الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي و البحث العلمي

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Relizane
Faculté des Sciences et Technologies
Département d'Informatique

Polycopié de cours

Sécurité Informatique

Troisième année Licence Systèmes Informatiques (SI)

Dr. BOUADJEMI Abdelkrim

Année universitaire : 2022/2023

Avant propos

Ce cours s'adresse particulièrement aux étudiants de troisième année Licence. Il peut aussi être utilisé par les étudiants du cycle Master comme support à leurs cours stratégie de la sécurité informatique.

Objectif du cours

L'objectif de ce cours est de permettre aux étudiants d'acquérir des connaissances et une base solide dans l'ingénierie de la sécurité et leurs donnera les fondements nécessaires afin de suivre d'autres modules en sécurité et en cryptographie dans la suite de leurs cursus universitaire.

A l'issue de ce cours, les étudiants seront capable de :

- Comprendre les concepts ainsi que les objectifs de la sécurité
- Découvrir les différentes menaces informatiques
- Utiliser efficacement les méthodes de défense
- Arborer / Assimiler le domaine de la cryptographie

Sommaire

Liste de figures

1. Sureté de fonctionnement (SdF)

1.1. Aperçu (Principaux concepts)

La sûreté de fonctionnement (SdF) est évoquée avec la révolution industrielle. L'idée de la SdF est d'atteindre l'objectif Zéro (zéro accident, zéro arrêt, zéro défaillance) par le déploiement des méthodes et des techniques permettant de :

- minimiser leur présence ;
- tolérer leur occurrence ;
- mesurer l'efficacité des techniques utilisées.

Afin de de répondre aux besoins de la SdF, il faudrait tester toutes les utilisations possibles d'un produit pendant une grande période ce qui est impossible à réaliser. La SdF est un domaine d'activité qui propose des moyens pour améliorer la fiabilité et la sûreté des systèmes avec des coûts raisonnables.

1.2. Définitions des SdF

SdF ou science de défaillances, est un domaine qui nécessite une connaissance intégrale du système comme les conditions d'utilisation, les risques extérieurs, les architectures fonctionnelle et matérielle, la structure des matériaux.

D'autres définitions stipule que :

La sûreté de fonctionnement (dependability, SdF) consiste à évaluer les risques potentiels, prévoir l'occurrence des défaillances et tenter de minimiser les conséquences des situations catastrophiques lorsqu'elles se présentent.

La sûreté de fonctionnement d'un système informatique est la propriété qui permet de placer une confiance justifiée dans le service qu'il délivre.

1.3. Taxonomie

La SdF manipule des concepts suivants :

- *Attributs* : points de vue pour évaluer la sûreté de fonctionnement ;
- *Entraves* : évènements qui peuvent affecter la sûreté de fonctionnement du système ;
- *Moyens* : moyens pour améliorer la sûreté de fonctionnement.

La figure 1 montre les différents concepts.

Figure 1. Arbre de SdF (1)

1.3.1. Attributs

Les attributs de la SdF sont appelés FDMS pour Fiabilité, Disponibilité, Maintenabilité et Sécurité ou RAMSS (Reliability, Availability, Maintainability, Safety, Security).

1.3.1.1. Fiabilité (Reliability):

Capacité d'un système à assurer la continuité de service

Définition : est l'aptitude d'un dispositif à accomplir une tâche requise dans des conditions données pendant une durée donnée.

1.3.1.2. Disponibilité (Availability) :

La disponibilité est le fait d'être prêt au service, c'est-à-dire capacité d'un système à être prêt à l'utilisation

Définition : la disponibilité est la capacité d'une entité d'accomplir une tâche requise dans des conditions données, à un instant donné ou pendant un intervalle de temps donné, en supposant que la fourniture des moyens extérieurs nécessaires soit assurée.

1.3.1.3. Maintenabilité (Maintainability)

Capacité d'effectuer des réparations et des évolutions, c'est-à-dire la capacité d'un système à se retourner dans un état de fonctionnement correct après modifications et réparations.

Définition : est la capacité d'un élément à être entretenu ou restauré pour remplir sa fonction requise lorsqu'il est entretenu dans des conditions spécifiées en utilisant des procédures et des moyens spécifiés pendant une période de temps spécifiée.

1.3.1.4. Sécurité

Innocuité : non-occurrence de défaillances catastrophiques

La sécurité est est une compétence à ne pas provoquer d'accidents catastrophiques.

Définition (Sécurité innocuité / Safety) : la sécurité innocuité est la capacité d'une entité à éviter des événements graves ou catastrophiques survenant dans certaines conditions.

1.3.2. Moyens

Les moyens sont des solutions aguerries pour casser les enchaînements :

Faute -> Erreur ->Défaillance et donc améliorer la fiabilité du système.

