisateurs. . . Un exemple concret est le modèle dérivé LBAC ( *Lattice-Based Access Control*). Une *lattice* est une règle définissant un ensemble de niveaux de sécurité. Par exemple, la définition suivante, illustrée à la figure ci-dessous :




- Les sujets avec un label L ne peuvent lire que des objets de label L0 <= L (*No Read-Up*) ;
- Les sujets avec un label L ne peuvent écrire que des objets de label L0 >= L (*No Write-Down*).

Comme l'illustre l'image ci-dessous, on voit que ce système est très contraignant. Peu adapté à un environnement commercial, il est peu répandu dans les contextes non-militaires. On le réserve à des *lattices* simples. Bien que très sûr, il reste sensible aux attaques par canaux cachés (*dialogue interprocessus à partir de l'utilisation du processeur par exemple*).

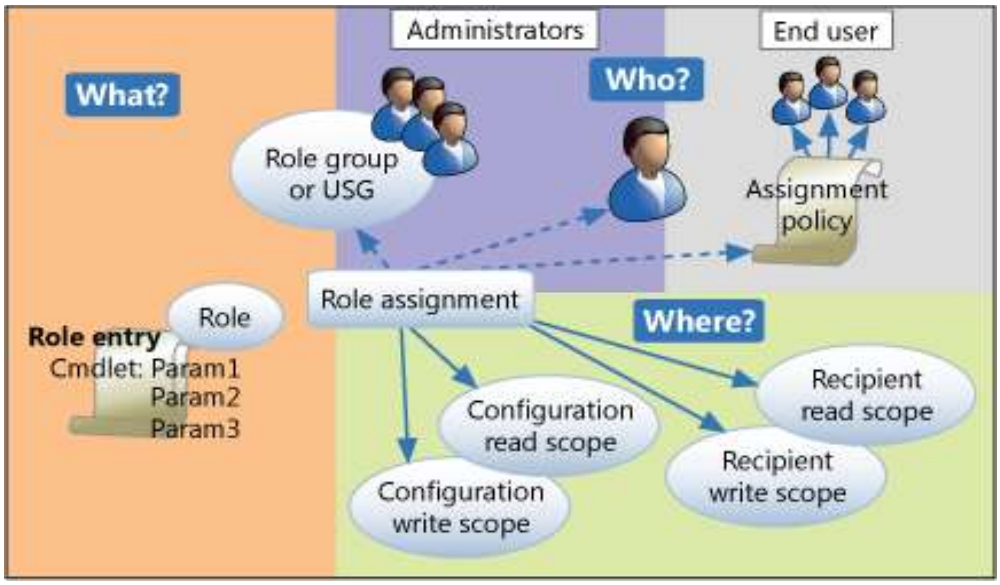



Dans le contexte grand public, ce type de contrôle d'accès est rarement utilisé pour garantir la confidentialité en tant que telle. Ainsi, les distributions Linux l'utilisent pour la vérification de l'intégrité des données. Son modèle porte le nom de "Biba Model", qui utilise une règle "*no write-up*". Cette technique est notamment utilise pour empêcher la modification des données de l'OS à partir de logiciels indésirables circulant sur Internet.

## IV.2.3. Role-Based Access Control (RBAC)

L'accès est ici aussi déterminé par le système. Cette méthode d'accès est régulièrement utilisée en entreprise, ou chaque personne possède un rôle particulier qui lui donne accès à certaines informations. Le rôle va permettre d'attribuer un ensemble de permissions à un type d'utilisateurs :

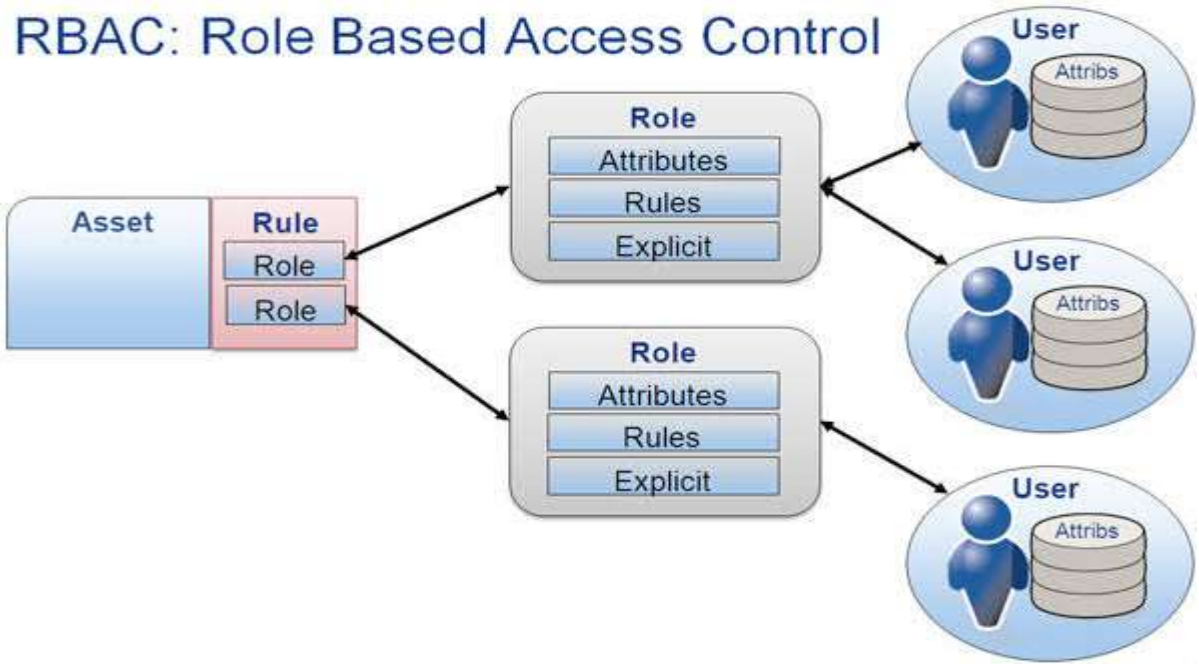
- Plusieurs utilisateurs peuvent avoir le même rôle ;
- Un utilisateur peut avoir plusieurs rôles, qu'il pourra activer au besoin.




Cette méthode est plus facile à mettre en œuvre que le DAC pour la simple raison qu'il suffit de changer de rôle lorsque cela s'avère nécessaire (*au lieu de modifier les accès pour chaque fichie*r). Il existe plusieurs variantes à cette méthode :

- Utilisation de rôles « hiérarchiques » : un rôle « *hiérarchiquement supérieur* » possède toutes les permissions des rôles « *inférieurs* » ;
- Contrôle des activations simultanées des rôles : plusieurs rôles peuvent selon les cas être autorisés simultanément pour un même utilisateur.

Ce type d'accès aux données apparait très proche de ce que nous connaissons sous Windows notamment, par l'intermédiaire de la gestion des comptes utilisateurs. Cette filiation n'est pas anodine puisqu'il est en effet possible de simuler un système RBAC en utilisant la notion de « *groupes utilisateurs* ». En réalité, il s'agit d'un DAC caché.




Windows utilise des ACL spécifiant les accès par groupe :

- ACE1 : Le groupe A ne peut pas lire le fichier ;
- ACE2 : Le groupe B peut modifier le fichier, ...

Avec la figure ci-haut, on voit que:

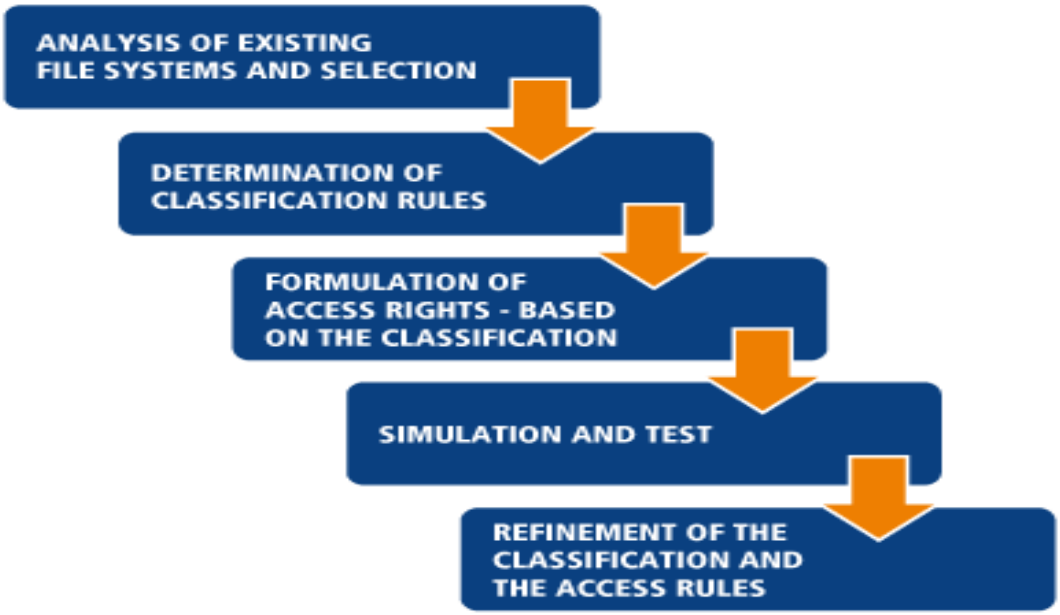
- A l'ouverture d'une session, l'utilisateur reçoit un jeton d'accès (*Access Token*). Ce jeton définira les actions qu'il peut effectuer.

SID - Security Identifier (unique) :

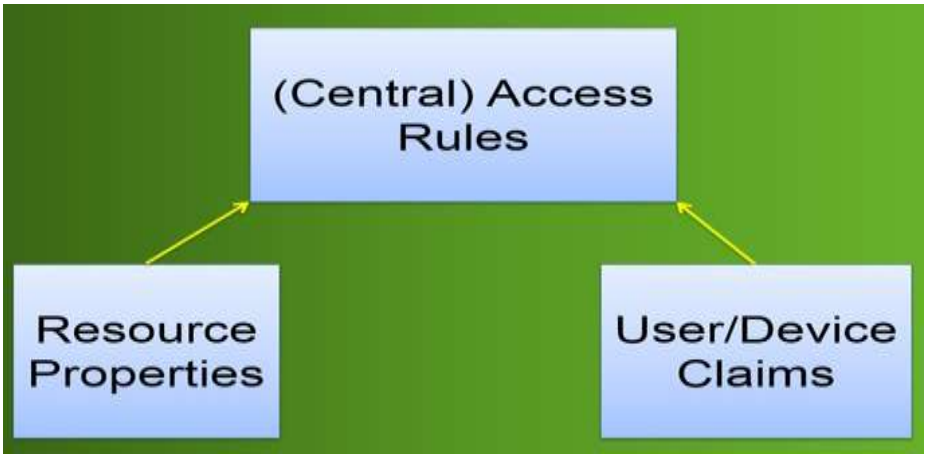
- SACL : System ACL utilisé pour l'Audit ;
- L'ordre établi dans la DACL est important : la première entrée correspondante est prise en compte ;
- Comme on le voit, la notion de groupe est présente, mais ce sont bel et bien les droits définis par le propriétaire qui détermine les actions possibles pour les autres utilisateurs.

## IV.2.4. Dynamic Access Control (DAC)

Le DAC est une solution de gouvernance de données, un ensemble de fonctionnalités qui permet aux entreprises d'organiser, de gérer, de distribuer et de sécuriser les dossiers et les fichiers au sein de l'infrastructure. L'enjeu principal du DAC est de pouvoir gérer le cycle de vie et l'intégrité de l'information au sein d'une organisation et de répondre finement aux besoins métier en termes d'autorisation d'accès, ce que ne permettent pas toujours les permissions NTFS afin d'assurer la sécurité des données.




Sur ce, le DAC fournit un contrôle d'accès non plus seulement basé sur des autorisations NTFS, mais également basé sur des expressions qui peuvent inclure des groupes de sécurités, des revendications, c'est-à-dire des propriétés propres à l'utilisateur, par exemple, ou à la machine qu'il utilise, des propriétés liées à la ressource, des propriétés de classification, celles que l'on retrouve plus classiquement dans le gestionnaire de ressources du serveur de fichiers de Windows. L'image ci-après illustre comment fonctionne le DAC.




En sus, le DAC utilise une surcouche des listes de contrôle d'accès NTFS, on est bien au-dessus de la couche ACE, Access Control Entry, dans ma liste ACL, ensuite le DAC va donc récupérer et utiliser des stratégies d'accès centralisé qui sont stockées dans l'annuaire Active Directory; ce qui permet la mise en place des expressions conditionnelles, qui en fonction de ses expressions vont accorder ou pas l'accès à un utilisateur sur une ressource.

# IV.3. NOTIONS SUR LE MOT DE PASSE

## IV.3.1. DEFINITION

Un *mot de passe* est un mot ou une série de caractères utilisés comme moyen *d'authentification* pour prouver son identité lorsque l'on désire accéder à un lieu protégé, à une ressource (*notamment informatique*) ou à un service dont l'accès est limité et protégé. Le mot de passe doit être tenu secret pour éviter qu'un tiers non autorisé puisse accéder à la ressource ou au service. C'est une méthode parmi d'autres pour vérifier qu'une personne correspond bien à l'identité déclarée. Il s'agit d'une preuve que l'on possède et que l'on communique au service chargé d'autoriser l'accès.
C'est par exemple : Dans le conte « *Ali Baba et les Quarante Voleurs des Mille et Une Nuits »* figure l'un des plus célèbres mots de passe : « *Sésame, ouvre-toi !* ».

Le terme « mot de passe » est d'origine militaire. Les « mots d'ordres » comprennent le « mot de sommation » (*c'est-à-dire la question convenue*) et le « mot de passe » (*c'est-à-dire la réponse correspondante*)[14]. Il s'agit des signes verbaux qui permettent à deux unités ou deux militaires de se reconnaître mutuellement, par exemple lors d'une patrouille de nuit, au moment délicat du retour dans le dispositif ami. Dans ce cas, le mot de passe est donc *partagé* par un groupe de personnes de confiance. Quand il s'agit d'un code personnel, il vaut mieux utiliser l'expression « code confidentiel » pour mettre en évidence le caractère secret du code et responsabiliser son détenteur.

## IV.3.2. PRINCIPE ET LIMITES

Une limite juridique existe, par exemple en France à la sécurisation par mot de passe : si des données chiffrées sont saisies par la justice, la loi sur la sécurité quotidienne oblige l'utilisateur à fournir la méthode de chiffrement et les clés ou mots de passe. L'utilisation de mots de passe est un compromis entre la sécurité et l'aspect pratique. Avec la multiplication de situations où il est nécessaire de disposer d'un mot de passe, chacun de nous possède un nombre de plus en plus important de mots de passe. S'il est légitime d'utiliser le même mot de passe pour l'ensemble des situations où celui-ci n'a pas grande importance, il reste néanmoins de nombreux cas où un mot de passe de qualité doit être utilisé. Ce mot de passe ne peut être le même partout, d'une part pour éviter que la compromission de celui-ci ne conduise à des situations malheureuses, d'autre part parce que certains sites et logiciels obligent à changer régulièrement son mot de passe et en limitent la réutilisation. La mémoire de l'utilisateur étant alors insuffisante pour mémoriser tous ces mots de passe, la tentation est grande de les lister. Il est indispensable que la liste de mots de passe ainsi constituée soit plus protégée encore. On parle alors de « *coffre-fort à mots de passe* » :

[14] « Quelle est l'origine du mot de passe ? », 23 mars 2015

- **Capture d'un mot de passe « en clair »** - Un mot de passe est « en clair » lorsqu'il n'a pas été transformé via une fonction de hachage. Il existe plusieurs situations où le mot de passe peut être trouvé en clair :

  → Espionnage direct du clavier de la personne qui saisit son mot de passe ;

  → Mise en place d'un enregistreur de frappes (keylogger), qui saisit tout texte tapé par un utilisateur à son insu ;

  → Écoute du réseau. Si un attaquant arrive à écouter une communication non chiffrée où la cible doit s'identifier par un mot de passe, ce mot de passe apparaîtra en clair à l'attaquant.

  → Vol d'un mot de passe manuscrit.

Certains logiciels permettent de rendre visibles les mots de passe des formulaires. Les caractères sont « cachés » par des ronds ou astérisques, qui sont là pour éviter qu'une personne derrière soi ne lise ce que l'on saisit. Dans le programme, à ce moment-là, le mot de passe est bien présent et encore non chiffré, le rendre visible consiste simplement à changer une option d'affichage.

- **Capture d'un mot de passe chiffré** - Dans le cas où un mot de passe chiffré est capturé, il n'est pas directement utilisable : la personne malintentionnée (l'attaquant) doit découvrir le clair correspondant, en le déchiffrant si c'est possible, ou avec d'autres techniques. On dit que l'attaquant casse ou « craque » le mot de passe. On distingue deux principaux cas de figure : le mot de passe fait partie d'une communication, ou c'est seulement le mot de passe chiffré qui est capturé :

  → *Toute la communication est chiffrée* : Dans ce cas, il faut trouver un moyen de déchiffrer toute la communication pour trouver le mot de passe. Il faut donc trouver une faille dans l'algorithme de chiffrement ou dans une de ses implémentations. Si le chiffrement est cassé, peu importe la taille du mot de passe il sera trouvé dans le texte déchiffré.

  → *Seul le mot de passe chiffré est capturé* : C'est généralement un condensat (ou *hash*) du mot de passe qui est capturé, c'est-à-dire le résultat d'un algorithme non réversible. C'est une bonne pratique, utilisée dans de nombreux cas : sites web, comptes d'utilisateur de système d'exploitation, etc. Dans le cas où cet algorithme n'est pas vraiment irréversible (*à cause d'erreurs de conception ou d'implémentation*), il peut être possible de retrouver le clair correspondant à un condensat. Par exemple, la gestion des mots de passe pour protéger les fichiers Excel et Word d'Office 97 comporte des failles qui font qu'il est facile de trouver les mots de passe utilisés.

Mais en général pour casser un condensat, on utilise d'autres techniques. En connaissant la fonction de hachage, on peut imaginer différentes attaques[15] :

- ✓ *l'attaque par force brute* : on se donne un espace de mots de passe à explorer en se fixant une longueur et un ensemble de caractères ; on énumère tous les mots de passe possibles de cet espace ; pour chacun de ces mots de passe on calcule l'empreinte par la fonction de hachage, et on compare cette empreinte avec celle que l'on a capturée. Pour empêcher ces attaques, l'utilisation d'un mot de passe long et complexe est recommandée. Par mot de passe complexe, on entend mot de passe comprenant différents types de caractères : des lettres minuscules et majuscules, des chiffres, et des caractères non alphanumériques (*comme !:/#@ ...*). La longueur du mot de passe assurera qu'il n'est pas énuméré lors d'une attaque par force brute : plus l'espace à énumérer est grand et plus l'attaque ne prend de temps. Voir le graphique ci-contre (*attention l'échelle est logarithmique*).