- **La prévention de fautes :** consiste à éviter d'éventuelles erreurs lors du développement du système. Ceci est réalisé grâce à l'utilisation de méthodologies de développement et de bonnes techniques de mise en œuvre ;
- **L'élimination de fautes :** peut être divisées en deux catégories : les exclusions pendant la phase de développement et pendant la phase d'utilisation. Pendant la phase de développement, l'idée est d'utiliser des techniques de vérification avancées pour détecter et éliminer les erreurs avant d'entrer en production. Les erreurs rencontrées lors de l'utilisation doivent être tenues à jour et corrigées lors des cycles de maintenance ;
- **La prévision de fautes :** consiste à prédire les fautes et leur impact sur le système ;
- **La tolérance aux fautes :** consiste à mettre en place des mécanismes qui assurent le service fourni par le système, même en présence de fautes. Un fonctionnement dégradé est autorisé.

La tolérance aux fautes repose sur l'utilisation de mécanismes de redondance, l'idée est de réaliser la même fonction par des moyens différents. On distingue plusieurs types de redondance :

- *Redondance homogène* : on réplique plusieurs composants identiques ;
- Redondance avec dissemblance : les sous-systèmes réalisent les mêmes fonctions mais sont différents (par exemple, plusieurs équipes de conception, matériel différent) ;
- *Redondance froide* : les composants sont activés quand ceux déjà actifs tombent en panne ;
- *Redondance chaude* : les composants tournent en parallèle et politique de prise de main ;
- *Redondance tiède* : les composants sont idle avant de prendre la main.

D'autres mécanismes existent comme les comparateurs ou les voteurs. L'idée est de récupérer plusieurs valeurs calculées par redondance et de déterminer quelle est la plus proche de la réalité.

1.3.3. Entraves

Les entraves (obstacles) sont des limitations qui affectent le système et peuvent affecter la sûreté de fonctionnement. Les entraves sont divisées en trois notions : les fautes, les erreurs et les défaillances qui s'enchaînent.

- La cause de l'erreur est une faute (par exemple un court-circuit sur un composant, une perturbation électromagnétique ou une faute de développement logiciel) ;
- La cause de la défaillance est une erreur affectant une partie de l'état du système (par exemple, une variable erronée) ;
- Une défaillance est la cessation de l'aptitude d'une entité à accomplir une fonction requise.

2. Sécurité informatique

Avec le développement des usages d'Internet, de plus en plus d'entreprises ouvrent leurs systèmes d'information à leurs partenaires et fournisseurs, il est impératif de contrôler les

accès et les droits des utilisateurs du SI. Il en va de même pour l'ouverture de l'accès des entreprises à Internet.

La sécurité informatique consiste à protéger les systèmes, les informations et les services contre les menaces accidentelles ou intentionnelles qui compromettent leur confidentialité, leur intégrité et leur disponibilité. Cela inclut toutes les technologies informatiques qui nous permettent de minimiser la possibilité de divulgation d'informations, de modification de données ou de dégradation de service.

Il faut qu'elle soit toujours protégée de manière appropriée contre :

- les accidents et inconsciences ;
- les différentes arnaques et attaques.

2.1. Définition

La sécurité informatique est l'ensemble des mesures, techniques, outils et ressources utilisés pour minimiser les vulnérabilités du système ou, dans la mesure du possible, pour protéger les systèmes contre les menaces accidentelles ou délibérées.

2.2. Critères de la sécurité

Les coûts de la sécurité peuvent être élevés. 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. La sécurité, quant à elle, est un compromis entre les coûts, les risques et les contraintes.

2.2.1. Risque

C'est la probabilité qu'une menace exploite une vulnérabilité. Autrement dit, c'est une possibilité qu'un fait dommageable se produise. (2)

On comprendra mieux le poids d'un risque en se fiant à la formule suivante :

$$Risque = \sum_{i} Ri = \sum_{i} (Mi * \sum_{j} Vij)$$

NB : en absence de menaces, le système ne court aucun risque, de même, en absence de vulnérabilités le système ne court aucun risque.

Risques accidentels

- Risques matériels accidentels ;
- Panne et dysfonctionnement de matériel ou de logiciel de base.

Risques d'erreur

- Erreur de saisie, de transmission ;
- Erreur d'exploitation ;
- Erreur de conception et de réalisation.

Risques de malveillance

- Fraude, Vol et sabotage ;

- Indiscrétion (espionnage industriel ou commercial..). (3)

2.2.2. Vulnérabilité

Une vulnérabilité est une erreur ou une faille dans un système informatique qui permet à un attaquant de compromettre la sécurité de ce système, c'est-à-dire a son fonctionnement normal, à la disponibilité, à la confidentialité et à l'intégrité des données. Ces vulnérabilités résultent de faiblesses dans la conception, la mise en œuvre ou l'utilisation des composants matériels ou logiciels du système.

2.2.3. Menace

Une menace est un danger qui existe indépendamment dans l'environnement du système. Il représente une série d'actions dans l'environnement du système qui peuvent entraîner une perte.

Il existe deux types de menaces : (3)

2.2.3.1. Les menaces intentionnelles

C'est l'ensemble des actions malveillantes qui constituent la plus grosse partie du risque et devrait être la cible principale des mesures de protection. Deux catégories sont définies ici :

- *Les Menaces passives* : elles ne modifient pas l'information et portent essentiellement sur la confidentialité des données ;
- *Les Menaces actives* : elles modifient le contenu de l'information ou le comportement des systèmes de traitement, elles portent sur l'intégrité des données.