- ✓ *l'attaque par dictionnaire* : même chose que pour l'attaque par force brute, mais où les mots sont choisis dans un dictionnaire. L'utilisation de caractères différents des lettres et chiffres assurera généralement que le mot de passe n'appartient pas à un dictionnaire, et qu'il ne sera donc pas sensible à une attaque par dictionnaire.

## IV.3.3. CHOIX DU MOT DE PASSE

S'il y a bien un domaine où la sécurité peut faire défaut, c'est dans la gestion des mots de passe utilisateur. En effet, la sécurité d'un système dépend aussi du niveau de sécurité du mot de passe mis sur pied pour son accès. Lors de la création du mot de passe, une attention particulière doit être portée sur certains points :

- Si un générateur de mots de passe est utilisé, il devra employer une grande variété de caractères (*pour le rendre plus robuste à la force brute*) ;
- Il peut être utile d'utiliser un vérificateur de mots de passe afin de tester la vulnérabilité aux attaques par dictionnaire. Dans le même temps, il pourra tester la taille des mots de passe face aux attaques par force brute ;
- Il est bon d'associer une durée de vie aux mots de passe. Un changement régulier permet une meilleure protection contre la force brute.
- On peut aussi limiter le nombre d'essais.

[15] Des attaques plus complexes, issues de la cryptanalyse, comme l'utilisation de tables arc-en-ciel. L'utilisation d'un mot de passe complexe ne protège pas nécessairement de ce type d'attaque. La préparation d'une attaque de ce type est longue et gourmande en espace de stockage, mais elle est loin d'être inaccessible de nos jours.

## A. CRITÈRES DE ROBUSTESSE

La *qualité* et la *longueur du mot de passe* sont des éléments cruciaux pour la sécurité. Un mot de passe trop court ou provenant d'un dictionnaire est susceptible d'être attaqué *via* une recherche dans une table contenant une liste de mots de passe. D'une manière plus systématique, une attaque par force brute tente toutes les possibilités et, avec suffisamment de temps, il est théoriquement possible de retrouver le mot de passe. Un compromis est la table arc-en-ciel, une amélioration du principe du compromis temps-mémoire. La robustesse d'un mot de passe dépend de plusieurs critères :

- **Sa longueur -** qui est le critère le plus important. Il est conseillé d'utiliser des mots de passe d'une longueur suffisante pour que celui-ci soit protégé des attaques de force brute (*cette longueur augmente au fil du temps avec la puissance des outils utilisés par les attaquants - pour un mot de passe à usage local, on recommande dans les années 2010 au moins 12 caractères, voire 16 caractères pour des mots de passe plus sûrs*).

- **Sa non-simplicité** - 123456, www, 111111, Love, 0000, azerty… sont à proscrire, de même que les dates de naissance, le nom du chien ou toutes autres informations ayant un rapport direct avec la vie privée. De même, les slogans et les citations sont facilement attaquables via une attaque par dictionnaire. Plus généralement, le contenu du mot de passe ne devrait suivre aucune logique, mais être une simple succession de caractères choisis aléatoirement.

- **Son unicité** - la réutilisation du même mot de passe pour des services différents est à proscrire, afin d'éviter des dégâts en cascade.

- **La variation des caractères utilisés-** le mot de passe est plus fort lorsqu'il mélange des majuscules, des minuscules, des chiffres, des signes de ponctuation et des caractères spéciaux. Notons par ailleurs qu'il est toujours plus sécurisé de chercher à augmenter la longueur d'un mot de passe que de chercher à utiliser le plus possible des caractères différents.

Par ailleurs, le choix d'un mot de passe doit se faire suivant la criticité de ce dernier (par exemple, un mot de passe permettant d'accéder à l'interface d'administration d'une application ou d'un équipement sera considéré comme étant très critique).

Dans la pratique, une étude portant sur 32 millions de mots de passe du site RockYou.com, obtenus en 2009 à la suite d'une attaque du site, a montré que 30 % de ces mots de passe comportaient six caractères ou moins, et que le plus fréquent (*un peu moins d'un sur cent*) est « **123456** »[16]. Tous les ans, *Splash-Data*, fournisseur de solutions de sécurité publie également une liste des mots de passe les plus utilisés, et désignés comme étant « les pires mots de passe, à ne pas utiliser ». Les 5 mots de passe les plus utilisés par les utilisateurs du monde entier en 2015 sont[17] :

- 123456 ;
- Password ;
- 12345678 ;
- Qwerty.

## B. LES "MAUVAIS" MOTS DE PASSE

- **Par défaut** : password, mdp, default, admin, . . .
- **Mots** : bonjour, test, voiture, silence, . . .
- **Mots numérotés** : clavier12, merci154, armoire98, . . .
- **Egaux au login** : Albert84, . . .
- **Mots doubles** : crabcrab, stopstop, treetree,...
- **Séquences** : qwerty, 12345678, bhunji,...
- **Personnels** : prénom, numéro d'imatriculation,...
- **Noms propres** : Anaël, Yende ,...

Certes, les besoins en sécurité étant ce qu'ils sont, la difficulté majeure aujourd'hui est que le nombre de mots de passe ne cesse de croître. Leur mémorisation reste donc problématique. Toutefois, quelques solutions existent :

- **Hardware -** Utiliser des clés usb comme accès aux données. Le problème est que c'est alors la clé qui authentifie, et non l'individu. De plus, que faire en cas de perte de la clé ?

- **Software -** utiliser un logiciel de gestion de mots de passe : un seul mot de passe (ou une phrase de passe) pour stocker tous les autres. Mais que faire en cas d'oubli du mot de passe maître ?

[16] Guillaume Belfiore, « Étude : les 20 mots de passe les plus populaires sont....», *Clubic*, 22 janvier 2010
[17] Worst Passwords of 2015 [archive], sur teamsid.com

- **utiliser la saisie semi-automatique** ;
- **Au niveau de l'OS -** Une solution avancée est connue sous le nom de SSO (Single Sign On). Le fait de se « *logguer* » sur une machine permet d'accéder à toutes les données. Une seule phase d'authentification a donc lieu, et si elle réussi, l'utilisateur est libre d'agir avec les données et logiciels correspondant à ses droits, sans avoir à donner son mot de passe à chaque accès. Une application connue basée sur un principe similaire porte le nom de Kerberos (*authentification d'utilisateurs sur les machines d'un réseau*).

## C. LES AUTRES CATEGORIES DE MOTS DE PASSE

Il existe une autre possibilité, mais qui n'est pas utilisable dans toutes les conditions, porte le nom *de One Time Password (OTP)* et *Mot de passe sous contrainte* :

- **Mot de passe sous contrainte -** Certaines sociétés de télésurveillance utilisent 2 mots de passe : le mot de passe normal et le mot de passe sous contrainte. En cas de déclenchement de l'alarme, la société de télésurveillance appelle le client, et demande son mot de passe pour s'assurer de son identité, si celui-ci donne le mot de passe sous contrainte, alors l'agent de télésurveillance sait que la personne qui le dit est menacée et déclenche une intervention.
- ***One Time Password* -** Le mot de passe ici généré ne reste valable que pour une durée déterminée. Deux méthodes coexistent :
  - ✓ **Méthode synchrone** : Fonction du temps (*Time-synchronous*) ;
  - ✓ **Méthode asynchrone** : En réponse à un "challenge" (*Challenge-response*).

Signalons qu'un bon mot de passe doit notamment (et principalement) être à la fois facile à retenir (pas besoin de le noter), et utiliser des caractères spéciaux, de casse différente (ce qui rend la force brute plus fastidieuse et l'attaque par dictionnaire presque impossible). Les conseils proposés par la Commission européenne devraient être appliqués lorsque l'on veut créer un mot de passe afin de le protéger contre les attaques. En fait, un bon mot de passe doit: ***Etre long ; Etre unique ; Etre complexe ; Modifié régulièrement ; Pas contenir n'importe quelle partie du nom du compte utilisateur ; Avoir un minimum de huit caractères ; Contenir des caractères d'au moins trois des catégories suivantes ; Symboles non alphanumériques ($,:;"%@#!); Chiffres; Lettres majuscules et Lettres minuscules***.

# CINQUIEME CHAPITRE – INTRODUCTION A LA CRYPTOLOGIE

## V.1. INTRODUCTION

La cryptologie : Voilà bien une science dont tout le monde connaît le nom sans vraiment savoir ce que c'est. Des mots sont lancés, tel « *agent secret* » , ou « *guerre* ». Une petite partie des gens interrogés considèrent même que la cryptologie n'est plus d'actualité à cause des ordinateurs ; Ce qui est complètement faux d'ailleurs, les ordinateurs ayant accentué la démocratisation de la cryptologie , qui n'était auparavant réservée qu'aux milieux militaires. Mais pour simplifier, il faudrait tout simplement retenir que la cryptologie est l'ancetre des principes utilisees par l'informatique, principalement en transmission numerique ; ce fut imputé à **Shannon** qui, en 1948 puis en 1949 avec **Weaver**, qui, le premier à définir les bases d'une transmission de données entre deux parties.


Résumons tout cela, en disant que les signaux émis et reçu seront ici l'envoi et la réception de messages, mais nous préciserons la terminologie employée par la suite. Avec la popularité grandissante des réseaux, des échanges de données (*des transmissions entre individus*), de nombreuses menaces sont nées. Parmi celles-ci, on trouve diverses catégories :

- **Les menaces accidentelles** - ne supposent aucune préméditation. Dans cette catégorie, sont repris les bugs logiciels, les pannes matérielles, et autres défaillances incontrôlables.

- **Les menaces intentionnelles passives et actives** - quant à elles, reposent sur l'action d'un tiers désirant s'introduire et dérober des informations. Dans le cas d'une attaque passive, l'intrus va tenter de dérober les informations par audit, ce qui rend sa détection relativement difficile. En effet, cet audit ne modifie pas les fichiers, ni n'altère les systèmes. Dans le cas d'une attaque active, la détection est facilitée, mais il peut être déjà trop tard lorsque celle-ci a lieu. Ici, l'intrus aura volontairement modifié les fichiers ou le système en place pour s'en emparer.

Les menaces actives appartiennent principalement à quatre catégories :

- **Interruption**: problème lié à la disponibilité des données ;
- **Interception:** problème lié à la confidentialité des données **;**
- **Modification**: problème lié à l'intégrité des données ;
- **Fabrication** : problème lié à l'authenticité des données.

Les auteurs de ces attaques sont notamment les hackers (agissant souvent par défi personnel), les concurrents industriels (vol d'informations concernant la stratégie de l'entreprise ou la conception de projets), les espions, la presse ou encore les agences nationales. Nous verrons dans le chapitre cinq la manière dont se prennent les hackers pour prendre le contrôle des systèmes informatiques.

## V.2. DEFINITION ET TERMINOLOGIE

La **cryptologie**, étymologiquement la *science du secret*, ne peut être vraiment considérée comme une science que depuis peu de temps. Cette science englobe la *cryptographie,* l'écriture secrète et la *cryptanalyse*, l'analyse de cette dernière. La cryptologie est un art ancien et une science nouvelle : un art ancien car les Spartiates l'utilisaient déjà (*la scytale*) ; une science nouvelle parce que ce n'est un thème de recherche scientifique académique, c'est-à-dire universitaire, que depuis les années 1970. Cette discipline est liée à beaucoup d'autres, par exemple l'arithmétique modulaire, l'algèbre, la théorie de la complexité, la théorie de l'information, ou encore les codes correcteurs d'erreurs.

Les premières méthodes de chiffrement remontent à l'Antiquité et se sont améliorées, avec la fabrication de différentes machines de chiffrement, pour obtenir un rôle majeur lors de la Première Guerre mondiale et de la Seconde Guerre mondiale. Voici les quelques terminologies liées à la cryptologie :

- **La cryptologie -** est l'ensemble formé de la cryptographie et de la cryptanalyse. Elle est une science mathématique des messages secrets

- **La cryptographie** - est l'art de rendre inintelligible, de crypter, de coder, un message pour ceux qui ne sont pas habilités à en prendre connaissance. Du grec : caché et écrire, la cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle (chiffrée) sur un support donné.

- **Crypter** - synonyme de "chiffrer".

- **La cryptanalyse** - est l'art pour une personne non habilitée, de décrypter, de décoder, de déchiffrer, un message. C'est donc l'ensemble des procèdes d'attaques d'un système cryptographique. elle a pour but de retrouver le texte clair à partir de textes chiffrés en déterminant les failles des algorithmes utilisés.
- **Chiffrement** - Le chiffrement consiste à transformer une donnée (texte, message, ...) afin de la rendre incompréhensible par une personne autre que celui qui a créé le message et celui qui en est le destinataire. La fonction permettant de retrouver le texte clair à partir du texte chiffré porte le nom de **déchiffrement**.
- **Texte chiffré** - Appelé également cryptogramme, le texte chiffré est le résultat de l'application d'un chiffrement à un texte clair.
- **Clef** - Il s'agit du paramètre impliqué et autorisant des opérations de chiffrement et/ou déchiffrement. Dans le cas d'un algorithme symétrique, la clef est identique lors des deux opérations. Dans le cas d'algorithmes asymétriques, elle diffère pour les deux opérations.
- **Cryptosystèmes** - Il est défini comme l'ensemble des clés possibles (espace de clés), des textes clairs et chiffrés possibles associés à un algorithme donné.
- **Code** - Système de chiffrement dans lequel chaque lettre (ou mot, syllabes,...) est remplacé par un -ou plusieurs- symboles (caractères, dessins, ...) , par un processus d'opérations.(le plus souvent des tables de correspondances entre la lettre et son symbole). Les codes ,s'ils peuvent être secrets, ne le sont généralement pas. Parmi les plus connus , on peut citer le code morse et le code ASCII .
- **Chiffre** - nom donné à un code secret , c'est à dire soit un code dont le processus d'opérations (l'algorithme) est tenu secrète , soit un code dont l'algorithme est connu, mais dont la clef est secrète.
- **Cryptogramme** - message chiffré ( qui a été codé par un Chiffre)
- **Déchiffrer** - opération inverse du chiffrage : transformer un texte chiffré en un texte en clair en connaissant le procédé de secret utilisé. ( c'est à dire l'algorithme du chiffre et sa clé, s'il en a une)
- **Décrypter** - transformer un texte chiffré en un texte en clair, sans connaître le procédé de secret utilisé. C'est sur ce point que déchiffrage et décryptage s'opposent et ne sont ,de ce fait, pas synonymes.
- **cryptolecte** - jargon réservé à un groupe restreint de personnes désirant dissimuler leur communication.

# V.3. PRINCIPES DE BASE DE LA CRYPTOLOGIE (PRINCIPES DE KERCKHOFFS)

En 1883 dans un article paru dans le Journal des sciences militaires, *Auguste Kerckhoffs* (1835-1903) posa les principes de la cryptologie. Ces principes stipulent entre autre que : « **la sécurité d'un cryptosystèmes ne doit pas reposer sur le secret de l'algorithme de codage mais qu'elle doit uniquement reposer sur la clef secrète du cryptosystèmes qui est un paramètre facile à changer, de taille réduite (*actuellement de 64 à 2048 bits suivant le type de code et la sécurité demandée*) et donc assez facile à transmettre secrètement** ». Ce principe n'est que la transposition des remarques de bon sens suivantes :

- Un cryptosystèmes sera d'autant plus résistant et sûr qu'il aura été conçu, choisi et implémenté avec la plus grande transparence et soumis ainsi à l'analyse de l'ensemble de la communauté cryptographique.
- Si un algorithme est suppose être secret, il se trouvera toujours quelqu'un soit pour vendre l'algorithme, soit pour le percer à jour, soit pour en découvrir une faiblesse ignorée de ses concepteurs. A ce moment là, c'est tout le cryptosystèmes qui est à changer et pas seulement la clé. Les systèmes conçus dans le secret révèlent souvent rapidement des défauts de sécurité qui n'avaient pas été envisages par les concepteurs.

# V.4. QUALIT´ES D'UN CRYPTOSYST`EME

Les qualités demandées à un système cryptographique sont résumées par les mots clefs suivants :

- **Confidentialité** : seules les personnes habilitées ont accès au contenu du message.
- **Intégrité des données** : le message ne peut pas être falsifié sans qu'on s'en aperçoive.
- **Authentification:** d'abord, l'émetteur est sûr de l'identité du destinataire c'est-a-dire que seul le destinataire pourra prendre connaissance du message car il est le seul à disposer de la clef de déchiffrement. Ensuite, le receveur est sûr de l'identité de l'émetteur.
- **Non-répudiation** qui se décompose en trois :

→ *La non-répudiation d'origine* de l'émetteur ne peut nier avoir écrit le message et il peut prouver qu'il ne l'a pas fait si c'est effectivement le cas.

→ *La non-répudiation de réception* le receveur ne peut nier avoir reçu le message et il peut prouver qu'il ne l'a pas reçu si c'est effectivement le cas.

→ *La non-répudiation de transmission* l'émetteur du message ne peut nier avoir envoyé le message et il peut prouver qu'il ne l'a pas fait si c'est effectivement le cas.

Ainsi, nous pouvons regarder ces quatre qualités du point de vue de l'émetteur veut être certaine que :

- une personne non-autorisée ne peut pas prendre connaissance des messages qu'elle envoie, *confidentialité*.
- ses messages ne seront pas falsifiés par un attaquant malveillant, *intégrité*.
- le destinataire a bien pris connaissance de ses messages et ne pourra pas nier l'avoir reçu, *non-répudiation*.
- le message reçu vient bien l'émetteur, par exemple qu'un attaquant malveillant ne puisse pas se faire passer pour l'émetteur, *mascarade ou usurpation d'identité (authentification)*.

# V.5. INTRODUCTION A LA CRYPTOGRAPHIE

## V.3.1. DEFINITION

Le mot cryptographie vient des mots en grec ancien *kruptos* (κρυπτός) « *caché* » et *graphein* (γράφειν) « *écrire* ». Beaucoup des termes de la cryptographie utilisent la racine « crypt- », ou des dérivés du terme « chiffre ». La **cryptographie** est une des disciplines de la cryptologie s'attachant à protéger des messages (*assurant confidentialité, authenticité et intégrité*) en s'aidant souvent de *secrets* ou *clés*. Elle se distingue de la « *stéganographie* » qui fait passer inaperçu un message dans un autre message alors que la cryptographie rend un message inintelligible à autre que qui-de-droit. Elle est utilisée depuis l'Antiquité, mais certaines de ses méthodes les plus importantes, comme la cryptographie asymétrique, datent de la fin du XX$^{e}$ siècle.

La cryptographie, ou *art de chiffrer*, coder les messages, est devenue aujourd'hui une science à part entière. Au croisement des *mathématiques,* de ***l'****informatique,* et parfois même de la *physique,* elle permet ce dont les civilisations ont besoin depuis qu'elles existent : le maintien du secret. Pour éviter une guerre, protéger un peuple, il est parfois nécessaire de cacher des choses...

La cryptographie étant un sujet très vaste, ce chapitre se focalisera essentiellement sur les méthodes de chiffrement dites *classiques* et celles dites *modernes*, c'est-à-dire celles étant apparues et utilisées après la Seconde Guerre mondiale. On passera en revue la saga du *DES* et de ***l'****AES*, en passant par le fameux RSA, le protocole le plus utilisé de nos jours. Ayant longtemps été l'apanage des militaires et des sociétés possédant de gros moyens financiers, la cryptographie s'est au fil du temps ouverte au grand public, et est donc un sujet digne d'intérêt. Toutes les méthodes de cryptographie seront présentées dans leur ordre chronologique d'apparition.

## V.3.2. PRINCIPE DE FONCTIONNEMENT DE LA CRYPTOGRAPHIE

La cryptologie fait partie d'un ensemble de théories et de techniques liées à la transmission de l'information (théorie des ondes électromagnétiques, théorie du signal, théorie des codes correcteur d'erreurs, théorie de l'information, théorie de la complexité,...).

Un expéditeur *Georgine* veut envoyer un message à un destinataire *Blaise* en évitant les oreilles indiscrète de *Merveille*, et les attaques malveillantes de *Vianney*. Pour cela, *Georgine* se met d'accord avec *Blaise* sur le cryptosystème qu'ils vont utiliser. Ce choix n'a pas besoin d'être secret en vertu du principe de Kerckhoffs,

L'information que *Georgine* souhaite transmettre à *Blaise* est le texte clair. Le processus de transformation d'un message, *M*, pour qu'il devienne incompréhensible à *Merveille* est appelé le chiffrement ou le codage. On génère ainsi un message chiffré, *C*, obtenu grâce à une fonction de chiffrement, *E*.

$$C = E(M).$$

Le processus de reconstruction du message clair à partir du message chiffré est appelé le déchiffrement ou décodage et utilise une fonction de déchiffrement, *D*. On demande que pour tout message clair *M.*

$$D(C) = D(E(M)) = M$$

Autrement dit, on demande que tout message codé provienne d'un et d'un seul message clair (*D* est une fonction surjective des messages codés vers les messages clairs et *E* est une fonction injective des messages clairs sur les messages codés). Un algorithme cryptographique est l'ensemble des fonctions (*mathématiques ou non*) utilisées pour le chiffrement et le déchiffrement. En pratique les fonctions E et D sont paramétrées par des clés, *Ke* la clé de chiffrement et *Kd* la clef de déchiffrement, qui peuvent prendre l'une des valeurs d'un ensemble appelé espace des clefs. On a donc la relation suivante :

$$\begin{cases} E_{Ke}(M) = C \\ D_{Kd}(C) = M \end{cases}$$

Le type de relation qui unit les clés *Ke* et *Kd* permet de définir deux grandes catégories de systèmes cryptographiques :

- *Les systèmes à clef secrètes ou symétriques* : (DES, AES, IDEA, Blowfish,...) ;
- *Les systèmes à clefs publiques ou asymétriques* : (RSA, El-Gamal, un cryptosystème elliptique,...).

En outre les fonctions de codage E et de décodage D peuvent fonctionner de deux façons :

- *En continu (flot)* : chaque nouveau bit est manipulé directement ;
- *Par bloc* : chaque message est d'abord partitionné en blocs de longueur fixe. Les fonctions de chiffrement et déchiffrement agissent alors sur chaque bloc.

Chacun de ces systèmes dépend d'un ou deux paramètres de taille assez réduite (128 à 2048 bits) appelés la clef de chiffrement et la clef de déchiffrement. Les clefs de chiffrement et de déchiffrement n'ont aucune raison d'être identiques. Seule la clef de déchiffrement doit impérativement être secrète.

Ainsi, la cryptographie se subdivise en deux grandes familles :

- La cryptographie classique ou Manuelle ;
- La cryptographie moderne ou Automatique ou Informatique.

## V.3.3. LA CRYPTOGRAPHIE CLASSIQUE

La cryptographie classique, aussi appelée « *cryptographie manuelle* » peut être considérée comme une transformation des messages clairs en faisant appel à l'intervention active de l'homme (activité physique) en le rendant incompréhensible par une tierce personne lors d'une transmission d'un message entre deux correspondants. La cryptographie classique est subdivisée à son tour à deux parties :

- **La cryptographie mono-alphabétique** – c'est un chiffrement des messages clairs qui utilise une seule valeur non variante pour chaque lettre chiffrée ;
- **La cryptographie poly-alphabétique** – est celle qui utilise plusieurs valeurs variantes de chiffrement pour chacune des lettres du message clair.

### V.3.3.1. LA CRYPTOGRAPHIE MONO-ALPHABETIQUE

Dans cette catégorie, nous distinguerons trois grandes sous-catégories de cryptographie (chiffrement) :

- **La cryptographie à répertoire ;**
- **La cryptographie par Substitution** (aussi appelée *code de César* ou *chiffrement par décalage*) ;
- **La cryptographie par transposition** (aussi appelée *chiffrement par permutation*).

### A. LA CRYPTOGRAPHIE A REPERTOIRE

La cryptographie à répertoire consiste en un dictionnaire qui permet de remplacer certains mots par des mots différents. Ils sont tr`es anciens et ont été utilisés intensivement jusqu'au début du 20eme siècle. Ils ont fait l'objet d'une critique sévère de « A. Kerckhoffs » dans son article fondateur. On peut par exemple créer le dictionnaire suivant :

Recteur de FAB = Considérablement
Demain = Intellectuel
Rendez-vous = Le niveau
Avant-midi = Au Congo
Au bureau = Baisse

La phrase en clair :

« **RENDEZ VOUS DEMAIN AU BUREAU DU RECTEUR DE FAB AVANT-MIDI** ».

Devient avec ce code :

« **LE NIVEAU INTELLECTUEL BAISSE CONSIDERABLEMENT AU CONGO** ».

Il faut donc disposer de dictionnaires qui prévoient toutes les possibilités. Donc, sauf si on se restreint à transmettre des informations très limitées, la taille du dictionnaire s'accroit démesurément. Au 19e siècle on avait ainsi pour des usages commerciaux ou militaires des dictionnaires de plusieurs milliers de mots de codes. Tout changement du code nécessitait l'envoi de documents volumineux avec un risque d'interception non négligeable.

Ces codes manquent de souplesse ils ne permettent pas de coder des mots nouveaux sans un accord préalable entre l'expéditeur et le destinataire. Pour cela il faut qu'ils échangent des documents ce qui accroit le risque d'interception du code. Ils ne sont pas adaptés à des usages intensifs entre de nombreux correspondants. Ils ne sont pratiquement plus utilisés pour les usages publics. Par contre ils peuvent rendre des services appréciables pour un usage unique.

## B. LA CRYPTOGRAPHIE PAR SUBSTITUTION

Dans les codes de substitution par flots ou par blocs l'ordre des lettres est conservé mais on les remplace par des symboles d'un nouvel alphabet suivant un algorithme précis. Ce procédé est basé sur l'arithmétique modulaire. En fait, étant donné un entier $n \geq 2$, l'arithmétique modulo n consiste à faire des calculs sur les restes dans la division euclidienne des entiers par $\boldsymbol{n}$.

Soient a, b et $n$ des entiers, avec $n \neq 0$; on dit que $a$ est congru à $b$ modulo $n$ et on note $a \equiv b \pmod{n}$ ssi $n$ divise (a - b), c'est-à-dire, s'il existe un entier $x$ tel que $a - b = nx$. Certes, le chiffrement par décalage est un chiffrement qui est défini dans $Z_{26}$. En effet, on utilise souvent les 26 lettres de l'alphabet. Toutefois, ou peut le définir sur n'importe quel $Z_n$. Ce chiffrement forme un système cryptographique tel que $D_k (C_k (x))=x$ pour tout $x$ $Z_{26}$. On peut représenter ce chiffrement par l'image ci-après:

| Lettre | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Nombre | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

D'après *Suetone*, dans son ouvrage « *Vie des douze Césars* », Jules César pendant la guerre des Gaules avait utilisé le code de substitution par flot suivant :

**Exemple$_1$** : lettre codée = lettre claire+3 modulo 26
Le message en clair :

« **RENDEZ-VOUS DEMAIN AUX FAB** ».

Devient :

« **UHQGHC YRXV GHPDLQ DXA IDE** »

On peut considérer toute la famille des codes lettre codée = lettre claire+n modulo 26 où *n* est un entier entre 0 et 25 appelé la clef du code. Avec la clef *n = 7,* le texte codé du message précédent devient :

« **YLUKLG CVBZ KLTHPU HBE MHI** »

Le décodage se fait en utilisant la relation lettre claire = lettre codée -n mod 26
On a affaire à un code en continu ou par flots symétrique ou à clef secrète.

**Exemple$_2$** : À supposer que la clé du chiffrement par décalage soit $k = 10$ et que le texte clair soit :

« **JE SUIS ETUDIANT AUX FAB** »

Pour cet exemple, nous allons d'abord convertir ce texte en une suite d'entiers en utilisant la table correspondante ci-dessus. Ce qui nous donne :

« **9 4 18 20 8 18 4 19 20 3 8 0 13 19 0 20 23 5 0 1** »

Ensuite, étant donné que notre $k = 11$, à chaque valeur nous ajoutons 11 en réduisant cela modulo 26 :

**« 19 14 2 4 18 2 14 3 4 13 18 10 23 3 10 4 7 15 10 11 »**

Enfin, nous allons convertir cette suite d'entiers obtenus en caractère alphabétique tout en respectant la table de correspondance. On obtient :

« **TO CESC ODENSKXD KEH PKL** »

# C. LA CRYPTOGRAPHIE PAR TRANSPOSITION

Dans la cryptographie par transposition, aussi appelée « les *codes de permutation* », On partage le texte en blocs, et on garde le même alphabet mais on change la place des lettres à l'intérieur d'un bloc (*on les permute*).

Un exemple historique dont le principe est encore utilisé est la méthode de la grille (principe de *la scytale* utilisée par les spartiates vers - 450 Av. JC). On veut envoyer le message suivant :

« **RENDEZ VOUS DEMAIN AUX FAB** ».

L'expéditeur et le destinataire du message se mettent d'accord sur une grille de largeur fixée à l' avance (*ici une grille de 6 cases de large*). L'expéditeur écrit le message dans la grille en remplaçant les espaces entre les mots par le symbole @. Il obtient :

| | | | | | |
|---|---|---|---|---|---|
| R | E | N | D | E | Z |
| @ | V | O | U | S | @ |
| D | E | M | A | I | N |
| A | U | X | @ | F | A |
| B | | | | | |

On lira le texte en colonne et obtient ainsi le message crypté :

« **R@DAB EVEU NOMX DUA@ ESIF Z@NA** »

Pour pouvoir modifier le code rapidement sans toucher `a son principe et pouvoir ainsi augmenter la sécurité les deux interlocuteurs peuvent décider l'ajout d'une clef. Le but est de pouvoir changer facilement le cryptage d'un message tout en gardant le même algorithme de codage. Pour cela on rajoute une clef secrète constituée par l'ordre de lecture des colonnes.

## V.3.3.2. LA CRYPTOGRAPHIE POLY-ALPHABETIQUE

Dans cette catégorie, nous distinguerons deux grandes sous-catégories de cryptographie (chiffrement) :

- **La cryptographie de Vigenère ;**
- **La cryptographie de Hill**

## A. LE CHIFFREMENT DE VIGENÈRE

Le **chiffre de Vigenère** est un système de chiffrement poly alphabétique, c'est un chiffrement par substitution, mais une même lettre du message clair peut, suivant sa position dans celui-ci, être remplacée par des lettres différentes, contrairement à un système de chiffrement mono alphabétique comme le chiffre de César (*qu'il utilise cependant comme composant*). Cette méthode résiste ainsi à l'analyse de fréquences, ce qui est un avantage décisif sur les chiffrements mono alphabétiques. Cependant le chiffre de Vigenère a été percé par le major prussien Friedrich Kasiski qui a publié sa méthode en 1863. Il n'offre plus depuis cette époque aucune sécurité.

## PRINCIPE DU CHIFFREMENT EN VIGENERE

Ce chiffrement introduit la notion *de clé.* Une clé se présente généralement sous la forme d'un mot ou d'une phrase. Pour pouvoir *chiffrer* notre texte, à chaque caractère nous utilisons une lettre de la clé pour effectuer la substitution. Évidemment, plus la clé ne sera longue et variée et mieux le texte sera chiffré. Il faut savoir qu'il y a eu une période où des passages entiers d'œuvres littéraires étaient utilisés pour chiffrer les plus grands secrets. Les deux correspondants n'avaient plus qu'à avoir en leurs mains un exemplaire du même livre pour s'assurer de la bonne compréhension des messages.

Pour chaque lettre en clair, on sélectionne la colonne correspondante et pour une lettre de la clé on sélectionne la ligne adéquate, puis au croisement de la ligne et de la colonne on trouve la lettre chiffrée. La lettre de la clé est à prendre dans l'ordre dans laquelle elle se présente et on répète la clé en boucle autant que nécessaire.

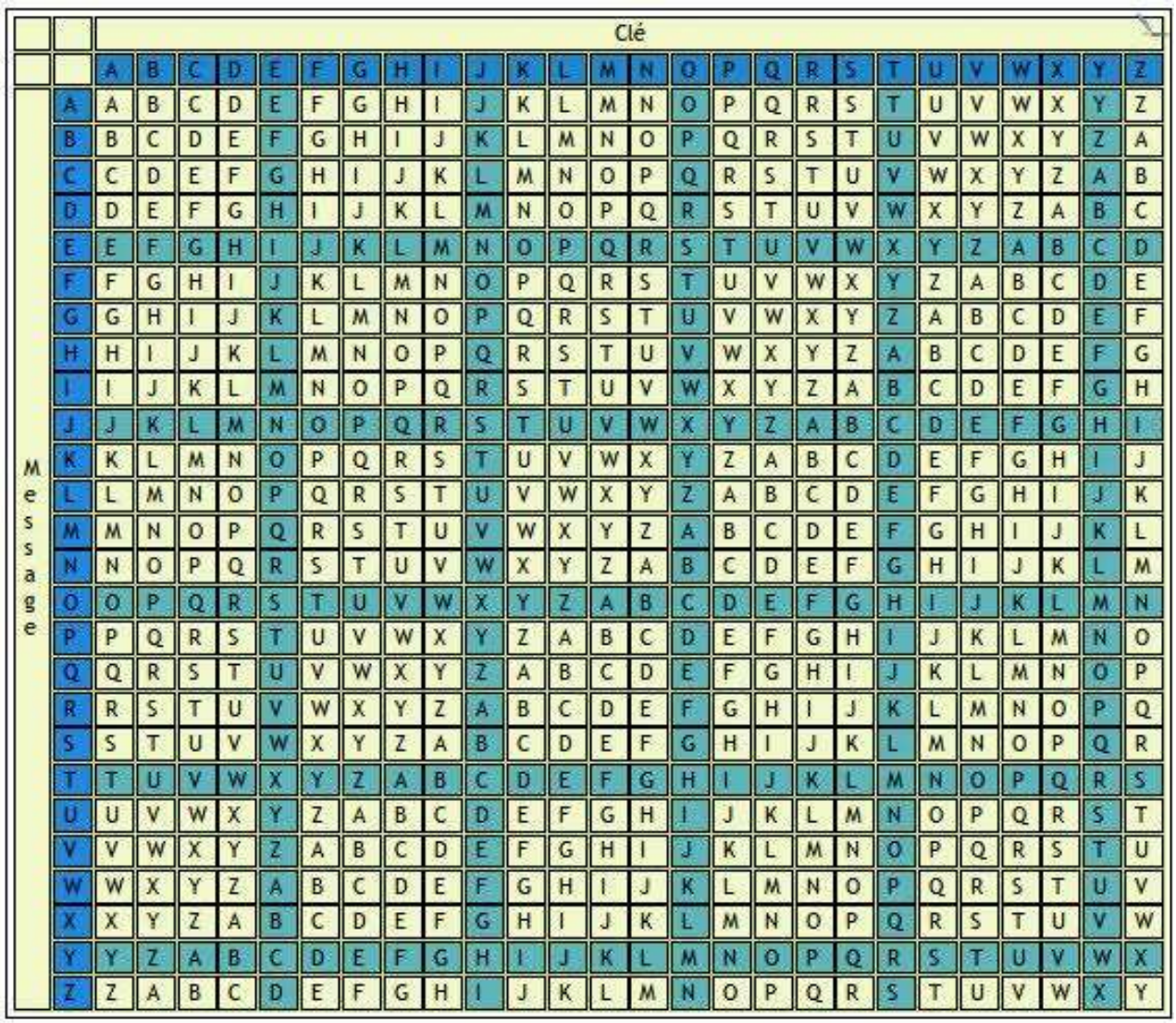
Clé : RECTEUR
Texte : RENDEZ VOUS DEMAIN AUX FAB

Cela donne alors : **R E N D E Z V O U S D E M A I N A U X F A B**
**R E C T E U R R E C T E U R R E C T E U R R**

Le texte chiffré est alors : **IIPWIT NFYU WIGRZR CNB ZRS**

Si on veut déchiffrer ce texte, on regarde pour chaque lettre de la clé répétée la ligne correspondante et on y cherche la lettre chiffrée. La première lettre de la colonne que l'on trouve ainsi est la lettre déchiffrée.

## LA TABLE DE VIGENERE.



| Message \ Clé | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| I | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| J | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| K | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| L | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| M | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| N | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| O | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| P | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| Q | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| R | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| S | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| T | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| U | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| V | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| W | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| X | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| Y | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
| Z | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |

Mathématiquement, on identifie les lettres de l'alphabet aux nombres de 0 à 25 (A=0, B=1...). Les opérations de chiffrement et de déchiffrement sont, pour chaque lettre, celles du chiffre de César. En désignant la $i^e$ lettre du texte clair par Texte[i], la $i^e$ du chiffré par Chiffré[i], et la $i^e$ lettre de la clé, répétée suffisamment de fois, par Clés[i], elle se formalise par :

- Chiffré[i] = (Texte[i] + Clés[i]) modulo 26
- Texte[i] = (Chiffré[i] - Clés[i]) modulo 26

Où x modulo 26 désigne le reste de la division entière de x par 26. Pour le chiffrement il suffit d'effectuer l'addition des deux lettres puis de soustraire 26 si le résultat dépasse 26. Pour le déchiffrement il suffit d'effectuer la soustraction et d'additionner 26 si le résultat est négatif. Le déchiffrement est aussi une opération identique à celle du chiffrement pour la clé obtenue par Clé'[i] = 26 - Clé[i]. Un disque à chiffrer, qui utilise une représentation circulaire de l'alphabet (après Z on a A), permet de réaliser directement cette opération. Le chiffré d'un texte suffisamment long constitué uniquement de A donne la clé ($0 + x = x$, soit A + Clés[i] = Clés[i]).