2.2.3.2. Les menaces non intentionnelles (accidentelles)

Ces menaces involontaires sont liées à des pannes et des erreurs. Les menaces involontaires incluent l'oubli de sauvegarde, les erreurs de manipulation des informations et les erreurs de conception des applications.

Selon la norme ISO 7498-2, le risque est une *violation potentielle de la sécurité.*

2.3. Les objectifs de la sécurité

2.3.1. Disponibilité

La disponibilité concerne les services (ordinateurs, réseaux, périphériques, applications, etc.) et les informations (données, fichiers, etc.) doivent être accessibles aux personnes autorisées quand elles en ont besoin : le système doit donc rester fonctionnel malgré la survenue d'erreurs, malicieuses ou accidentelles. Cela signifie que les utilisateurs autorisés ont accès aux informations et aux ressources associées lorsqu'ils en ont besoin (pas d'accès non autorisé). De même, la disponibilité fait référence aux données qui ne doivent pas être supprimées ou rendues inaccessibles.

X.800 et RFC 2828 définissent la disponibilité comme une propriété d'un système ou d'une ressource système accessible et utilisée selon les besoins par les entités système autorisées conformément aux spécifications de performances du système. Diverses attaques peuvent entraîner une perte ou une réduction de la disponibilité. Certaines de ces attaques font l'objet de contre-mesures automatisées telles que l'authentification et le cryptage, tandis que d'autres nécessitent une action physique pour éviter ou restaurer la perte de disponibilité d'éléments du système distribué.

X.800 traite également la disponibilité comme une propriété d'appartenance à divers services de sécurité. Le service de disponibilité protège le système pour garantir la disponibilité. Ce service répond aux problèmes de sécurité causés par les attaques par déni de service. (4)

2.3.2. Confidentialité

La confidentialité est la protection des données transmises contre les attaques passives. Définit les limites de divulgation des informations. L'information n'appartient pas à tout le monde. Seules les personnes autorisées peuvent y accéder selon des conditions prédéfinies. Par conséquent, le système ne doit pas divulguer d'informations à des personnes qui ne sont pas autorisées à y accéder.

2.3.3. Intégrité

L'intégrité définit les restrictions sur les modifications apportées aux informations. H. Garantir l'exactitude et la fidélité des informations et des méthodes de traitement des données (3). Les services et informations (fichiers, messages, etc.) ne peuvent être modifiés que par les personnes autorisées (administrateurs, propriétaires, etc.). Par conséquent, les personnes qui n'ont pas les droits sur le système ne doivent pas le modifier. Il s'agit d'une propriété qui garantit que les informations ne peuvent changer que dans des conditions prédéfinies (selon des contraintes strictes).

Selon la norme ISO 7498-2, l‘intégrité est la prévention d’une modification non autorisée de l’information.

2.3.4. Non répudiation

La non-répudiation est la propriété qui assure que l'auteur d'un acte ne peut ensuite dénier l'avoir commis. Elle définit les contraintes sur la responsabilité des actions. Deux aspects spécifiques de la non-répudiation dans les transactions électroniques :

- **La preuve d'origine :** un message (une transaction) ne peut être dénié par son émetteur ;
- **La preuve de réception :** un récepteur ne peut ultérieurement nier avoir reçu un message. Exemple : un ordre s'il ne lui a pas plu de l'exécuter alors qu'il le devait juridiquement.

3. Les attaques informatiques

3.1. Qu’est ce qu’une attaque

Le piratage (hacking) est un ensemble de techniques informatiques, visant à exposer, de modifier, de désactiver, de détruire, de voler ou d'obtenir un accès non autorisé. Les attaques peuvent être locales (sur le même ordinateur, voir sur le même réseau) ou distantes (sur internet, par télécommunication). (5)

Internet Engineering Task Force définit l'attaque dans RFC 2828 (6) comme : « *Un assaut sur la sécurité du système qui découle d'une menace intelligente, c'est-à-dire d'un acte intelligent qui est une tentative délibérée (en particulier dans le sens d'une méthode ou d'une technique) pour échapper aux services de sécurité et violer la politique de sécurité d'un système*. »

Selon l'instruction CNSS n°4009 du 26 avril 2010 par le Comité des systèmes de sécurité nationale des États-Unis d'Amérique (7) définit une attaque comme suit : « *Toute activité malveillante qui tente de collecter, perturber, nier, dégrader ou détruire les ressources du système d'information ou l'information elle-même*. »

3.2. Le but de l'attaque

Le but du hacking est divers. Selon les individus (les "hackers"), on y retrouve :

- Vérification de la sécurisation d'un système ;
- Vol d'informations (fiches de paye...) ;
- Terrorisme ;
- Espionnage classique ou industriel ;
- Chantage ;
- Manifestation politique ;
- Par simple jeu, par défi ;
- Pour apprendre ;
- Etc.

3.3. Les différents types de pirates

En réalité il existe de nombreux types d'attaquants catégorisés selon leur expérience et selon leurs motivations :

- Les « **white hat hackers** », hackers au sens noble du terme, dont le but est d'aider à l'amélioration des systèmes et technologies informatiques, sont généralement à l'origine des principaux protocoles et outils informatiques que nous utilisons aujourd'hui ; Le courrier électronique est un des meilleurs exemples ;
- Les « **black hat hackers** », plus couramment appelés pirates, c'est-à-dire des personnes s'introduisant dans les systèmes informatiques dans un but nuisible ;

Les « **scripts kiddies** », sont de jeunes utilisateurs du réseau utilisant des programmes trouvés sur Internet, généralement de façon maladroite, pour vandaliser des systèmes informatiques afin de s'amuser ;

- Les « **phreakers** » sont des pirates s'intéressant au réseau téléphonique commuté (RTC) afin de téléphoner gratuitement grâce à des circuits électroniques (qualifiées de box, comme la blue box, ...) connectés à la ligne téléphonique dans le but d'en falsifier le fonctionnement ;
- Les « **carders** » s'attaquent principalement aux systèmes de cartes à puces (en particulier les cartes bancaires) pour en comprendre le fonctionnement et en exploiter les failles ;
- Les « **crackers** », sont des personnes dont le but est de créer des outils logiciels permettant d'attaquer des systèmes informatiques ou de casser les protections contre la copie des logiciels payants. Un « crack » est ainsi un programme créé exécutable chargé de modifier (patcher) le logiciel original afin d'en supprimer les protections ;
- Les **« hacktivistes »**, contraction de hackers et activistes que l'on peut traduire en cybermilitant ou cyberrésistant), sont des hackers dont la motivation est

principalement idéologique. Ce terme a été largement porté par la presse, aimant à véhiculer l'idée d'une communauté parallèle.

3.4. Les types d'attaques

Les hackers utilisent plusieurs techniques d'attaques. Ces attaques peuvent être regroupées en trois familles différentes : (5)

- Les attaques directes ;
- Les attaques indirectes par rebond ;
- Les attaques indirectes par réponses.

3.4.1. Les attaques directes

C'est l'attaque la plus simple. Les pirates attaquent les victimes directement depuis leur ordinateur. La plupart des "script kiddies" utilisent cette technique. En fait, les programmes de piratage qu'ils utilisent sont faiblement configurés et nombre de ces programmes envoient des paquets directement à la victime. (5)

Avec ce type d'attaque, il est plus probable que la source de l'attaque puisse être tracée tout en identifiant l'attaquant.

C'est la plus simple des attaques à réaliser : (8)

- Le hacker attaque directement sa victime à partir de son ordinateur par des scripts d'attaques faiblement paramétrable ;
- les programmes de hack qu'ils utilisent envoient directement les paquets à la victime ;
- Dans ce cas, il est possible en général de remonter à l'origine de l'attaque, identifiant par la même occasion l'identité de l'attaquant.

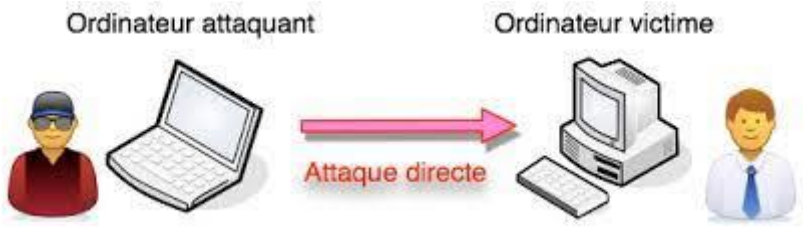


Figure 2. Attaque directe

3.4.2. Les attaques indirectes par rebond

Cette attaque est très prisée des hackers. En effet, le rebond a deux avantages : (5)

- Masquer l'identité (l'adresse IP) du hacker ;
- Éventuellement, utiliser les ressources de l'ordinateur intermédiaire car il est plus puissant (CPU, bande passante...) pour attaquer.

Le principe est simple, Les paquets d'attaque sont envoyés à l'ordinateur intermédiaire, qui répercute l'attaque vers la victime. D'où le terme de rebond.

Cela montre l'intérêt de protéger son réseau ou son ordinateur personnel, il est possible de se retrouver « complice » d'une attaque et en cas de plainte de la victime, la première personne interrogée sera le propriétaire de la machine ayant servi de rebond. (8)