## B. LE CHIFFREMENT DE HILL

Le chiffrement que nous allons étudier a été publié par *Lester S. Hill* en 1929. C'est un chiffrement polygraphique, c'est-à-dire qu'on ne (dé)chiffre pas les lettres les unes après les autres, mais par paquets. On étudie ici la version biographique, c'est-à-dire que l'on groupe les lettres deux par deux, mais on peut envisager des paquets plus grands. Pour coder un message selon ce procédé, on commence par grouper les lettres de ce message deux par deux, puis on remplace chaque lettre par un nombre.

Chaque caractère est d'abord codé par un nombre compris entre 0 et $n - 1$ (son rang dans l'alphabet diminué de 1 ou son *code ASCII* diminué de 32). Les caractères sont alors regroupés par blocs de $p$ caractères formant un certain nombre de *vecteurs* $X = (x_1, x_2 \ldots x_p)$. Les nombres $\boldsymbol{x_i}$ étant compris entre 0 et $n - 1$, on peut les considérer comme des éléments de $\mathbb{Z}/n\mathbb{Z}$ et $X$ est alors un élément de $(\mathbb{Z}/n\mathbb{Z})^p$. On a construit au préalable une matrice $p \times p$ d'entiers : $A$. Le bloc $X$ est alors chiffré par le bloc $Y = AX$, le produit s'effectuant modulo $n$. Pour déchiffrer le message, il s'agit *d'inverser la matrice A* modulo $n$. Cela peut se faire si le *déterminant* de cette matrice possède un inverse modulo $n$ (c'est-à-dire, d'après le *théorème de Bachet-Bézout, si det(A) est premier avec n)*.

En effet, le produit de $A$ et de la *transposée* de sa *comatrice* donne :

$$A\ {}^{t}\mathrm{com}A = {}^{t}\mathrm{com}A\ A = \det A\ I_p$$

(Où désigne la *matrice identité* de taille $p$) donc s'il existe un entier $k$ tel que :

$$k \times \det(A) \equiv 1 \pmod{n}$$

Alors, en notant $B$ n'importe quelle matrice congrue modulo $n$ à $k\ {}^{t}\mathrm{com}(A)$, on aura :

$$AB \equiv BA \equiv I_p \pmod{n},$$

Soit encore :

$$Y = AX \Leftrightarrow X = BY.$$

**EXEMPLE** : On imagine dans cette section que chaque lettre est codée par son rang dans l'alphabet diminué de 1 et que le chiffrement s'effectue par blocs de 2 lettres. Ici $n = 26$ et $p = 2$. Et l'on cherche à chiffrer le message suivant : TEXTEACRYPTER en utilisant une matrice $A$ dont le déterminant est premier avec 26.

Pour construire une telle matrice, il suffit de choisir trois entiers $a$, $b$, $c$ au hasard mais tels que $a$ soit premier avec 26, ce qui permet de choisir le dernier terme $d$ tel que $ad - bc$ soit inversible modulo 26. Pour la suite on prendra dont le déterminant est 21. Comme $5 \times 21 = 105 \equiv 1 \pmod{26}$, 5 est un inverse de det($A$) modulo 26.

$$A = \begin{pmatrix} 3 & 5 \\ 6 & 17 \end{pmatrix}$$

**CHIFFREMENT**

On remplace chaque lettre par son rang à l'aide du tableau suivant :

| Lettre | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Nombre | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

Puis on code le message :

TEXTEACHIFFRER→ 19 ; 4 ; 23 ; 19 ; 4 ; 0 ; 2 ; 7 ; 8 ; 5 ; 5 ; 17 ; 4 ; 17

On regroupe les lettres par paires créant ainsi 7 vecteurs de dimension deux, la dernière paire étant complétée arbitrairement :

$$X_1 = (19;4); X_2 = (23;19); X_3 = (4;0); X_4 = (2;7); X_5 = (8;5); X_6 = (5;17); X_7 = (4;17).$$

On multiplie ensuite ces vecteurs par la matrice $A$ en travaillant sur des congruences modulo 26 :

$$Y_1 = \begin{pmatrix} 3 & 5 \\ 6 & 17 \end{pmatrix}\begin{pmatrix} 19 \\ 4 \end{pmatrix} = \begin{pmatrix} 25 \\ 0 \end{pmatrix} \text{ etc.}$$

On obtient alors 7 vecteurs, soit 14 lettres :

« **(25 ; 0) ; (8;19) ; (12 ; 24) ; (15 ; 1) ; (23 ; 3) ; (22 ; 7) ; (19 ; 2)** »

« **ZAITMYPBXDWHTB** ».

**DECHIFFREMENT**

Il faut inverser la matrice A. Il suffit de prendre la transposée de sa comatrice, c'est-à-dire :

$$^{t}\mathrm{com}A = \begin{pmatrix} 17 & -5 \\ -6 & 3 \end{pmatrix}$$

et la multiplier (modulo 26) par l'inverse du déterminant de A c'est-à-dire par 21 :

$$B = \begin{pmatrix} 17/21 & -5/21 \\ -6/21 & 3/21 \end{pmatrix}$$

Connaissant les couples Y, il suffit de les multiplier (modulo 26) par la matrice B pour retrouver les couples X et réussir à déchiffrer le message.

## V.3.4. LA CRYPTOGRAPHIE MODERNE

Contrairement à ce que l'on peut penser, la cryptographie n'est pas seulement une technique moderne, ni un produit de l'ère numérique. Certes, de tout temps, les hommes ont éprouvés le besoin de cacher des informations confidentielles. *César envoyait ses messages de manière chiffrée (grâce au très célèbre algorithme dit "de César") mais partait du principe que seul le destinataire connaissait «la clé de déchiffrement ».* Cependant, la majeure partie des techniques d'autrefois reposait sur deux principes fondamentaux: la substitution (remplacement de certaines lettres par d'autres) et la transposition (permutation des lettres des messages en vue de brouiller l'intercepteur). Dans cette section, nous allons aborder des nouvelles techniques de chiffrements basées sur des principes mathématiques. Il sera question de :

- La cryptographie symétrique ;
- La cryptographie asymétrique.

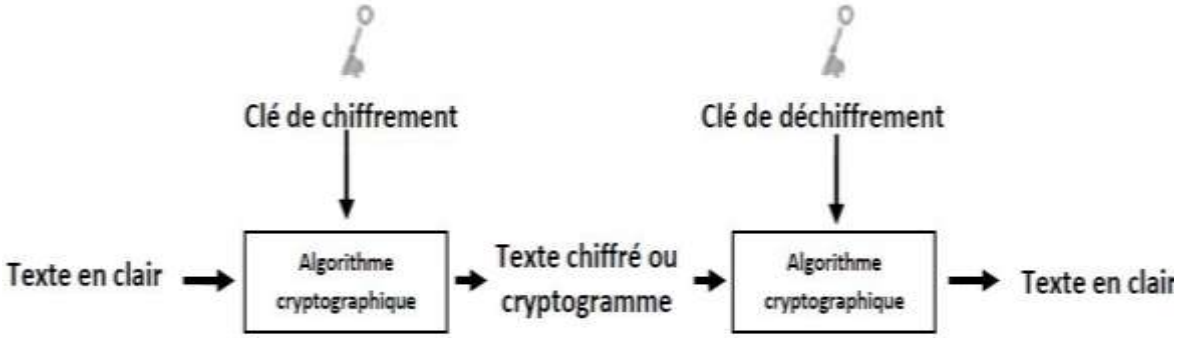
### V.3.4.1. LA CRYPTOGRAPHIE SYMETRIQUE

Aussi appelé chiffrement à clé privée ou chiffrement à clé secrète, il consiste à utiliser la même clé pour le chiffrement et le déchiffrement.

**Caractéristiques :**

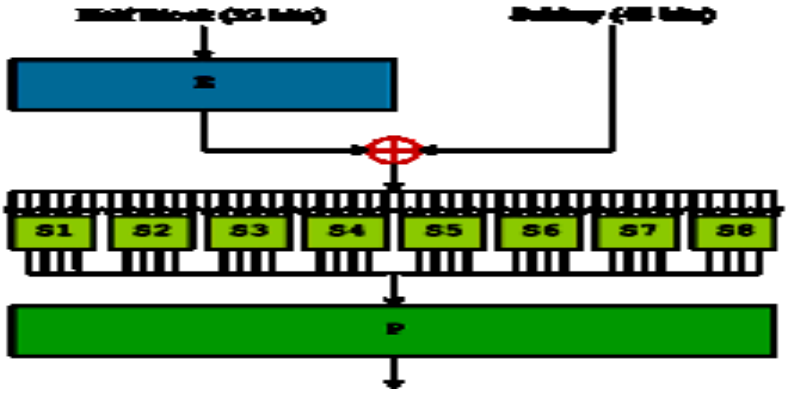
- Les clés sont identiques : KE = KD = K ;
- La clé doit rester secrète ;
- Les algorithmes les plus répandus sont le DES, AES, 3DES, ...
- Au niveau de la génération des clés, elle est choisie aléatoirement dans l'espace des clés ;
- Ces algorithmes sont basés sur des opérations de transposition et de substitution des bits du texte clair en fonction de la clé ;

- La taille des clés est souvent de l'ordre de 128 bits. Le DES en utilise 56, mais l'AES peut aller jusque 256 ;
- L'avantage principal de ce mode de chiffrement est sa rapidité ;
- Le principal désavantage réside dans la distribution des clés : pour une meilleure sécurité, on préfèrera l'échange manuel. Malheureusement, pour de grands systèmes, le nombre de clés peut devenir conséquent. C'est pourquoi on utilisera souvent des échanges sécurisés pour transmettre les clés. En effet, pour un système à N utilisateurs, il y aura N*(N − 1)/2 paires de clés.




Les algorithmes de chiffrement symétrique se fondent sur une même clé pour chiffrer et déchiffrer un message. L'un des problèmes de cette technique est que la clé, qui doit rester totalement confidentielle, doit être transmise au correspondant de façon sûre. La mise en œuvre peut s'avérer difficile, surtout avec un grand nombre de correspondants car il faut autant de clés que de correspondants. Quelques algorithmes de chiffrement symétrique très utilisés : *Chiffre de Vernam* (le seul offrant une sécurité théorique absolue, à condition que la clé ait au moins la même longueur que le message à chiffrer, qu'elle ne soit utilisée qu'une seule fois et qu'elle soit totalement aléatoire) ; *DES ; 3DES ; AES ; RC4 ; RC5 ; MISTY1.*

## A. DES (DATA ENCRYPTION STANDARD)




***Fonction – F des DES***

Le **Data Encryption Standard** (**DES**) est un algorithme de chiffrement symétrique (chiffrement par bloc) utilisant des clés de 56 bits. Son emploi n'est plus recommandé aujourd'hui, du fait de sa lenteur à l'exécution 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 premier standard DES est publié par FIPS le 15 janvier 1977 sous le nom FIPS PUB 46. La dernière version avant l'obsolescence date du 25 octobre 1999[1]. L'algorithme initialement conçu par IBM utilisait une clé de 112 bits. L'intervention de la NSA a ramené la taille de clé à 56 bits. De nos jours, le Triple DES reste très répandu, et le DES « simple » ne subsiste que dans d'anciennes applications. Le standard DES a été remplacé en 2001 par l'AES (*Advanced Encryption Standard*). En mai 1973, le *National Bureau of Standards* américain demande la création d'un algorithme de chiffrement utilisable par les entreprises. À cette époque, IBM dispose déjà d'un algorithme appelé Lucifer, conçu en 1971 par Horst Feistel.

En bonne logique, cet algorithme aurait dû être sélectionné par le NBS. En pratique, ce fut presque le cas : la NSA demanda que *Lucifer* soit modifié, par ses soins. Ainsi fut créé le DES, qui fut adopté comme standard en novembre 1976. Cela suscita des soupçons selon lesquels la NSA aurait volontairement affaibli l'algorithme, dans le but de pouvoir le casser. Étrangement, le DES s'est révélé résistant à plusieurs attaques ne devant apparaître dans la communauté académique que beaucoup plus tard. Encore plus étonnant, *Lucifer*, lui, résistait moins bien. Ceci permet de penser que la NSA avait connaissance dès cette époque de ces techniques de cryptanalyse et qu'elle aurait donc, en réalité, rendu DES moins faible[18].

## FONCTIONNEMENT DE DES

L'algorithme DES transforme un bloc de 64 bits en un autre bloc de 64 bits. Il manipule des clés individuelles de 56 bits, représentées par 64 bits (avec un bit de chaque octet servant pour le contrôle de parité). Ce système de chiffrement symétrique fait partie de la famille des chiffrements itératifs par blocs, plus particulièrement il s'agit d'un schéma de Feistel *(du nom de Horst Feistel à l'origine du chiffrement Lucifer*). D'une manière générale, on peut dire que DES fonctionne en trois étapes :

- Permutation initiale et fixe d'un bloc (*sans aucune incidence sur le niveau de sécurit*é) ;
- Le résultat est soumis à 16 itérations d'une transformation, ces itérations dépendent à chaque tour d'une autre clé partielle de 48 bits. Cette clé de tour

[18] Don Coppersmith, « The Data Encryption Standard (DES) and its strength against attacks », *IBM Journal of Research and Development*, vol. 38, n° 3, mai 1994, p. 243

intermédiaire est calculée à partir de la clé initiale de l'utilisateur (*grâce à un réseau de tables de substitution et d'opérateurs XOR*). Lors de chaque tour, le bloc de 64 bits est découpé en deux blocs de 32 bits, et ces blocs sont échangés l'un avec l'autre selon un schéma de Feistel. Le bloc de 32 bits ayant le poids le plus fort (*celui qui s'étend du bit 32 au bit 64*) subira une transformation ;

- le résultat du dernier tour est transformé par la fonction inverse de la permutation initiale.

Le DES utilise huit tables de substitution (*les S-Boxes*) qui furent l'objet de nombreuses controverses quant à leur contenu. On soupçonnait une faiblesse volontairement insérée par les concepteurs. Ces rumeurs furent dissipées au début des années 1990 par la découverte de la cryptanalyse différentielle qui démontra que les tables étaient bien conçues.

## LE TRIPLE DES

Le Triple DES (aussi appelé 3DES) est un algorithme de chiffrement symétrique enchainant trois applications successives de l'algorithme DES sur le même bloc de données de 64 bits, avec 2 ou 3 clés DES différentes. Cette utilisation de trois chiffrements DES a été développée par Walter TUCHMAN. Même quand 3 clés de 56 bits différentes sont utilisées, la force effective de l'algorithme n'est que de 112 bits et non 168 bits, à cause d'une attaque type rencontre au milieu. Bien que normalisé, bien connu, et assez simple à implémenter, il est assez lent.

## V.3.4.2. LA CRYPTOGRAPHIE ASYMETRIQUE

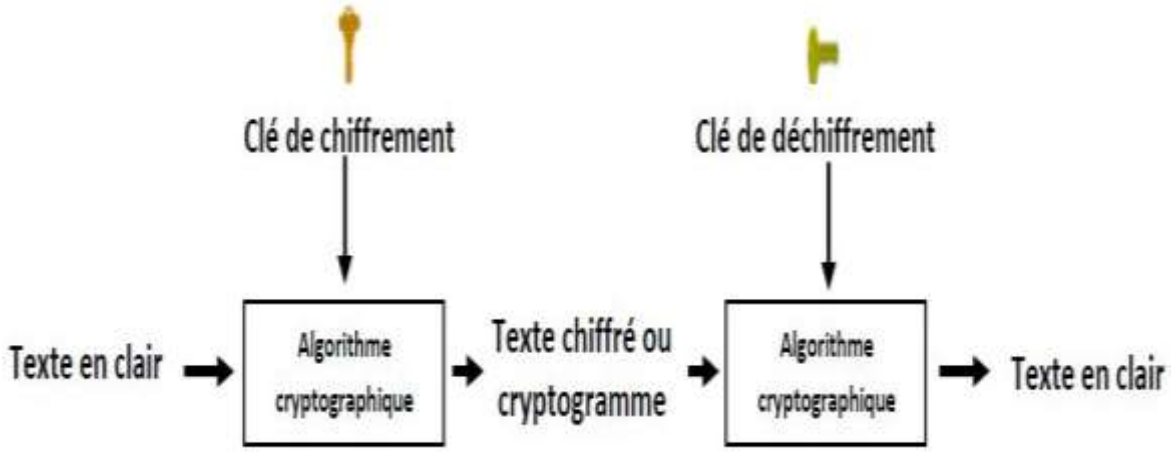
Ce chiffrement est aussi appelé chiffrement à clés publiques. Apparu en 1976, avec la publication de l'ouvrage sur la cryptographie de Wilfried Diffie et Martin Hellman, il a pour caractéristiques les éléments suivants :

- Une clé publique PK (symbolisée par la clé verticale) ;
- Une clé privée secrète SK (symbolisée par la clé horizontale) ;
- Propriété : La connaissance de PK ne permet pas de déduire SK ;
- DSK(EPK(M)) = M ;
- L'algorithme de cryptographie asymétrique le plus connu est le RSA ;
- Le principe de ce genre d'algorithme est qu'il s'agit d'une fonction unidirectionnelle à trappe. Une telle fonction a la particularité d'être facile à calculer dans un sens, mais difficile voire impossible dans le sens inverse. La

seule manière de pouvoir réaliser le calcul inverse est de connaître une trappe. Une trappe pourrait par exemple être une faille dans le générateur de clés. Cette faille peut être soit intentionnelle de la part du concepteur (définition stricte d'une trappe) ou accidentelle.

- Les algorithmes se basent sur des concepts mathématiques tels que l'exponentiation de grands nombres premiers (RSA), le problème des logarithmes discrets (ElGamal), ou encore le problème du sac à dos (Merkle-Hellman).
- La taille des clés s'étend de 512 bits à 2048 bits en standard. Dans le cas du RSA, une clé de 512 bits n'est plus sûre au sens "militaire" du terme, mais est toujours utilisable de particulier à particulier.
- Au niveau des performances, le chiffrement par voie asymétrique est environ 1000 fois plus lent que le chiffrement symétrique.
- Cependant, à l'inverse du chiffrement symétrique où le nombre de clés est le problème majeur, ici, seules n paires sont nécessaires. En effet, chaque utilisateur possède une paire (SK, PK) et tous les transferts de message ont lieu avec ces clés.
- La distribution des clés est grandement facilitée car l'échange de clés secrètes n'est plus nécessaire. Chaque utilisateur conserve sa clé secrète sans jamais la divulguer. Seule la clé publique devra être distribuée.

En somme, ici, il y a une clé publique pour le chiffrement et une clé secrète pour le déchiffrement.