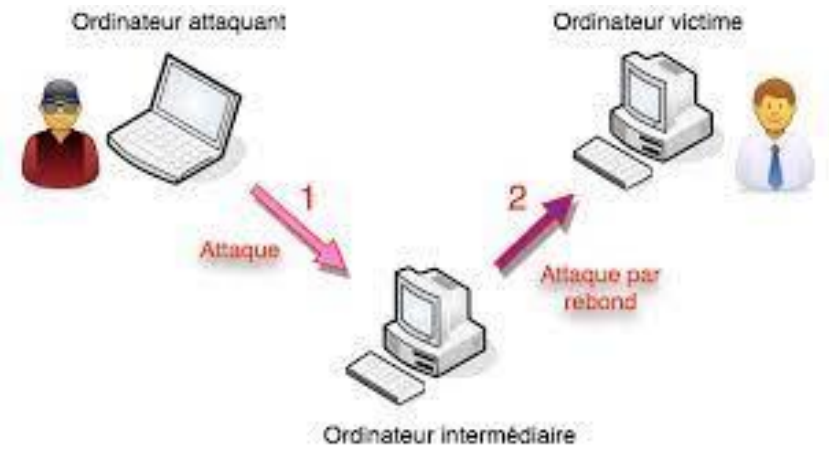


Figure 3. Attaque indirecte par rebond

3.4.3. Les attaques indirectes par réponse

Cette attaque est un dérivé de l'attaque par rebond. Du point de vue d'un pirate informatique, il a les mêmes avantages. Mais au lieu d'envoyer l'attaque à un ordinateur intermédiaire pour relais, l'attaquant envoie la requête. Et c'est cette réponse à la requête qui est envoyée à l'ordinateur de la victime. Là aussi, il n'est pas aisé de remonter à la source.

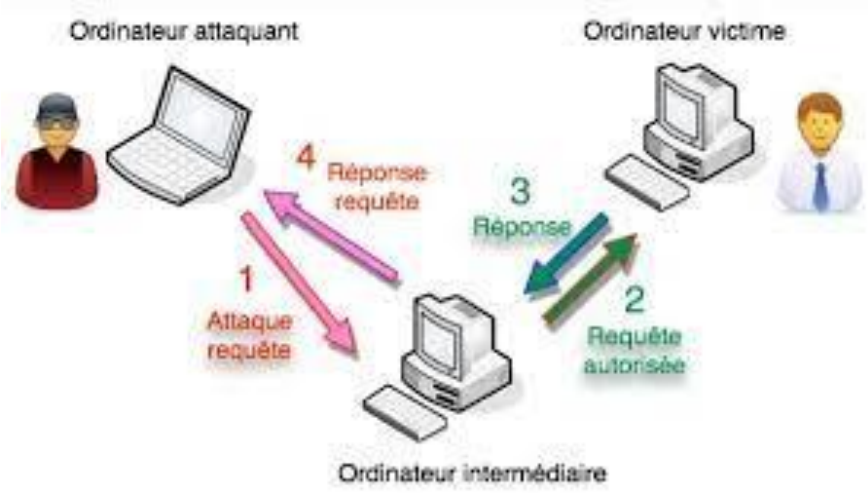


Figure 4. Attaque indirecte par réponse

3.5. Les catégories d'attaques

Les attaques portées à la sécurité des ordinateurs ou des réseaux peuvent être caractérisées en considérant le système comme un fournisseur d'informations. Il existe quatre catégories d'attaques : l'interruption, l'interception, la modification et la fabrication. (9)

3.5.1. Interruption

Les atouts du système sont détruits ou rendus indisponibles ou inutilisables. Il s'agit d'une attaque de disponibilité. Des exemples de ceci incluent la destruction du matériel (disques durs, etc.), la coupure des lignes de communication et la mise hors service des systèmes de gestion de fichiers.

3.5.2. Interception

Un tiers non autorisé accède aux actifs. C'est une atteinte à la confidentialité. Il peut s'agir d'une personne (vie privée), d'un programme ou d'un ordinateur. Des exemples de ceci

incluent l'écoute clandestine pour recueillir des données sur un réseau, ou la copie non autorisée de fichiers ou de programmes.

3.5.3. Modification

Un tiers non autorisé accède à l'actif et le modifie de sorte qu'il est (presque) indétectable (introuvable). Il s'agit d'une atteinte à l'intégrité. Des exemples de telles attaques incluent la modification des valeurs dans les fichiers de données, altérer un programme de façon à bouleverser son comportement ou la modification du contenu des messages envoyés sur le réseau.

3.5.4. Fabrication

Un tiers malhonnête insère un faux (contrefaçon) dans le système. C'est une atteinte à l'authenticité. Ils peuvent insérer (publier) de faux messages dans le réseau ou ajouter des enregistrements à des fichiers.

3.6. Taxonomie des attaques

Aujourd'hui, la vie est devenue plus confortable grâce à divers appareils numériques et à Internet. Internet a apporté des changements positifs, mais il pose également des défis majeurs en matière de protection des données. Cela conduit à des cyberattaques.

3.6.1. Les différentes attaques

3.6.1.1. Les attaques DoS et DDoS

Une attaque par déni de service (DoS) surcharge un système cible et le rend incapable de répondre aux demandes légitimes. Une attaque par déni de service distribué (DDoS) est similaire, mais implique plusieurs machines hôtes. Les attaques DDoS sont lancées à partir de divers ordinateurs hôtes infectés par des logiciels malveillants contrôlés par l'attaquant.

Ces attaques ne fournissent pas à l'attaquant un accès au système cible ni aucun avantage direct. Ils ne sont utilisés qu'à des fins de sabotage ou comme outil de diversion pour distraire les équipes de sécurité pendant que l'attaquant mène d'autres attaques. Les attaques DoS peuvent également être utilisées pour créer une vulnérabilité à d'autres types d'attaques. Une attaque DoS ou DDoS réussie nécessite souvent que le système soit mis hors ligne. Cela peut le rendre vulnérable à d'autres types d'attaques.

Les attaques DDoS ont fermé des sites comme Twitter, SoundCloud et Spotify, et même contre Amazon Web Services (AWS) qui s'est produit en février 2020

3.6.1.2. Les attaques Man in The Middle MITM

Une attaque l'homme au milieu (Man in the Middle - MITM), également connue sous le nom d'attaque d'écoute clandestine, fait référence à une vulnérabilité de cybersécurité qui permet à un attaquant d'écouter clandestinement les échanges de données entre deux personnes, réseaux ou ordinateurs. Un attaquant se trouve "au milieu" entre les deux parties et peut souvent intercepter les communications sans être détecté. Un attaquant pourrait également modifier le message avant de l'envoyer au destinataire prévu.

3.6.1.3. Les attaques Phishing

Une attaque de phishing se produit lorsqu'un acteur malveillant (escroc) tente d'obtenir des informations sensibles d'une cible à l'aide d'un e-mail, d'un SMS, d'un appel téléphonique ou d'un réseau social qui semble provenir d'une source fiable et légitime. Les attaques de phishing sont une combinaison d'ingénierie sociale et de technologie qui permettent aux

attaquants de "pêcher" l'accès à des zones restreintes en utilisant "l'appât" d'expéditeurs apparemment dignes de confiance.

3.6.1.4. Les attaques Ransomware

Un ransomware (rançongiciels) est un logiciel malveillant qui utilise le cryptage pour refuser l'accès aux ressources (telles que les fichiers de l'utilisateur), généralement dans le but de retenir une victime en otage et de la contraindre à payer une rançon. Une fois qu'un système a été infecté, les fichiers sont chiffrés de manière irréversible et la victime doit soit payer la rançon pour déverrouiller les ressources chiffrées, soit utiliser des sauvegardes pour les restaurer. Le nom « ransomware » est approprié car le malware demande une rançon à la victime.

Selon l'enquête CrowdStrike Global Security Attitude Survey (10), publiée en novembre 2020, plus de la moitié des 2200 personnes interrogées ont subi des attaques de ransomware au cours des 12 derniers mois.

3.6.1.5. Les attaques par mot de passe

Les attaques par mot de passe incluent toutes les cyberattaques où les pirates tentent de deviner, pirater et forcer les mots de passe à l'aide de divers programmes et outils de craquage de mots de passe tels que Aircrack, Cain, Abel, John the Ripper, Hashcat. Il existe différents types d'attaques par mot de passe telles que les attaques par force brute, les attaques par dictionnaire, l'attaque Rainbow Table, le Credential Stuffing, le Password Spraying et l'attaques de keylogger.

Selon le rapport Verizon 2021 Data Breach Investigations (11), *les informations d'identification compromises, telles que les mots de passe faibles, sont le principal point d'accès des pirates. Plus de six violations sur dix (61 %) proviennent des informations d'identification de l'utilisateur.*

3.6.1.6. Les attaques Malware

Les logiciels malveillants ou malveillants peuvent infecter un ordinateur et modifier son comportement, en détruisant des données ou en interceptant le trafic du réseau. Les logiciels malveillants peuvent se propager d'un appareil à l'autre, ou ils peuvent rester en place et n'affecter que l'appareil hôte.

Les pirates incitent à installer des logiciels malveillants sur un appareil. Une fois installé, un script malveillant s'exécute en arrière plan et contourne la sécurité, ce qui permet aux pirates d'accéder aux données sensibles et même de détourner le contrôle.

3.6.1.7. Les attaques DNS Spoofing

L'usurpation du système de noms de domaine (DNS) permet aux pirates d'envoyer du trafic vers de faux sites Web. Ces sites ressemblent presque à la destination (par exemple, la page de connexion d'une banque ou un compte de réseau social). Cependant, toutes les informations capturées (contrôlées) sont transmises directement aux pirates, qui ont accès aux données. Dans les attaques d'usurpation de DNS, les attaquants profitent du fait que les utilisateurs pensent que le site Web qu'ils visitent est légitime. Dans un exemple célèbre, la page d'accueil de Google a été usurpée en Roumanie et au Pakistan, redirigeant les utilisateurs

vers un site Web inconnu. Heureusement, dans ce cas, les pirates ne semblaient pas malveillants au-delà de la redirection du visiteur.

4. Méthodes de défense

Un système de protection informatique est un ensemble de techniques de protection contre les attaques informatiques et le piratage.

Les systèmes de protection informatique les plus connus sont :

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

4.1. 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).