***Cryptographie asymétrique***

# A. LE RSA (RIVEST SHAMIR ADELMAN)

Le **chiffrement RSA** (*nommé par les initiales de ses trois inventeurs*) est un algorithme de cryptographie asymétrique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Cet algorithme a été décrit en 1977 par *Ronald Rivest, Adi Shamir* et *Leonard Adleman*. RSA a été breveté par le Massachusetts Institute of Technology (MIT) en 1983 aux États-Unis. Le brevet a expiré le 21 septembre 2000.

## FONCTIONNEMENT GÉNÉRAL DU RSA

Le chiffrement RSA est *asymétrique* : il utilise une paire de clés (des nombres entiers) composé d'une *clé publique* pour chiffrer et d'une *clé privée* pour déchiffrer des données confidentielles. Les deux clés sont créées par une personne, souvent nommée par convention *Alice*, qui souhaite que lui soient envoyées des données confidentielles. Alice rend la clé publique accessible. Cette clé est utilisée par ses correspondants (*Bob*, etc.) pour chiffrer les données qui lui sont envoyées. La clé privée est quant à elle réservée à Alice, et lui permet de déchiffrer ces données. La clé privée peut aussi être utilisée par Alice pour signer une donnée qu'elle envoie, la clé publique permettant à n'importe lequel de ses correspondants de vérifier la signature.

Une condition indispensable est qu'il soit « *calculatoirement impossible* » de déchiffrer à l'aide de la seule clé publique, en particulier de reconstituer la clé privée à partir de la clé publique, c'est-à-dire que les moyens de calcul disponibles et les méthodes connues au moment de l'échange (et le temps que le secret doit être conservé) ne le permettent pas. Le chiffrement RSA est souvent utilisé pour communiquer une clé de chiffrement symétrique, qui permet alors de poursuivre l'échange de façon confidentielle : Bob envoie à Alice une clé de chiffrement symétrique qui peut ensuite être utilisée par Alice et Bob pour échanger des données.

Ils utilisent *les congruences sur les entiers* et *le petit théorème de Fermat*, pour obtenir des fonctions à sens unique, avec brèche secrète (ou porte dérobée). Tous les calculs se font modulo un nombre entier $n$ qui est le produit de deux nombres premiers. *Le petit théorème de Fermat* joue un rôle important dans la conception du chiffrement. Les messages clairs et chiffrés sont des entiers inférieurs à l'entier $n$ (*tout message peut être codé par un entier*). Les opérations de chiffrement et de déchiffrement consistent à élever le message à une certaine puissance modulo $n$ (*c'est l'opération d'exponentiation modulaire*).

## V.3.5. LA CRYPTANALYSE

**La cryptanalyse** est la technique qui consiste à déduire un texte en clair d'un texte chiffré sans posséder la clé de chiffrement. Le processus par lequel on tente de comprendre un message en particulier est appelé une *attaque*. Une attaque est souvent caractérisée par les données qu'elle nécessite :

- *attaque sur texte chiffré seul* (*ciphertext-only* en anglais) : le cryptanalyste possède des exemplaires chiffrés des messages, il peut faire des hypothèses sur les messages originaux qu'il ne possède pas. La cryptanalyse est plus ardue de par le manque d'informations à disposition.
- *attaque à texte clair connu* (*known-plaintext attack* en anglais) : le cryptanalyste possède des messages ou des parties de messages en clair ainsi que les versions chiffrées. La cryptanalyse linéaire fait partie de cette catégorie.
- *attaque à texte clair choisi* (*chosen-plaintext attack* en anglais) : le cryptanalyste possède des messages en clair, il peut créer les versions chiffrées de ces messages avec l'algorithme que l'on peut dès lors considérer comme une boîte noire. La cryptanalyse différentielle est un exemple d'attaque à texte clair choisi.
- *attaque à texte chiffré choisi* (*chosen-ciphertext attack* en anglais) : le cryptanalyste possède des messages chiffrés et demande la version en clair de certains de ces messages pour mener l'attaque.

### V.3.5.1. LES ATTAQUES CRYPTANALYTIQUES CLASSIQUES

Il existe plusieurs familles d'attaques cryptanalytiques, les plus connues étant :

- **L'analyse fréquentielle -** L'analyse fréquentielle, découverte au IX$^{e}$ siècle par Al-Kindi, examine les répétitions des lettres du message chiffré afin de trouver la clé. Elle est inefficace contre les chiffrements modernes tels que DES, RSA. Elle est principalement utilisée contre les chiffrements mono-alphabétiques qui substituent chaque lettre par une autre et qui présentent un biais statistique.
- **L'indice de coïncidence -** L'indice de coïncidence, inventé en 1920 par William F. Friedman, permet de calculer la probabilité de répétitions des lettres du message chiffré. Il est souvent couplé avec l'analyse fréquentielle. Cela permet de savoir le type de chiffrement d'un message (*chiffrement mono-alphabétique ou poly-alphabétique*) ainsi que la longueur probable de la clé.

- **L'attaque par mot probable -** L'attaque par mot probable consiste à supposer l'existence d'un mot probable dans le message chiffré. Il est donc possible d'en déduire la clé du message si le mot choisi est correct. Ce type d'attaque a été mené contre la machine *Enigma* durant la Seconde Guerre mondiale.

- **L'attaque par dictionnaire -** L'attaque par dictionnaire consiste à tester tous les mots d'une liste comme mot clé. Elle est souvent couplée à l'attaque par force brute.

- **L'attaque par force brute -** L'attaque par force brute consiste à tester toutes les solutions possibles de mots de passe ou de clés. C'est le seul moyen de récupérer la clé dans les algorithmes les plus modernes et encore inviolés comme AES. Il est peu utilisé pour des mots de passe possédant un très grand nombre de caractères car le temps nécessaire devient alors trop important.

- **L'attaque par paradoxe des anniversaires -** Le paradoxe des anniversaires est un résultat probabiliste qui est utilisé dans les attaques contre les fonctions de hachage. Ce paradoxe permet de donner une borne supérieure de résistance aux collisions d'une telle fonction. Cette limite est de l'ordre de la racine de la taille de la sortie, ce qui signifie que, pour un algorithme comme MD5 (*empreinte sur 128 bits*), trouver une collision quelconque avec 50 % de chance nécessite $2^{64}$ hachages d'entrées distinctes.

## V.3.5.2. LA CRYPTANALYSE MODERNE

Dès les années 70 apparaissent les méthodes de chiffrement modernes par blocs comme DES. Il sera passablement étudié et attaqué ce qui mènera à des attaques majeures dans le monde de la cryptographie. Les méthodes présentées ci-dessous ne sont pas vraiment génériques et des modifications sont nécessaires pour attaquer un type de chiffrement donné. Souvent, on ne s'attaque pas à une version complète de l'algorithme de chiffrement mais une variante avec moins de tours (*dans le cas des schémas de type Feistel ou les fonctions de hachage*). Cette analyse préliminaire, si elle permet de déceler des vulnérabilités, laisse entrevoir une attaque sur l'algorithme complet :

- **La Cryptanalyse linéaire** - La cryptanalyse linéaire, due à Mitsuru Matsui, consiste à faire une approximation linéaire de la structure interne de la méthode de chiffrement. Elle remonte à 1993 et s'avère être l'attaque la plus efficace contre DES. Les algorithmes plus récents sont insensibles à cette attaque.

- **La Cryptanalyse différentielle** - La cryptanalyse différentielle est une analyse statistique des changements dans la structure de la méthode de chiffrement après avoir légèrement modifié les entrées. Avec un très grand nombre de perturbations, il est possible d'extraire la clé. Cette attaque date de 1990 (présentée à la conférence *Crypto 90*). Elle est due à Eli Biham et Adi Shamir. Toutefois, on sait maintenant que les concepteurs de DES connaissaient une variante de cette attaque nommée *attaque-T*. Les algorithmes récents (AES, IDEA, etc.) sont conçus pour résister à ce type d'attaque. Les attaques différentielles sont aussi possibles sur les fonctions de hachage, moyennant des modifications dans la conduite de l'attaque. Une telle attaque a été menée contre MD5.
- **La Cryptanalyse différentielle-linéaire** - Introduite par *Martin Hellman* et Langford en 1994, la cryptanalyse différentielle-linéaire combine les deux principes. L'attaque différentielle produit une approximation linéaire de l'algorithme. Avec cette attaque, Hellman et Langford ont pu attaquer un DES de 8 rondes avec seulement 512 textes en clair et quelques secondes sur un PC de l'époque. Cette méthode a également été employée pour trouver des clés faibles dans IDEA. Ce type de cryptanalyse a été améliorée par Eli Biham en 2002.

- **La Cryptanalyse $\chi^2$** - La cryptanalyse $\chi^2$, concept dû à Serge Vaudenay, permet d'obtenir des résultats similaires à des attaques linéaires ou différentielles. L'analyse statistique associée permet de s'affranchir des défauts de ces dernières en évitant d'avoir à connaître le fonctionnement exact du chiffrement.

- **La Cryptanalyse quadratique** - La cryptanalyse quadratique est une invention récente de Nicolas Courtois et Josef Pieprzyk. Cette attaque (nommée *attaque XSL*) vise en particulier AES et les autres chiffrements basés sur Rijndael. L'attaque XSL est le sujet de beaucoup de controverses quant à sa véritable efficacité de par sa nature heuristique. Elle consiste à résoudre un système d'équations de très grande taille.

- **La Cryptanalyse modulo n** - Suggérée par Bruce Schneier, David Wagner et John Kelsey en 1999, cette technique consiste à exploiter les différences de fonctionnement (*selon une congruence variable*) des algorithmes qui utilisent des rotations binaires.

- **Les attaques par canal auxiliaire** - Les attaques par canaux auxiliaires font partie d'une vaste famille de techniques cryptanalytiques qui exploitent des propriétés inattendues d'un algorithme de cryptographie lors de son implémentation logicielle ou matérielle. En effet, une sécurité « *mathématique* » ne garantit pas forcément une sécurité lors de l'utilisation « en pratique ». Les attaques portent sur différents paramètres : le temps, le bruit, la consommation électrique, etc.

- **Le Compromis temps/mémoire** - Ce concept a été introduit par *Martin Hellman* en 1980. Il a été amélioré en 1993 par Philippe Oechslin avec le concept de table arc-en-ciel, qui lui a permis par exemple d'attaquer les mots de passe de sessions Windows, lorsqu'ils sont stockés au format *LanManager*, comme c'est encore le plus souvent le cas. Il s'agit d'un compromis entre une attaque par force brute et l'utilisation de dictionnaires. Une recherche exhaustive nécessite en effet beaucoup de temps alors qu'un dictionnaire de tous les mots de passe possibles nécessiterait énormément de place. Grâce à des procédés algorithmiques, on cherche à trouver un juste milieu entre ces deux principes, en construisant des tables de taille gérable.

- **Les attaques sur les modes opératoires** - Les chiffrements par bloc comme DES ou AES ne peuvent chiffrer qu'un bloc de taille donnée (*128 bits dans le cas d'AES*). Pour chiffrer des données plus longues, on utilise des modes opératoires. Un mode opératoire est la manière de chaîner plusieurs blocs ensemble pour obtenir un chiffrement par flux. Par exemple, on peut découper les données en blocs de 128 bits et les chiffrer séparément. C'est le mode ECB qui est vulnérable puisque la présence de deux blocs chiffrés identiques indique que les deux blocs respectifs dans le message original sont également identiques. D'autres modes évitent ce problème mais ne sont pas totalement exempts de vulnérabilités. On utilise alors des vecteurs d'initialisation qui permettent d'éviter la répétition de séquences identiques entre plusieurs messages chiffrés. Les chiffrements par flot (par exemple RC4) utilisent aussi un vecteur d'initialisation pour les mêmes raisons. Une telle attaque a été récemment menée à ce propos sur le chiffrement des documents de la suite Microsoft Office, qui emploie RC4. Le vecteur d'initialisation y est toujours le même pour un document donné ; un grand nombre d'informations peuvent donc être récupérées en comparant le résultat du chiffrement d'un document après légère modification.

- **Les attaques par rencontre au milieu** - Chiffrer deux fois avec le même algorithme mais via deux clés différentes n'est pas équivalent à un chiffrement avec une clé deux fois plus longue (*dans le cas de DES, on ne passe pas de $2^{56}$ à $2^{112}$ opérations pour casser le chiffrement*), à cause d'une attaque dite *par rencontre au milieu*, de type compromis temps-mémoire. L'attaque fonctionne théoriquement de la façon suivante, dans le cas d'un double chiffrement, on suppose connus un clair $M$ et un chiffré $C$, $C$ étant obtenu par deux applications d'un même chiffrement avec deux clefs a priori distinctes. Il s'agit de déterminer un couple de clefs qui permet de passer de $M$ à $C$ par double chiffrement. L'opération peut être réitérée sur d'autres couples de clair-chiffré, s'il ne reste pas qu'un seul couple de clefs possibles. Là où les couples de clés candidates sont ceux qui permettent d'obtenir le même bloc par un seul chiffrement de $M$ d'une part, par un seul déchiffrement de $C$ d'autre part (c'est la rencontre au milieu).

Vue ainsi, l'attaque permet un compromis temps-mémoire :

→ on peut stocker tous les blocs obtenus à partir de *M* par une seule opération de chiffrement en essayant toutes les clefs possibles ;

→ puis pour toutes les clefs possibles, et pour chaque bloc obtenu à partir de *C* par une seule opération de déchiffrement, chercher parmi les blocs stockés lors de l'étape précédente un bloc identique.

Le couple des deux clefs qui ont permis d'obtenir ce bloc intermédiaire (*l'une en chiffrant M l'autre en déchiffrant C*) est alors candidat à être la clé du double chiffrement. Dans le cas du DES et pour un bloc de donnée de 64 bits, la première étape demande $2^{56}$ opérations de chiffrement, et un espace mémoire de $2^{56}$ blocs de 64 bits, la seconde $2^{56}$ opérations (plus à chaque fois la recherche du bloc). La complexité de l'attaque par rencontre au milieu sur le double chiffrement a été seulement multipliée par 2 (*en négligeant l'étape finale de comparaison*) vis-à-vis de l'attaque par recherche exhaustive sur le chiffrement simple, alors que pour l'attaque par force brute on passe au carré. Elle nécessite cependant un espace mémoire considérablement augmenté, mais des ajustements sont possibles (*compromis temps-mémoire moins radicaux*) si l'espace mémoire est trop important. L'attaque vaut en fait également pour l'enchaînement de deux chiffrements différents, et il est possible de pratiquer symétriquement. Dans le cas de DES, on obtient une attaque théorique de l'ordre de $2^{57}$ opérations de chiffrements (*sans modification elle demanderait un espace mémoire de $2^{56}\times 64$ bits*). C'est à cause de cette attaque que le double DES n'est pas utilisé, mais aussi que l'on estime la sécurité du 3DES avec 3 clefs distinctes (*168 bits*) à 112 bits, soit de l'ordre de $2^{112}$ opérations pour casser le chiffrement.

- **Les attaques sur les systèmes asymétriques** - Casser un chiffrement assuré par de la cryptographie asymétrique nécessite d'autres approches. Dans le cas de RSA, c'est la difficulté de la factorisation qui assure la résistance du chiffrement. Pour ElGamal, c'est le problème du logarithme discret qui est employé. Toutefois, certaines failles peuvent apparaître selon l'utilisation que l'on fait de ces algorithmes. RSA est vulnérable si des exposants de faible magnitude sont utilisés (*attaques de Don Coppersmith et Wiener*). Sous des conditions particulières, un surchiffrement avec RSA peut être attaqué. Le standard PKCS assure une utilisation plus robuste de RSA, même si les premières ébauches du standard étaient sensibles à des attaques par des canaux auxiliaires (*Bleichenbacher*).

# SIXIEME CHAPITRE – INTRODUCTION A LA BIOMETRIE

La sécurité est une préoccupation de plus en plus importante au sein des entreprises et commence par l'accès à l'information. Pour se prémunir contre d'éventuelles personnes indélicates, une nouvelle technique de contrôle d'accès a fait son apparition et ne cesse de croître depuis 1997 : il s'agit des contrôles d'accès par les systèmes biométriques. Ces systèmes sont utilisés aussi bien pour des contrôles d'accès physiques que pour des contrôles d'accès logiques. Depuis 2001, sont organisés des salons professionnels entièrement consacrés à ce type de technique. Les techniques de contrôle d'accès sont basées sur les critères suivants :

- Ce que l'on sait ;
- Ce que l'on possède ;
- Ce que l'on est.

## VI. DEFINITION

Un système de contrôle biométrique est un système automatique de mesure basé sur la reconnaissance de caractéristiques propres à un individu : physique, comportement et expérimental. Le mot « *biométrie* » utilisé dans le domaine de la sécurité est une traduction de l'anglais « biométrics » qui correspond en fait à un notre mot « *anthropométrie* : *ensemble des techniques de mesure de l'organisme humain utilisées en anthropobiologie et dans le domaine de la justice* ».

Le mot français biométrie définit « *l'étude mathématique des variations biologiques à l'intérieur d'un groupe déterminé* ». La biométrie est basée sur l'analyse de données liées à l'individu et peut être classée en trois grandes catégories :

- Analyse basée sur l'analyse morphologique. (*empreinte digitale, forme de la main, les traits du visage, réseau veineux de la rétine, iris de l'œil, voix, etc*.) ;

- Analyse de traces biologiques. (odeur, salive, urine, sang, ADN, etc.) ;

- Analyse basée sur l'analyse comportementale. (dynamique du tracé de signature, frappe sur un clavier d'ordinateur).

La biométrie est une méthode d'identification basée sur certaines données humaines, physiques ou comportementales, qui peuvent aller de la rétine, à l'empreinte digitale en passant par la voix ou la forme de la main. Chacune de ces méthodes est ce que l'on nomme un « *Moyen Biométrique* ». Les techniques biométriques furent mises au point pour pallier le problème des pertes de mot de passe et autres vols de cartes à puce. Comme cela vient d'être dit, on classe les moyens en deux catégories :

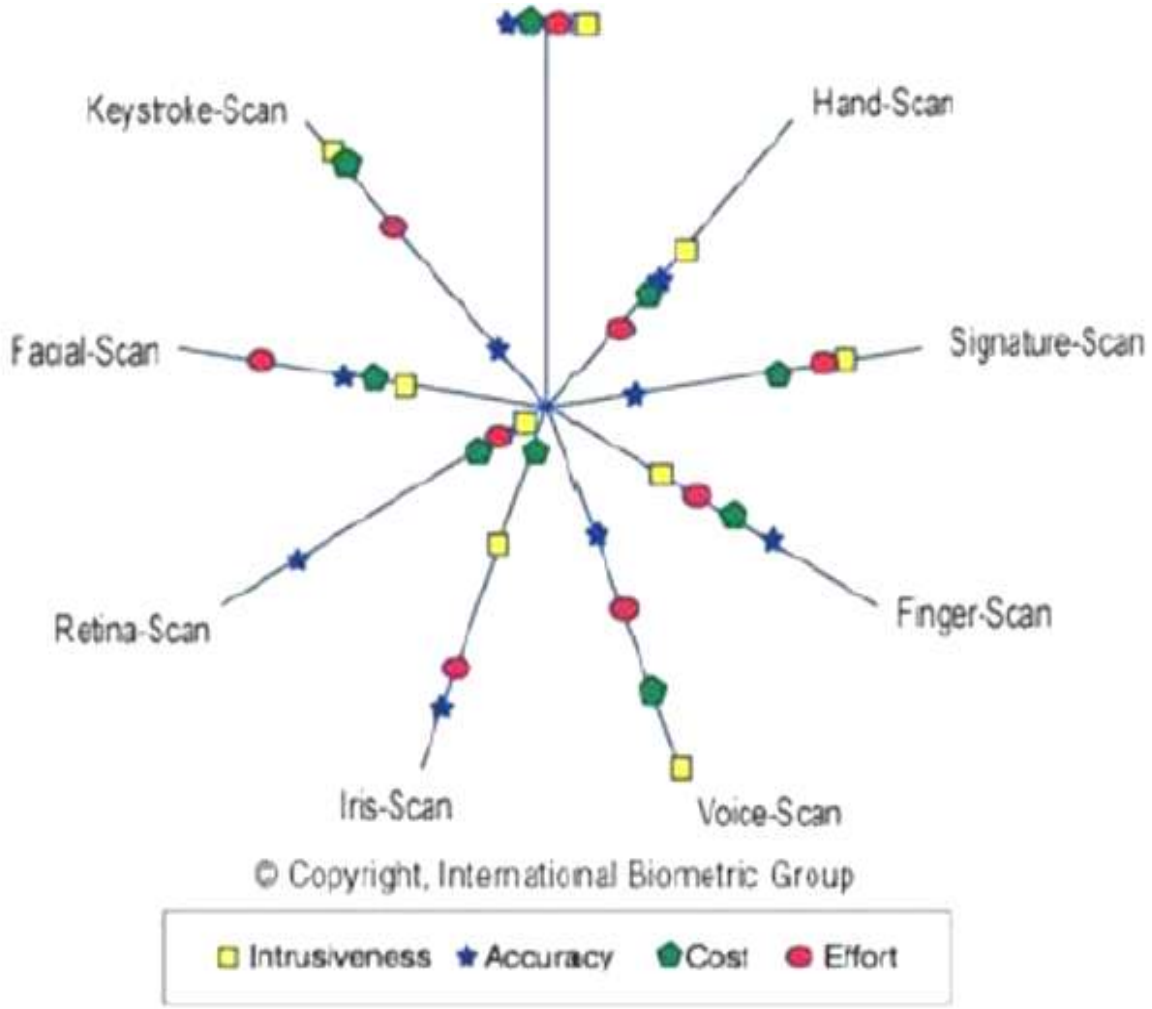
- Les moyens physiques (physiologiques) tels que l'iris, l'empreinte digitale, . . .
- Les moyens comportementaux tels que la voix, la dynamique de signature ou de frappe, . . .

Chaque moyen possède ses avantages, ses inconvénients et ses applications. Plusieurs questions devront être posées telles que :

- Le mode de saisie est-il optimal ?
- Quel type d'analyse est-il permis d'opérer ?
- Quelles sont les capacités de stockage disponibles ?
- Quel est le type de vérification demandé ?

Les différents moyens n'ont pas le même niveau de sécurité, ni la même facilité de mise en œuvre. De plus, à la différence des systèmes travaillant à partir de mots de passe, les réponses ne seront jamais fiables à 100%. Dans le cas des mots de passe, on peut tout de suite dire si la phase d'authentification a réussi ou non. Avec la biométrie, il faudra travailler selon des taux de similitude prédéfinis. Il faudra aussi veiller au fait que les données humaines se modifient avec le temps, et que la manière avec laquelle sont relevées ces mêmes données n'est jamais identique. Par exemple, le doigt n'est jamais posé tout à fait de la même manière sur le support de saisie. Un problème similaire se produit avec l'écartement des doigts de la main. La figure ci-dessous illustre les différents moyens biométriques, ainsi que les critères de référence utilisés pour les classer. On définira ces derniers comme suit :

- *Intrusiveness* : indique à quel point l'utilisateur se sent "agressé" par la méthode d'identification ;
- *Accuracy* : indique l'efficacité du système pour identifier un utilisateur ;
- *Cost* : indique le coût de la mise en place d'un tel système (lecteurs, capteurs, stockage, ...) ;
- *Effort* : indique l'effort nécessité par l'utilisateur pour permettre la mesure.




## VI.2. CARACTÉRISTIQUES COMMUNES DES SYSTÈMES BIOMÉTRIQUES

- **L'unicité** - Pour identifier ou authentifier une personne au sein d'une population donnée, il est nécessaire que la donnée biométrique utilisée soit unique à cette personne. L'empreinte digitale, la rétine et l'iris sont réputés pour présenter des caractéristiques uniques au sein de très larges populations. En particulier, ces techniques permettent de distinguer les vrais jumeaux, et l'empreinte digitale est reconnue juridiquement comme identifiant un individu. Ces caractéristiques uniques tiennent autant à l'environnement aléatoire de leur formation qu'au patrimoine génétique. Cette formation aléatoire est illustrée par exemple par les variations de robe des animaux clonés. D'autres techniques biométriques sont beaucoup plus liées au patrimoine génétique. C'est le cas de la forme de la main ou du visage qui n'ont pas vraiment la capacité de distinguer de vrais jumeaux.

- **Caractère public d'une donnée biométrique** - Un code personnel (PIN) est secret et doit le rester pour qu'un système de contrôle d'accès fonctionne. Une caractéristique biométrique n'est pas secrète. Elle peut être plus ou moins facilement capturée et imitée. Un système de contrôle d'accès biométrique doit donc prendre en compte cette menace et éliminer les artefacts construits pour le tromper.

- **Mesure d'un système biométrique** - Un système biométrique n'utilise pas toute l'information contenue dans l'image ou le signal capté. Il en extrait certaines caractéristiques, ce qui réduit la quantité d'information, donc la capacité du système à reconnaître l'unicité d'une donnée. Puis il effectue un calcul et obtient un résultat à partir des données recueillies. Sa robustesse dépend du nombre de critères retenus et de la méthode de modélisation (*ou de calcul*) utilisée. Un système biométrique est alors mesuré par deux paramètres :

  → Le taux de fausse acceptation, qui est la probabilité de confusion d'identité (*FAR*) ;

  → Le taux de faux rejet, qui est la probabilité de ne pas reconnaître une identité lors d'un essai (*FRR*) ;

  → Un troisième paramètre mesure l'utilité du système. C'est le taux d'échec à l'enrôlement, qui traduit la probabilité d'absence d'une caractéristique biométrique pour un individu dans une population (*FER*).

Les types d'application les plus courants sont :

- Accès à des locaux sensibles. (équipements techniques, archives, stocks, laboratoires ;
- casino, coffres des banques, etc.) ;
- Gestion d'horaire, etc.
- Contrôles d'accès logiques.

L'ensemble des secteurs d'activité ayant un besoin d'authentification forte est susceptible de recourir aux contrôles d'accès biométriques.

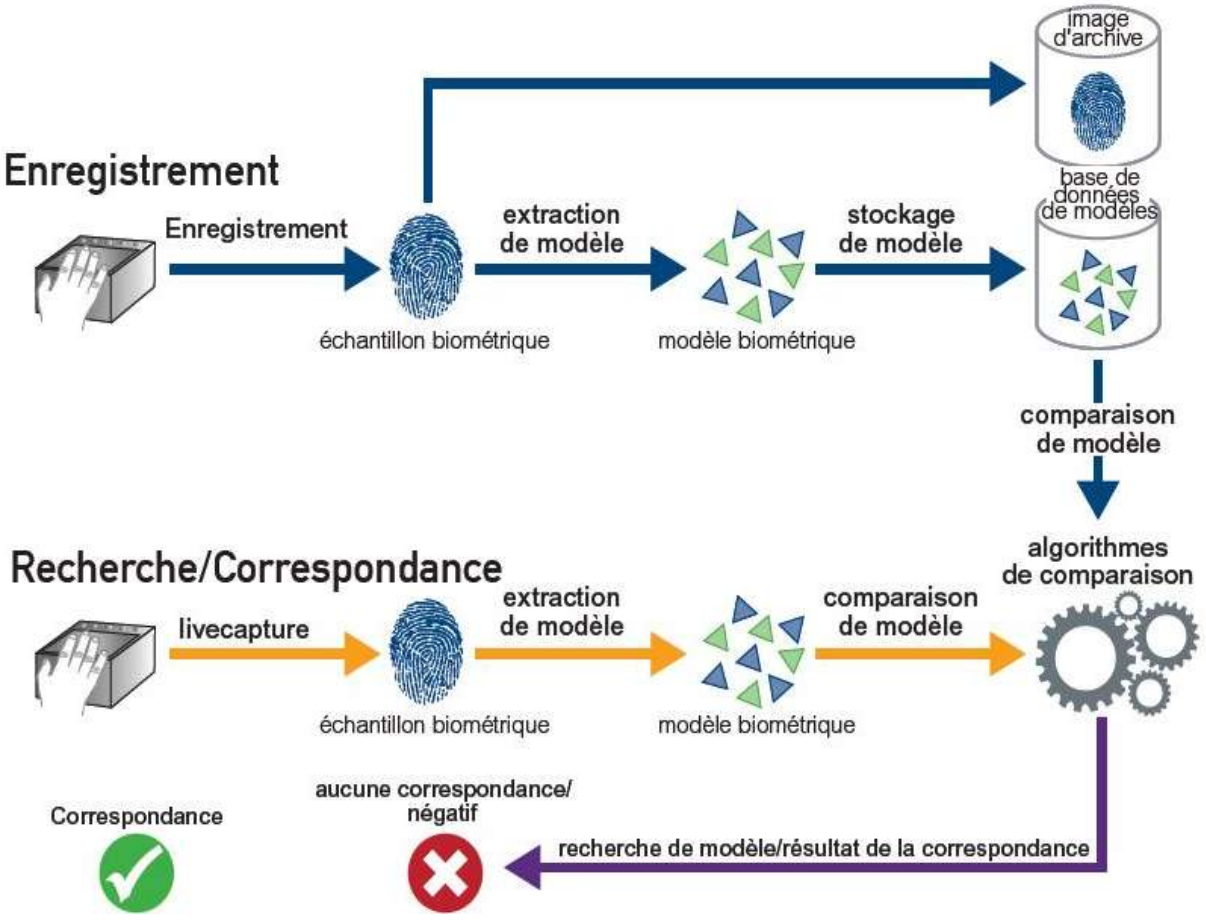
# VI.3. MODE DE FONCTIONNEMENT

Le processus biométrique se déroule en trois étapes :

1. **Enregistrement** - l'utilisateur donne éventuellement un identifiant (*PIN*), puis utilise le capteur. Le système vérifie la qualité de la mesure, en extrait les points d'intérêt, et place la référence (*template*) dans sa base de données.

2. **Vérification** - l'utilisateur fournit son PIN et se présente au capteur. Le système cherche l'entrée correspondant au PIN dans sa base de données et en extrait la référence associée. Le système compare la mesure et le template. Lorsqu'elle est possible, cette phase est la plus rapide. On parle de « *positive recognition* ».

3. **Identification** - L'utilisateur se présente devant le capteur, sans fournir de code PIN. Le système extrait les points d'intérêt de la mesure et cherche dans sa base de données s'il y a une correspondance. En conséquence, cette phase est plus lente que la précédente puisqu'il est plus rapide d'extraire une donnée en vue de la comparer que de comparer tous les templates. On parle de « *negative recognition* ».



## VI.4. MESURES DES PERFORMANCES

L'inconvénient majeur d'un système biométrique est qu'il ne permet pas une authentification maximale à 100%. En effet, les mesures se basent sur des propriétés physiques, elles peuvent se modifier avec le temps *(âge, accident, blessure, ...)*. On parlera plutôt d'identification. L'objectif pour les créateurs de tels systèmes n'est donc pas la sécurité absolue, mais un taux de certitude suffisant. Les performances s'expriment de plusieurs manières :

- **T.F.R.** (ou FRR, pour False Rejection Rate, ou FNMR, False Non-Match Rate) : Taux de Faux Rejets : nombre de personnes rejetées par erreur.
- **T.F.A.** (ou FAR, pour False Acceptation Rate, ou FMR, pour False Match Rate): Taux de Fausses Acceptations : nombre de personnes ayant été acceptées alors que cela n'aurait pas du être le cas.
- **E.E.R.** (x, t): Equal Error Rate. Le seuil représentant le niveau d'erreurs acceptées.

En ce sens, un système sera fonctionnel lorsque son *T.F.R*. sera faible, et un système sera sûr lorsque son *T.F.A.* sera faible. L'objectif sera de minimiser ces deux valeurs, comme l'illustre la figure ci-haut. Le x (*EER*) représente la marge d'erreur autorisée. Plus il est grand, moins le système est sûr, mais plus on le réduit, moins le système est utilisable pour identifier des personnes. La recherche tente de limiter cet EER afin d'obtenir simultanément des *TFA* et *TFR* acceptables.

## VI.5. TECHNIQUES BIOMÉTRIQUES

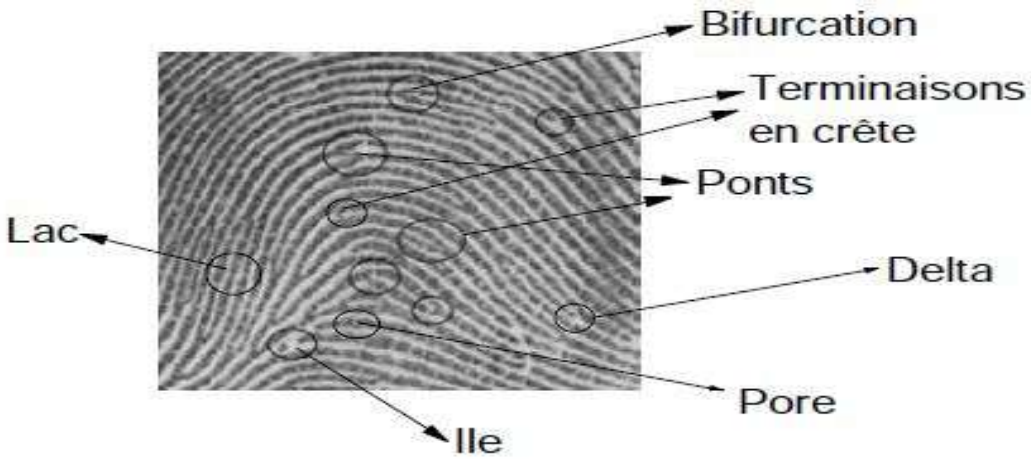
En informatique ; les techniques biométriques se subdivisent en trois grandes (+ 1 mixtes) catégories des techniques aussi appelée « *Moyens biométriques* » :

## VI.5.1. TECHNIQUES BIOMETRIQUES PHYSIQUES

### VI.5.1.1. LES EMPREINTES DIGITALES (FINGER-SCAN)

Elle est une des plus anciennes techniques d'identification, datant du début du XXe siècle. Cette technique prend ses racines dans la découverte de la permanence des dessins de la naissance à la mort, son caractère individuel et son inaltérabilité par le chercheur britannique Galton. Galton a défini la notion de minuties des empreintes digitales comme étant l'arrangement particulier des lignes des empreintes. Ces lignes forment des points caractéristiques permettant d'identifier de manière unique un individu. Ces points peuvent être des arrêts de lignes, des bifurcations, ou encore des îlots.

Les minuties : codifiées à la fin des années 1800 en « caractéristiques de Galton [19]», les minuties sont composées, de façon rudimentaire, de terminaisons en crêtes, soit le point où la crête s'arrête, et de bifurcations, soit le point où la crête se divise en deux. Le noyau est le point intérieur, situé en général au milieu de l'empreinte. Il sert souvent de point de repère pour situer les autres minuties. D'autres termes sont également rencontrés : le lac, l'île, le pont, le croisement, le delta, la vallée, le pore… Notons que dans l'analyse des minuties, une douzaine de variables doivent être prises en compte.




***Type des minuties***

Il sied de dire qu'il existe de nombreux types de capteurs pour obtenir l'image et les minuties associées :

- **La technique Optique** *(avec ou sans contact)* : Le capteur utilise un CMOS (*anciennement un CCD : Charged Coupled Device*). Ici, le scan est réalisé avec ou sans contact (*doigt posé sur le capteur ou non*). Si aucun contact n'a lieu, la lecture des empreintes a lieu par réflexion.

| AVANTAGES | INCONVENIENTS |
|---|---|
| * Son ancienneté et sa mise à l'épreuve.<br>* Sa résistance aux changements de température, jusqu'à un certain point.<br>* Son coût abordable.<br>* Sa capacité à fournir des | * Il est possible que l'empreinte d'utilisateurs précédents reste latente, d'où une possibilité de dégradation de l'image par surimpression.<br>* Apparition possible de rayures sur la fenêtre.<br>* D'autre part, le dispositif CCD peut s'user avec le temps et devenir moins |

[19] Du nom de Sir Francis Galton.

| | |
|---|---|
| résolutions de plus de 500 dpi. | fiable.<br><br>* Problèmes de contrastes (doigt propre et sec devient trop clair tandis qu'un doigt humide et recouvert d'un film gras devient très foncé), problème résolu grâce au film liquide mais système mal accepté. (mouille le doigt) |

- La technique *Par balayage ou Silicium* : le doigt défile sur le capteur et l'image est reconstruite au fur et à mesure de façon logicielle. Cette technique est apparue à la fin des années 90. Le doigt est placé sur un capteur CMDS. L'image est transférée à un convertisseur analogique-numérique, l'intégration se faisant en une seule puce. Cette technique produit des images de meilleure qualité avec une surface de contact moindre que pour la technique optique. Les données fournies sont très détaillées. Elle possède une bonne résistance dans des conditions non-optimales. Cette technique est adaptée à un développement de masse, notamment par ses coûts réduits.
- *Par capacité électrique* : on mesure la capacité électrique existante entre la peau et le capteur. Elle varie comme l'inverse de la distance entre le doigt et le capteur. Ainsi, en présence d'une crête, la capacité électrique sera plus importante (*car la distance est plus courte à cet endroit, puisque le doigt est posé sur le capteur*).
- *Par ultrasons* : cette technique permet d'éviter le problème des doigts salis qui ne permettrait pas une bonne réflexion de la lumière.
- *Par transmission de lumière* : le capteur mesure l'intensité d'un faisceau traversant le doigt. La technique d'extraction des empreintes porte le nom d'EDR : *Empreinte Digitale Réduite*.