On parle ainsi de signature de virus. Cette signature consiste en une série de bits ajoutés au fichier. La reconnaissance de cette séquence permet de détecter le virus. Si un virus est détecté par l'antivirus, celui-ci propose plusieurs méthodes pour le supprimer.

Chaque virus a sa propre signature, qui consiste en une série de bits attachés au fichier. Les programmes antivirus s'appuient sur des signatures de virus propres à chaque virus pour détecter les virus. Il s'agit de méthode de recherche de signature (scan).

Un antivirus utilise plusieurs méthodes pour l'éradication des virus, nous avons : (12)

- La suppression du code correspondant au virus dans le fichier infecté ;
- La suppression totale du fichier infecté ;
- La mise en quarantaine du fichier infecté, c'est-à-dire le déplacer dans un emplacement où il ne pourra pas être exécuté.

4.2. Pare-feu (Firewall)

Un pare-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. (12)

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.

4.2.1. Principe de fonctionnement

Jusqu'à récemment, les pare-feux étaient considérés comme l'une des pierres angulaires de la sécurité des réseaux informatiques. Les politiques d'accès peuvent être appliquées aux ressources du réseau (serveurs). Sa tâche principale est de contrôler le trafic entre différentes zones de confiance en filtrant les flux de données qui traversent différentes zones de confiance. Une zone de confiance comprend généralement Internet (zone non fiable) et au moins un réseau interne (zone hautement fiable).

L'objectif est de fournir une connectivité contrôlée et maîtrisée entre les zones avec différents niveaux de confiance grâce à l'application de politiques de sécurité et à un modèle de connexion basé sur le principe du moindre privilège. Ensuite, nous pouvons distinguer trois types de principes de fonctionnement de pare-feu : (12)

- Le filtrage de paquets (Packet Filtering) ;
- Le filtrage du flux (Circuit Filtering) ;
- La passerelle applicative (Application Gateway).

4.2.2. Catégories de Pare-feu

Les pare-feu existent en trois modèles. Chacun a ses avantages et ses inconvénients. Par conséquent, il faudra analyser les exigences réelles en termes de sécurité et les coûts associés avant toute utilisation :

- Les pare-feu Bridge ;
- Les pare-feu hardwares ;
- Les pare-feu logiciels ;
- Les pare-feu sans état (stateless firewall) ;
- Les pare-feu à états (stateful firewall) ;
- Les pare-feu applicatif ;
- Les pare-feu identifiant ;
- Les pare-feu personnel ;
- Les pare-feu Portail captif.

4.3. Système de détection d'intrusion

Un système de détection d'intrusion (IDS : Intusion Detection System) est un système qui surveille le trafic réseau ou examine les journaux d'audit de l'ordinateur hôte pour déterminer s'il y a eu une violation des politiques de sécurité spécifiques d'une organisation. IDS peut détecter les tentatives d'intrusion qui traversent un pare-feu, ou qui se produisent au sein d'un réseau local (LAN) derrière un pare-feu.

4.3.1. Les techniques d'analyse de trafic des IDS

Les techniques d'analyse de trafic se répartissent en deux classes principales : la détection d'anomalies, également appelée approche comportementale, et la détection d'intrusion, également appelée approche par scénarios. (12)

4.3.1.1. Approche comportementale

L'approche comportementale ("détection d'anomalies" en anglais) consiste à comparer un comportement observé à une référence comportementale normale. Une alerte est déclenchée en cas d'incompatibilité entre les deux comportements. Sur la base du comportement normal déterminé, IDS analyse le comportement de la machine. Une alarme IDS peut être déclenchée si l'ordinateur se connecte au milieu de la nuit sans qu'une personne soit présente. Dans ce type d'analyse, des profils sont créés et IDS réagit lorsque les machines connectées s'écartent des profils typiques. Diverses méthodes ont été proposées pour définir ce qui est normal (outils statistiques, systèmes experts, etc.).

- ***Avantages*** : Ce type d'analyse peut détecter des attaques inconnues. Aucune base de données requise.
- ***Inconvénients*** : Cette détection est très aléatoire, de sorte que les fausses alarmes peuvent se produire relativement facilement.

4.3.1.2. Approche par signature (par scénarios)

L'approche de signature est basée sur la comparaison du comportement observé avec les indications correspondantes qu'un scénario d'attaque est perçu comme intrusif et que le reste est considéré comme normal. L'IDS utilise donc ici une base de données de signatures d'attaques. Ces signatures peuvent être comparées à des séquences d'attaque. En effet, chaque attaque a ses propres caractéristiques (numéro de port, taille de paquet, protocole utilisé, etc.). Ces caractéristiques peuvent être collectées et placées dans une base de données qu'IDS interroge. Ce type d'IDS utilise des fichiers journaux. Déclenchez une alerte dès qu'il détecte une séquence suspecte (associée à des signatures dans la base de données).

Différents mécanismes ont été utilisés : l'analyse de signatures (pattern matching), les algorithmes génétiques, etc.

- ***Avantages*** : Ce type d'analyse peut détecter des attaques inconnues. Aucune base de données requise.
- ***Inconvénients*** : Cette détection est très aléatoire, de sorte que les fausses alarmes peuvent se produire relativement facilement.