## A. PRINCIPE DE FONCTIONNEMENT

L'authentification par les empreintes digitales repose sur la concordance entre le fichier d'enregistrement, ou « *signature* », obtenu lors de l'enrôlement et le fichier obtenu lors de l'authentification. Ces deux fonctions se décomposent chacune en plusieurs étapes :

- **Enrôlement**

  → Capture de l'image de l'empreinte. Les données d'un doigt sont en principe suffisantes à l'enrôlement, mais la plupart des systèmes enregistrent au moins deux doigts (*un par main par exemple*) pour parer l'indisponibilité résultant de petites blessures.

→ Numérisation de l'image afin d'extraire les minuties, ou éléments caractéristiques.

→ Enregistrement sur un support. (carte à puce, disque dur…).

- **Authentification**

→ Capture de l'image de l'empreinte.

→ Numérisation de l'image afin d'extraire les minuties, ou éléments caractéristiques.

→ Comparaison entre l'échantillon et le gabarit « signature ».

→ Prise de décision.

Lors de la capture de l'image, celle-ci est toujours constituée à partir des points de contact du doigt sur le capteur.

- **Etapes de traitement**

→ Lorsque la capture de l'image est réalisée, elle doit être convertie dans un format approprié. L'extraction des minuties est réalisée grâce à différents algorithmes. Il s'agit ensuite par une technique mathématique (*segmentation*) d'éliminer les informations non utiles au système : niveau de bruit trop élevé (*image sale, doigt mal placé*).

→ L'image est numérisée. Afin de localiser précisément les terminaisons et les bifurcations, les crêtes sont affinées de 5 à 8 pixels à 1 pixel.

A ce stade, l'image a des distorsions et de fausses minuties, qui peuvent être dues à des cicatrices, de la sueur, un défaut de propreté du doigt comme du capteur. Les minuties vont être filtrées afin de ne conserver que les plus fiables. Les avis divergent sur le rapport de proportion entre minuties extraites pour l'enrôlement et minuties suffisamment fiables pour la vérification. A partir de 31 minuties extraites, seulement 10 pourront correspondre lors de l'authentification. A titre informatif, une empreinte numérisée occupe en moyenne entre 250 et 1000 octets.

| AVANTAGES | INCOVENIENTS |
|---|---|
| ▪ Connaissance avancée du grand public, ce qui la rend moins intrusive ; | ▪ L'aspect « *policier* » donne un caractère négatif à l'identification ; |
| ▪ Petite taille des nouveaux types de lecteurs permet une meilleure intégration dans de nombreuses applications ; | ▪ Il est difficile de lire les données en présence de doigts blessés ou sales, selon la technique employée ; |

| | |
|---|---|
| ▪ Lecteurs biométriques à un moindre coût ;<br>▪ Le traitement des données est relativement rapide ;<br>▪ T.F.R. et T.F.A. assez faibles *(si le nombre de points caractéristiques est assez élevé*). | ▪ L'utilisateur doit poser correctement son doigt sur le capteur, au risque de se voir rejeté ;<br>▪ Le lecteur est exposé aux dégradations dues aux passages réguliers des doigts et aux pressions exercées lorsqu'un contact est nécessaire ;<br>▪ Il existe des techniques de contournement qui ne sont pas compliquées à mettre en œuvre (*ex : gummy fingers*). |

## B. APPLICATIONS

- Les domaines d'applications sont très larges et variés : accès au PC, GSM, contrôle du démarrage d'un véhicule, accès à certains locaux, accès au coffre-fort, ...
- Par contre, dans le domaine judiciaire, entre huit et vingt points caractéristiques sont suffisantes pour une identification.

## VI.5.1.2. LA FORME DE LA MAIN (HAND-SCAN)

La silhouette de la main est également une caractéristique propre à chaque individu. Les paramètres pris en compte (*jusqu'à 90*) peuvent être par exemple la longueur des doigts, la forme des articulations, ou encore leur épaisseur (*dans une vue à trois dimensions*). Le scanner utilisé est généralement à infra rouges, et possède des guides afin de positionner correctement les doigts de l'individu.

| AVANTAGES | INCOVENIENTS |
|---|---|
| ▪ C'est une technique simple à utiliser ;<br>▪ Elle est également très connue du grand public par l'intermédiaire du cinéma ;<br>▪ Le résultat est indépendant de l'humidité des doigts et de souillures éventuelles car il n'y a pas de contact direct avec le capteur ou une fenêtre, donc pas de risque d'encrassement ; | ▪ La mise en place d'un tel système est parfois impossible au vu de la place nécessaire (il est impossible d'utiliser un tel lecteur dans une voiture, ou pour un téléphone portable ...) ;<br>▪ Ce moyen biométrique est sujet aux modifications de forme dues au vieillissement de l'individu ou à l'accident ;<br>▪ Système encombrant ;<br>▪ Risque élevé du taux de fausses acceptations et faux rejets, par exemple à cause d'une |

| | |
|---|---|
| ▪ Facilité de l'enrôlement du point de vue de l'usager et bonne acceptation psychologique ;<br>▪ Faible volume de stockage par fichier. | blessure ou pour les jumeaux ou les membres d'une même famille ;<br>▪ Cette technique n'a pas évoluée depuis plusieurs années ;<br>▪ Le lecteur est plus cher que pour les autres types de capture de données physiques. |

## A. APPLICATIONS

- Utilisée souvent dans les contrôles d'accès aux locaux.

Pour la capture de l'image, la personne pose sa main sur une platine où les emplacements du pouce, de l'index et du majeur sont matérialisés. Une caméra CCD (*Charged Coupled Device / en français : DTC : Dispositif à Transfert de Charge*) prend l'image, reliée à un lecteur où sont enregistrées les informations. Ce lecteur inclut des logiciels de traitement et de codage. Quatre vingt dix caractéristiques sont examinées parmi lesquelles la forme tridimensionnelle de la main, la longueur et la largeur des doigts ainsi que la forme des articulations, et constituent un fichier d'environ neuf octets de mémoire. Cette technique, très répandue aux USA.

### VI.5.1.3. LE VISAGE (FACIAL-SCAN)

Le visage possède également de nombreuses caractéristiques pouvant déterminer une personne. On peut citer la forme des yeux, du nez, de la bouche, ou encore le creusement des joues. L'une des premières difficultés, jadis, était de tenir compte des éléments additionnels physionomiques tels que la présence de lunettes, d'une barbe, ou du maquillage. Aujourd'hui, c'est un peu moins vrai, les techniques ayant fortement évolué. Il faut aussi prendre garde aux conditions d'éclairage qui peuvent provoquer des ombres. Il y a plusieurs techniques utilisées au niveau de reconnaissance faciale telles que :

- **Eigenface** (*développée à MIT en 1987*) : le visage est décomposé en plusieurs images mettant chacune en évidence une caractéristique du visage en question. Ces caractéristiques sont définies par une série de valeurs (vecteurs propres, valeurs propres et matrice de covariance) permettant d'établir un ensemble de statistiques en vue d'identifier les individus avec plus ou moins de précision.
- **Feature Analysis** : basée sur la méthode *Eigenface*, elle calcule certaines distances entre les éléments, les positions relatives de ces éléments. Cette méthode semble plus souple pour l'identification. En effet, en cas d'inclinaison de la tête par exemple, la première méthode ne reconnaitra pas le visage, alors que le calcul des positions relatives permettra l'identification de l'individu.

- **Etudes des expressions et postures** : Via les réseaux de neurones, ce système permet, à partir d'un nombre réduit de données sources, de déduire un nombre impressionnant d'images qui pourront par la suite être comparées à la saisie du visage de l'individu désirant être identifié. La figure ci-dessous illustre les possibilités de cette technique.
- **Etude de la texture de la peau** : avec une précision assez grande, cette technique permet d'en étudier certaines caractéristiques (pores, taches, défauts de peau, etc.) ;
- **Etude 3D** : Elle permet d'analyser la forme des éléments du visage tels que le creusement des joues, la profondeur des orbites, etc.

| AVANTAGES | INCOVENIENTS |
|---|---|
| ▪ Technique peu coûteuse, peu encombrante ;<br>▪ Absence de contact avec le capteur, méthode non intrusive pour la personne ;<br>▪ pas de risques pour la santé.<br>▪ Technique non invasive ;<br>▪ User-friendly ;<br>▪ Une des techniques les plus développées après les empreintes digitales ;<br>▪ Technique permettant de sécuriser des organisations publiques de grande envergure. | ▪ Les vrais jumeaux ne sont pas différenciés ;<br>▪ Psychologiquement, certaines personnes rejettent leur image photographique (refus de son image, ajout d'accessoires, rôle, religion, critique de la qualité de la caméra, etc.). L'image est considérée comme trop personnelle pour être utilisée ;<br>▪ En tant que contrôle d'accès, le visage n'est pas, traditionnellement, reconnu comme un mécanisme fiable d'authentification. (Peut être dupé par l'utilisation de maquillage ou d'un masque en silicone) ;<br>▪ Dans l'état des systèmes actuels, technique trop sensible au changement d'éclairage, changement d'échelle (*taille du visage ou distance de la caméra*), présence d'arrière plan non stationnaire, changement de position lors de l'acquisition de l'image (*inclinaison de la tête ou expression*) ;<br>▪ Tout élément tel que lunettes de soleil, chapeau, moustache, barbe, percing, blessure peut causer des anomalies avec des systèmes d'identification du visage. |

## A. APPLICATIONS

- Entrées de casino ;
- Hall d'aéroports.

## VI.5.1.4. L'IRIS (IRIS-SCAN)

Dès 1950, il est fait mention de l'utilisation de l'iris comme moyen d'authentification, mais les travaux de *J. Daugmann* de 1980 basés sur les ondelettes de Gabor v3ont conduire à son développement. Il a été démontré que la probabilité de trouver deux iris identiques est inférieur à l'inverse du nombre d'humains ayant vécu sur terre. Le traitement relativement rapide exige que la personne soit très proche de l'objectif qui doit être un objectif macro. Le traitement s'effectue en trois phases :

- Recherche de la position de l'iris dans l'image de l'œil ;
- Extraction des paramètres caractéristiques ;
- Comparaison avec les éléments connus.

D'emblée, il faut distinguer clairement l'iris et la rétine. Il sied de signaler que chaque œil est unique. Dans l'iris de l'œil, il est possible de compter plus de 200 paramètres indépendants. Dès les années 80, des ophtalmologues ont remarqué que si la couleur de l'iris peut varier à travers le temps, il n'en est pas de même pour le motif qui reste constant.

La capture se fait ici par la prise de vue, souvent par infrarouge pour éviter la dilatation de la pupille. On réalise ensuite un aplatissement de l'image, et après traitements mathématiques (*ondelettes de Gabor*), on obtient un code qui pourra prendre place dans une base de données. Par contre, cette méthode biométrique nécessite un nombre élevé de prises de vue. En effet, ici, non seulement la distance entre le capteur et l'œil peut varier, mais également la position de la tête, l'ouverture de la paupière, l'éventuel maquillage des cils masquant une partie du motif, etc.

| AVANTAGES | INCOVENIENTS |
|---|---|
| ▪ L'iris ne subit pas de modification à travers le temps et donc Fiable ;<br>▪ Grande quantité d'information présente dans l'iris ;<br>▪ Le capteur est moins exposé qu'un capteur tactile. | ▪ Le motif de l'iris peut être photographié en vue d'une usurpation. Tout dépend des mesures de sécurité prises (*utilisation d'infrarouge*) ;<br>▪ Caractère plus intrusif que l'empreinte ou que la main. |

## A. APPLICATIONS

- Identification dans les distributeurs de billets de banque ;
- Contrôle d'accès aux locaux et machines.

## VI.5.1.5. LA RÉTINE (RETINA-SCAN)

La rétine est composée en grande partie d'un ensemble de vaisseaux sanguins. La complexité du réseau sanguin ainsi constitué permet une identification unique de l'individu. Cette technique est plus ancienne que la saisie de l'iris, mais bien moins utilisée. Alors que l'aspect des vaisseaux change avec l'âge, leur position ne se modifie pas. Pour effectuer la saisie, il faut éclairer le fond de l'œil. On pourra alors opérer une cartographie du réseau et ainsi la stocker dans une base de données.

| AVANTAGES | INCOVENIENTS |
|---|---|
| ▪ L'empreinte rétinienne est bien moins exposée aux blessures que les empreintes digitales ;<br>▪ Les T.F.R. et T.F.A. sont faibles ;<br>▪ On peut mesurer jusqu'à 400 points caractéristiques, contre seulement 30 à 40 pour une empreinte digitale. | ▪ Souvent mal accepté par le public, en raison de l'aspect sensible de l'organe. |

### A. APPLICATIONS

- Utiliser dans certains distributeurs de billets automatiques ;
- Utiliser pour le contrôle d'accès aux locaux à haute sécurité.

## VI.5.2. LES TECHNIQUES BIOMÉTRIQUES COMPORTEMENTALES

### VI.5.2.1. LA RECONNAISSANCE VOCALE (VOICE-SCAN)

La reconnaissance de la voix n'est pas intrusive pour la personne et n'exige aucun contact physique avec le lecteur du système. Le logiciel de reconnaissance peut être centralisé et la voix transmise par le réseau, d'où un impact de réduction des coûts. Le dispositif nécessite un micro en source de capture. Les systèmes d'identification de la voix sont basés sur les caractéristiques de voix, uniques pour chaque individu. Ces caractéristiques de la parole sont constituées par une combinaison des facteurs comportementaux (*vitesse, rythme, etc.…)* et physiologiques. (*Tonalité, âge, sexe, fréquence, accent, harmoniques, …).*

Il est à savoir que chaque individu possède une voix unique caractérisée par une fréquence, une intensité et une tonalité. Bien que deux voix peuvent sembler similaires pour l'oreille humaine, le traitement informatique permet de les isoler. Les prises sont réalisées à partir d'un simple microphone. On distingue deux types de systèmes à reconnaissance vocale :

- **Text Dependent System** - l'identification se fait sur des termes déterminés. Il existe 4 types de systèmes dépendants du texte :

  → *A texte suggéré (text-prompted)* : lors de chaque session d'identification, et pour chaque utilisateur, le système affiche un texte aléatoire.
  → *A traits phonétiques (speech event dependent)* : certains traits phonétiques doivent être mis en avant dans le texte prononcé par l'individu.
  → *A vocabulaire limité (vocabulary dependent)* : l'individu doit prononcer des mots issus d'un langage limité prédéfini (par exemple, une suite de chiffres) ;
  → *A texte personnalisé (user-specific text dependent)* : l'utilisateur possède un mot de passe, ou une passphrase.

- **Text Independent System (ou Free-Text)** - C'est le fait que l'on laisse l'utilisateur parler librement. La sécurité est bien entendu plus forte dans le cas d'une identification par texte dépendant qu'indépendant. La phase d'apprentissage nécessite souvent plusieurs phrases afin de prendre en compte les différentes intonations de la personne. Après l'acquisition, le signal est découpé en unités qui peuvent être soit des mots, soit des phonèmes. Cette technique porte aussi le nom de *(« AAL : Authentification Automatique du Locuteur »).*

Pour être stockée, la voix est numérisée puis segmentée par unités échantillonnées. Les méthodes sont basées sur des algorithmes mathématiques (Shannon). Les systèmes d'identification de la voix utilisent soit un texte libre, soit un texte imposé, les mots devant être lus devant un micro.

| AVANTAGES | INCOVENIENTS |
|---|---|
| ▪ Bien accepté par les utilisateurs, car il suffit de parler ;<br>▪ Il est facile de protéger le lecteur (*par l'intermédiaire d'une grille, dans le cas d'un parlophone*) ;<br>▪ Disponible via le réseau téléphonique ; | ▪ C'est un système sensible à l'état physique de l'individu selon son état (stress, fatigue, maladie, . . .) ; selon ses émotions (joie, peine, tristesse,. . .) ; et selon son âge ;<br>▪ Ce système peut être compromis par l'intermédiaire d'un enregistrement de bonne qualité. |

| | |
|---|---|
| ▪ Les imitateurs utilisent les caractéristiques vocales sensibles au système auditif humain, mais ne sont pas capables de récréer les harmoniques de la voix, servant de base à l'identification. Il est quasi impossible d'imiter la voix stockée dans la base de données ;<br>▪ Non intrusif. | ▪ Ce système est sensible aux bruits parasites (*environnement, usure du micro*) ;<br>▪ L'utilisation d'un micro nécessite un dispositif adapté présent sur l'environnement ;<br>▪ Sensibilité à l'état physique et émotionnel d'un individu ;<br>▪ Sensibilité aux conditions d'enregistrement du signal de parole : bruit ambiant, parasites, qualité du microphone utilisé, qualité de l'équipement, lignes de transmission ;<br>▪ Fraude possible en utilisant un enregistrement de la voix de la personne autorisée, facilitée dans le cas de système basé sur la lecture d'un texte fixe. |

## A. APPLICATIONS

- Utilisé dans la reconnaissance téléphonique ;
- Identification des propriétaires dans des immeubles à appartements, dont les lecteurs peuvent être protégés par une grille.

**Remarque** : Les inconvénients signalés montrent que ce système est vulnérable et doit être utilisé couplé avec un système d'identification (*lecteur de badges*).

## VI.5.2.2. LA DYNAMIQUE DE FRAPPE (KEYSTROKE-SCAN)

Cette reconnaissance identifie les personnes selon leur manière de taper sur un clavier. On parle dans ce cas précis de solution biométrique essentiellement logicielle, car elle consiste uniquement en un relevé de données basées sur la dynamique de frappe des utilisateurs. Par dynamique de frappe, on entend le temps utilisé entre deux frappes sur la même touche (*flight time*), le temps de pression sur chaque touche (*Dwell Time*), ou encore le temps pour taper un mot donné.

Lors de la phase d'apprentissage, il sera demandé à l'utilisateur de taper un certain nombre de fois un mot de passe ou une passphrase, et un algorithme sera chargé de moyenner les temps relevés. Par la suite, et selon le niveau de ressemblance demandé, les utilisateurs seront acceptés ou refusés.

La dynamique de la frappe au clavier est caractéristique de l'individu, c'est en quelque sorte la transposition de la graphologie aux moyens électroniques. Les paramètres suivants sont généralement pris en compte :

- Vitesse de frappe ;
- Suite de lettres ;
- Mesure des temps de frappe ;
- Pause entre chaque mot ;
- Reconnaissance de mot(s) précis.

| AVANTAGES | INCOVENIENTS |
|---|---|
| ▪ Pas de système hardware, seul le logiciel suffit, ce qui permet de réaliser des économies substantielles ;<br>▪ Très pratique lorsque le nombre d'utilisateurs est élevé, de par sa simplicité de mise en place et d'apprentissage ;<br>▪ T.F.A. inférieur à 0.5% si le mot de passe dépasse 8 caractères. | ▪ Exige une attention au clavier utilisé (distinction d'emplacement des touches en QWERTY et en AZERTY). Les vitesses de frappe changeant d'un type de clavier à l'autre, un profil d'identification sera nécessaire pour chaque configuration. |

## A. APPLICATIONS

- Utilisé la plupart de fois comme complément de sécurité (*notamment pour des cartes à puces*).

## VI. 5.2.3. LA SIGNATURE DYNAMIQUE (SIGNATURE-SCAN)

Le principe premier de cette technique est d'identifier une personne à partir des parties fixes de sa signature. Car une signature, même si elle reste approximativement identique, possède des parties fixes et des parties variables. La différentiation de ces parties permettra l'identification. Celle-ci pourra également prendre d'autres paramètres en compte, tels que la pression exercée avec le stylo, la vitesse d'écriture, ou encore les accélérations.

Le système de signature dynamique se compose principalement d'une tablette digitale et de son crayon. La phase d'apprentissage nécessite quelques signatures. Par la suite, le logiciel en extraira les caractéristiques, et tentera d'en calculer les variantes possibles et acceptables.

| AVANTAGES | INCOVENIENTS |
|---|---|
| ▪ La sécurité contre la copie est plus forte que pour un simple mot de passe ;<br>▪ Faible coût du matériel nécessaire ;<br>▪ Moyen non intrusif qui exploite un geste naturel. | ▪ Dégradation du matériel avec le temps ;<br>▪ L'écriture change au cours de la vie de l'individu ;<br>▪ Dépendance de l'état physique de la personne. (âge, maladies,…) |

## A. APPLICATIONS

- E-Commerce : Validation de la signature avant toute transaction ;
- Gestion bancaire : toutes les opérations sont signées ;
- Sécurisation des sessions sur ordinateur : A la place d'un simple mot de passe, l'utilisateur doit soumettre sa signature, ce qui est beaucoup plus difficile à copier ;
- Verrouillage de fichiers : on protège les fichiers par signature, ce qui permettra de l'ouvrir, de le modifier, de l'imprimer, de façon sécurisée.

# VI.5.3. LES TECHNIQUES BIOMÉTRIQUES EXPÉRIMENTALES

## VI.5.3.1. LA THERMOGRAPHIE

Ce système fonctionne avec une caméra thermique qui réalise une photographie infrarouge du visage. En plus, une répartition la chaleur intervient. Cette répartition est propre à chaque individu, y compris les vrais jumeaux. Ce système reste cependant encore très coûteux.

## VI.5.3.2. L'OREILLE

Technique peu répandue, cette technique consiste à comparer les empreintes d'oreille que peuvent laisser certains individus dans le cadre d'un délit. Cette méthode est uniquement utilisée par la police, mais est admissible devant une cour de justice.

### VI.5.3.3. L'AND

C'est une technique sûre car elle ne se base pas sur des points caractéristiques, mais sur l'entièreté des données. Il donnerait des T.F.R. et T.F.A. nuls (*si on fait exceptions des vrais jumeaux monozygotes qui possèdent le même ADN*). Cependant, sa mise en place pose énormément de problèmes en raison de son caractère "intrusif". En effet, le relevé de cet ADN permettrait la création d'un répertoire des utilisateurs, et ouvrirait la porte à de nombreuses dérives. Il existe toutefois des bases de données réelles qui recensent l'ADN de détenus (*Ex: Angleterre, France*).

### VI.5.3.4. AUTRES

D'autres méthodes sont encore exploitables, telles que l'odeur, la dentition, les battements de cœur, la démarche ou encore les pores de la peau, ...

## VI.5.4. LES TECHNIQUES BIOMÉTRIQUES MULTIMODALES

Dans le cas où un seul moyen biométrique est mesuré, on parle de biométrie unimodale. Lorsque ces moyens sont associés, ou que plusieurs mesures distinctes sont réalisées pour l'individu, on parlera de biométrie multimodale. La figure ci-dessous illustre la biométrie multimodale. Ce type de biométrie offre de nombreux avantages tels que :

- Plus de fiabilité car les vérifications sont indépendantes (moins de faux positifs et faux négatifs) ;
- Plus de compatibilité : un moyen biométrique pris seul n'est jamais utilisable par 100% de la population ;
- Plus de sécurité : la multiplication des moyens rend plus compliquée la tâche du pirate (*plusieurs mesures à contourner*). Il est aussi possible de vérifier le caractère "réel" de l'utilisateur en interagissant avec lui, par exemple en testant plusieurs doigts dans un ordre précis inconnu à l'avance.

# VI.6. LES CONTRAINTES TECHNIQUES ET ORGANISATIONNELLES DE CONTROLE BIOMETRIQUE

La mise en place d'un système de contrôle d'accès biométrique doit prendre en compte des éléments propres au facteur humain pour que les contrôles fonctionnent efficacement. En particulier, il convient de prendre en compte les éléments suivants :

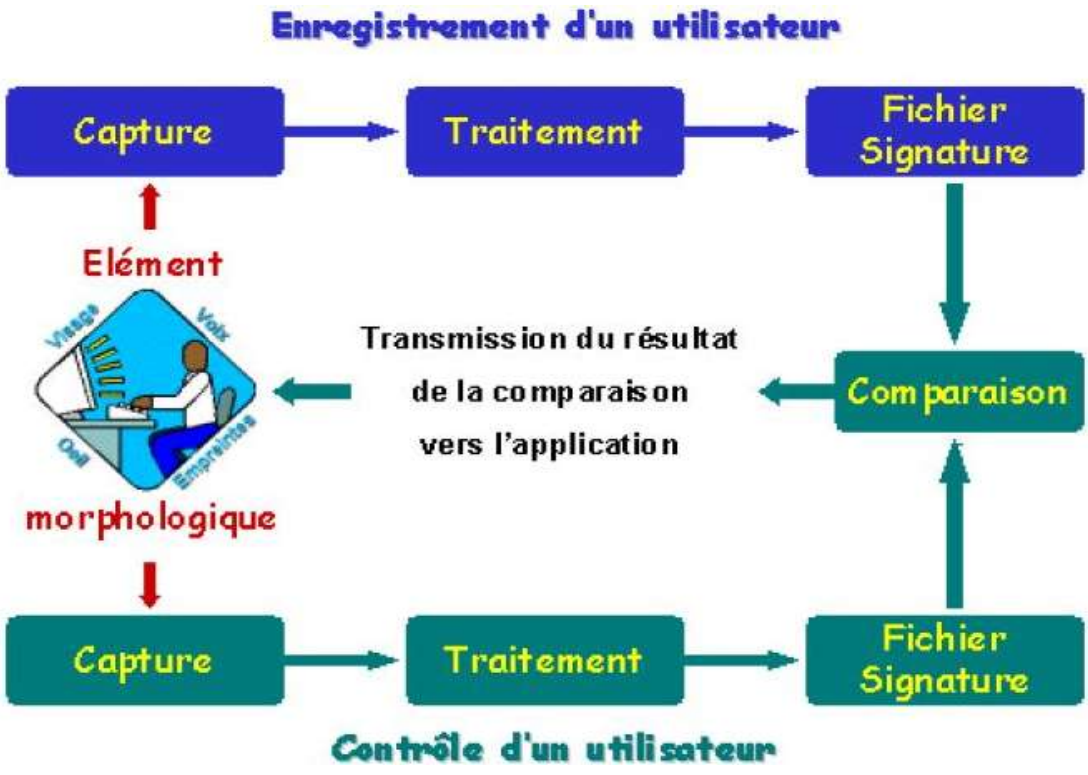
- Appareils communs à toute une population ;
- Capteur : problèmes d'hygiène ;
- Durée du contrôle.

Dans le cadre de cette étude l'ensemble des contrôles sont réalisés avec l'autorisation formelle de la personne.

## VI.6.1. LE CYCLE DE VIE D'UN PROCESSUS D'IDENTIFICATION BIOMÉTRIQUE

### VI. 6.1.1. PROCESSUS MACROSCOPIQUE

Le cycle macroscopique d'un processus d'identification biométrique se décompose en deux grandes étapes l'enrôlement et le contrôle.



- **Enrôlement** - L'enrôlement des personnes est la phase initiale de création du gabarit biométrique et de son stockage en liaison avec une identité déclarée. Les caractéristiques physiques sont transformées en un gabarit représentatif de la personne et propre au système de reconnaissance. Durant cette phase, des données additionnelles propres à la personne qui s'enrôle sont enregistrées comme par exemple ses nom et prénom et un identifiant personnel (PIN). Cette étape n'est effectuée qu'une seule fois.

- **Contrôle (vérification / identification)** - C'est l'action de contrôle des données d'une personne afin de procéder à la vérification de son identité déclarée ou, dans une investigation, à la recherche de l'identité de cette personne. Cette étape se déroule à chaque fois qu'une personne se présente devant le système.

  → *La vérification* consiste à confirmer l'identité prétendue d'une personne (authentifier) par le contrôle de ses caractéristiques physiques. Des données d'identification (nom, PIN, identifiant, etc.) sont présentées par la personne au système en même temps que ses caractéristiques physiques. C'est une comparaison « un-pour-un » dans laquelle le gabarit biométrique saisi est comparé au gabarit de référence correspondant dans une carte (ou autre dispositif physique personnel équivalent) ou dans une base de données. Dans le cas d'une carte, le gabarit saisi est directement rapproché du gabarit stocké dans la carte. La réponse est donnée par le terminal biométrique. Elle peut aussi être donnée par la carte moyennant l'emploi de protocoles cryptographiques garantissant l'authenticité de la réponse. Dans le cas du stockage des gabarits dans une base de données, l'accès au gabarit de référence

  → se fait par accès direct sur l'index de l'identifiant déclaré de la personne. une *L'identification* consiste à identifier une personne à l'aide de ses seules C'est comparaison « un pour-plusieurs » dans laquelle le gabarit biométrique saisi est comparé à tous les gabarits stockés dans une base de caractéristiques physiques au sein d'une population préalablement enregistrée. données.

A ces deux processus s'ajoutent souvent les deux processus suivants :

- **Le rafraîchissement ou actualisation** - le système biométrique peut périodiquement corriger le gabarit de référence lors d'un contrôle de façon à prendre en compte des évolutions des données personnelles de la personne, en particulier pour des systèmes comportementaux (*dynamique de signature*) ;

- **La fin de vie** - le gabarit et autres données de références propres à la personne sont détruites pour prendre en compte la suppression de la personne du système de contrôle.

## VI.6.1.2. PROCESSUS DÉTAILLÉS

Chaque système biométrique utilise des spécificités liées à la caractéristique physique analysée (*empreinte, iris, forme de la main, etc*.) et également liées à la technologie du système. Il est néanmoins possible d'identifier une série d'étapes ou de composantes génériques au processus. Ici, Nous ne présenterons que le macro-processus de contrôle (*hors enrôlement*) :

## A. COLLECTE [CAPTURE] DES DONNÉES D'IDENTIFICATION

C'est l'étape de saisie des données d'identification de la personne et en particulier de ses caractéristiques physiques par l'intermédiaire d'un capteur spécialisé correspondant à la caractéristique physique analysée. Les données saisies peuvent être selon le système :

- Les données sur l'identité prétendue (*PIN, nom, identifiant, etc*..). La collecte des données d'identité se fait à l'aide d'un lecteur approprié (*PIN pad, clavier, lecteur de carte, etc*.).
- Les données physiques personnelles (biologiques, morphologiques, comportementales). La collecte des données physiques se fait à l'aide d'un capteur approprié.

## B. SYSTÈME DE TRANSMISSION

Le système de transmission sert à transporter les données entre les différents sous-systèmes du système biométrique. En particulier il est possible que le sous-système de collecte et le sous-système de comparaison, voire celui de transformation soient distants l'un de l'autre. Le système de transmission qui peut être local (*interne à un boîtier) ou distant (liaison Ethernet, réseau ouvert, etc.)* doit être clairement identifié afin d'en assurer la protection contre des écoutes ou des manipulations de données.

Que ce soit dans le processus initial d'enrôlement ou dans celui de capture, il faut que l'identification et l'authentification soient garanties en terme d'intégrité. Le premier type de transmission intéresse la partie capteur : généralement cette partie capteur comprend une partie analogique (*empreinte, voix, morphologie signature*..) qui est digitalisée par la suite. Dans ce schéma nous trouvons des signaux optiques, sons, vidéo...

Ces signaux se trouvent généralement très proche du système de numérisation. Le problème est principalement un problème d'intégrité : cette intégrité est obtenue par un échantillonnage du signal adéquat. Le cas le plus général est la transmission entre le capteur et la base de données où se trouve le gabarit de référence. Si la transmission utilise un réseau (*local ou public*) des mesures de sécurité adéquates doivent être utilisées afin de protéger les données échangées en intégrité (des données et des flux). Les données véhiculées sont des données personnelles. A ce titre elles doivent également être protégées en confidentialité.

## C. TRANSFORMATION EN UN GABARIT BIOMETRIQUE

Le capteur des caractéristiques physiques transmet les données capturées à un système d'analyse qui a pour rôle de les transformer en un gabarit, selon un algorithme approprié à la caractéristique physique analysée. (*Empreinte, iris, forme de la main, etc*.).

## D. COMPARAISON À UNE RÉFÉRENCE

Le gabarit calculé doit être ensuite rapproché du gabarit de référence afin de vérifier l'identité de la personne ou de l'identifier. Ce processus se compose de :

- La recherche de la référence pour la comparaison : Via les données d'identification pour accès à la référence stockée pour une vérification ; et Via la comparaison directe du gabarit calculé à la valeur de référence stockée pour une identification ;

- La comparaison du gabarit calculé à une valeur de référence : à cette étape intervient une analyse qui est propre à la technologie du système développé. La comparaison fait généralement intervenir un calcul de score qui permet de considérer la comparaison en succès ou en échec selon que le score calculé est à l'intérieur ou à l'extérieur d'une plage d'acceptation.

## E. PRISE DE DÉCISION

Le sous-système de décision reçoit le résultat du score calculé de rapprochement au gabarit stocké. En fonction d'une politique de décision lié à une analyse de risque propre à l'application utilisatrice du système biométrique, le sous-système de décision décide des actions à suivre. Le sous-système de décision peut considérer la vérification ou l'identification :

- En succès et rendre une réponse positive au système applicatif utilisateur ;
- En échec complet et rendre une réponse négative au système applicatif utilisateur ;
- En indécision : dans certains cas de systèmes plus sophistiqués, le système peut demander une re-saisie voire la saisie d'une autre caractéristique physique *(ex : autre doigt pour l'analyse d'empreinte*).

En cas d'échec le système peut offrir la possibilité de recommencer le processus à l'étape de collecte ou alors comptabiliser le nombre de contrôle en échec de la même personne et décider d'un blocage du contrôle pour la personne considérée. La réponse est rendue au système applicatif qui utilise le système biométrique. C'est ensuite à l'application de décider des droits qu'elle accorde à la personne identifiée. En particulier, dans un contrôle par rapport à un fichier dit « négatif », le fait de ne pas être reconnu par le système biométrique peut être aussi important que l'inverse.

## VI.6.1.3. SYSTÈME DE STOCKAGE

Le système de stockage permet de maintenir le gabarit de référence des personnes enregistrées dans le système. Il permet de créer, modifier, supprimer des gabarits dans le système en relation avec des données d'identification de la personne.
Ce système est principalement lié à l'application utilisatrice. En particulier, selon que le but final de l'application est de vérifier l'identité d'une personne ou au contraire de l'identifier au sein d'une population, le mode de stockage sera différent.

*Dans le premier cas*, en raison de la réglementation sur la protection des personnes et des données personnelles, le système pourra mettre en œuvre un contrôle un-pour-un, le gabarit de référence étant stocké dans un support portable en possession de l'utilisateur. *Dans le second cas*, seul un stockage centralisé dans une base de données peut répondre au besoin. Là encore, la réglementation nationale peut ou non permettre une telle utilisation.

## VI.6.1.4. SYSTÈME DE RAFRAÎCHISSEMENT (D'ACTUALISATION)

La plupart des caractéristiques physiques sont stables dans le temps (empreintes, iris, forme de la main, etc.). En revanche, des caractéristiques physiques comportementales comme la dynamique de la signature, le rythme de la frappe au clavier, etc. peuvent évoluer dans le temps. Il peut alors être nécessaire d'actualiser le gabarit de référence de la personne selon une procédure propre au dispositif technologique. Si un tel rafraîchissement est opéré, le système devra garder des traces d'audit des mises à jour afin d'éviter toute fraude sur le système.

## VI.6.2. CHOIX DES PARAMÈTRES (SEUIL D'ACCEPTABILITÉ)

Quel que soit le procédé biométrique utilisé, la coïncidence à 100 % entre les deux fichiers signatures, celui établi lors de l'enrôlement et celui établi lors de l'authentification est impossible. La performance et la fiabilité d'un système s'expriment donc par le taux de faux rejets (T.F.R.) et le taux de fausses acceptations (T.F.A.). Un système émet un faux rejet lorsqu'il rejette par erreur un vrai utilisateur. A l'inverse, un système émettra une fausse acceptation en donnant accès à quelqu'un qui n'a pas de droit. Le seuil de décision doit être estimé en fonction du niveau de sécurité souhaité.

# CONCLUSION

Après cette longue excursion sur la sécurité informatique vue sous plusieurs allures, nous pouvons sitôt approvisionner l'hypothèse selon laquelle la sécurité à 100% n'existe et que le risque 0 est chimérique. En effet, ce que nous devons faire ; c'est de défendre habilement les ressources informatiques à notre disposition, car l'information est le principe réacteur d'une entreprise, et sa sécurité vaut son pesant d'or. C'est ainsi que pour sécuriser, il faut examiner 2 éléments dont : *Ce qu'il faut protéger* et *Comment le protéger.*

Ainsi, Assurer un niveau adéquat de sécurité, ne pas synonyme de surprotéger ce qui n'en vaut pas/plus la peine plutôt Protéger efficacement ce qui en vaut la peine en plaçant la sécurité au bon endroit dans l'administration d'une entreprise. Cela étant, ne jamais sous-estimer le facteur humain. Puisque, de tout temps, on a fait confiance à l'être humain pour conserver des informations. Rien ne sert d'avoir des protections techniques infaillibles si une personne interne à votre système permet à son insu à un attaquant de déjouer toutes ces protections. L'humain est une faille qu'il faut surveiller. Je vous recommande de rester attentif aux nouvelles technologies, aux mises à jour de sécurité des logiciels, et Ne pas se reposer sur une sécurité jugée bonne à un moment donné. La sécurité est omniprésente et nécessaire à tous les niveaux d'utilisation de l'information, de sa création à sa destruction. Nous devons nous rappeler toujours que: ***La sécurité est un compromis entre efficacité et convivialité***. Nous espérons que le support de ce cours de sécurité informatique et cryptologie vous sera utile dans votre cursus universitaire.

**YENDE RAPHAEL Grevisse, PhD.**
***Professeur associé***