4.3.2. Les types des IDS

Il existe trois grandes familles d'IDS :

- **NIDS** (Network Based Intrusion Detection System) pour surveiller l'état de la sécurité au niveau du réseau ;
- **HIDS** (Host-based Intrusion Detection System) pour surveiller votre état de sécurité au niveau de l'hôte. HIDS est particulièrement efficace pour déterminer si un hôte est compromis ;
- Des alertes plus pertinentes avec l'**IDS hybride** utilisant NIDS et HIDS.

Chapitre 2

L’origine de la cryptologie réside dans la Grèce antique. La cryptologie est un mot composé de deux éléments : «cryptos », qui signifie caché et « logos » qui signifie mot. La cryptologie est aussi vieille que l’écriture elle-même, et a été utilisé depuis des milliers d’années pour assurer les communications militaires et diplomatiques. Par exemple, le célèbre empereur romain Jule César utilisait un algorithme de chiffrement pour protéger les messages à ses troupes.

1. Définition de la cryptologie

La cryptologie, étymologiquement la science du secret, n'a que récemment été qualifiée de science. Cette science comprend la cryptographie, l'écriture secrète, et la cryptanalyse.

La cryptographie est un art ancien et une science nouvelle. C'est un art ancien car les Spartiates (Scytales) l'utilisaient déjà. Depuis les années 1970, la nouvelle science n'a fait l'objet que de recherches scientifiques académiques, c'est-à-dire universitaires, et est liée à de nombreuses autres études, telles que l'arithmétique modulo, l'algèbre, la théorie de la complexité, la théorie de l'information et les codes correcteurs d'erreurs. (12)

2. Définition de la cryptographie

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 de la manière dont les données peuvent être secrètement (cryptées) transmises sur un support particulier.

3. Définition de la cryptanalyse

La cryptanalyse, à l’inverse, est l’étude des procédés cryptographiques dans le but de trouver des faiblesses, en particulier, de pouvoir décrypter des messages chiffrés. La cryptanalyse est l’action consistant à trouver le message en clair sans connaitre la clé de déchiffrement.

4. Histoire de la cryptographie

L'histoire de la cryptographie ne remonte pas à nos jours, et pour en retrouver les premières traces il faut remonter à la civilisation babylonienne, environ 3000 ans avant notre ère. Quant à son application, elle s'est progressivement étendue du domaine militaire et politique au domaine civil, notamment sous l'influence d'Internet et de l’explosion des volumes de données qui révolutionnent notre quotidien. (13)

L’histoire du chiffrement retrace une épopée passionnante dans laquelle cryptographes et cryptanalystes se livrent une bataille acharnée, éternel recommencement de développement d’un algorithme par les uns, de décodage par les autres, de développement d’un nouvel algorithme plus puissant, etc.

L'histoire de la cryptographie a suivi une épopée fascinante dans laquelle les cryptographes et les décrypteurs (cryptanalystes) sont engagés dans une lutte acharnée, où les algorithmes développés par l'un sont piratés par d'autres et créés à nouveau. Des algorithmes plus puissants sont en cours de développement. (13)

Cette section propose un aperçu chronologique des technologies qui ont révolutionné la cryptographie, leur fonctionnement et leur histoire, avant d'énumérer l'enchaînement des actions mises en œuvre dans le monde de la cryptographie aujourd'hui.

5. 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. (12)

5.1. Algorithmes de substitution

La substitution cryptographique est une méthode de substitution qui maintient l'ordre des caractères et les remplace par les symboles d'un nouvel alphabet selon un algorithme précis. Cette méthode est basée sur l'arithmétique modulaire. La substitution est divisée en deux parties. Le premier est le remplacement monoalphabétique et le second est le remplacement polyalphabétique. (12)

5.1.1. Substitution Monoalphabétique

Consiste à remplacer chaque lettre du message par une autre lettre de l'alphabet.

5.1.1.1. Chiffre de César

C'est le chiffrement classique le plus simple et le plus ancien. Son principe est le décalage des lettres de l'alphabet.

Exemple

A partir de ce décalage de 3 lettres

Texte clair	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
Texte codé	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

Le texte à coder est : CE MESSAGE EST BIEN LISIBLE

Le texte codé sera alors : FH PHVVDJH HVW ELHQ OLVLEOH

5.1.2. Substitution Polyalphabétique

Consiste à utiliser une suite de chiffres monoalphabétique réutilisée périodiquement.

5.1.2.1. Algorithme de VIGENERE.

Il s'agit d'une amélioration significative par rapport au chiffre de César. Sa force réside dans l'encodage du message en utilisant 26 alphabets décalés au lieu d'un. Ce cryptage introduit le concept de clés. Les clés prennent généralement la forme de mots ou de phrases. Afin de pouvoir chiffrer le texte, nous utilisons la lettre clé de chaque lettre pour effectuer la substitution. Formellement, plus la clé ne sera longue et diversifiée, mieux le texte sera crypté.

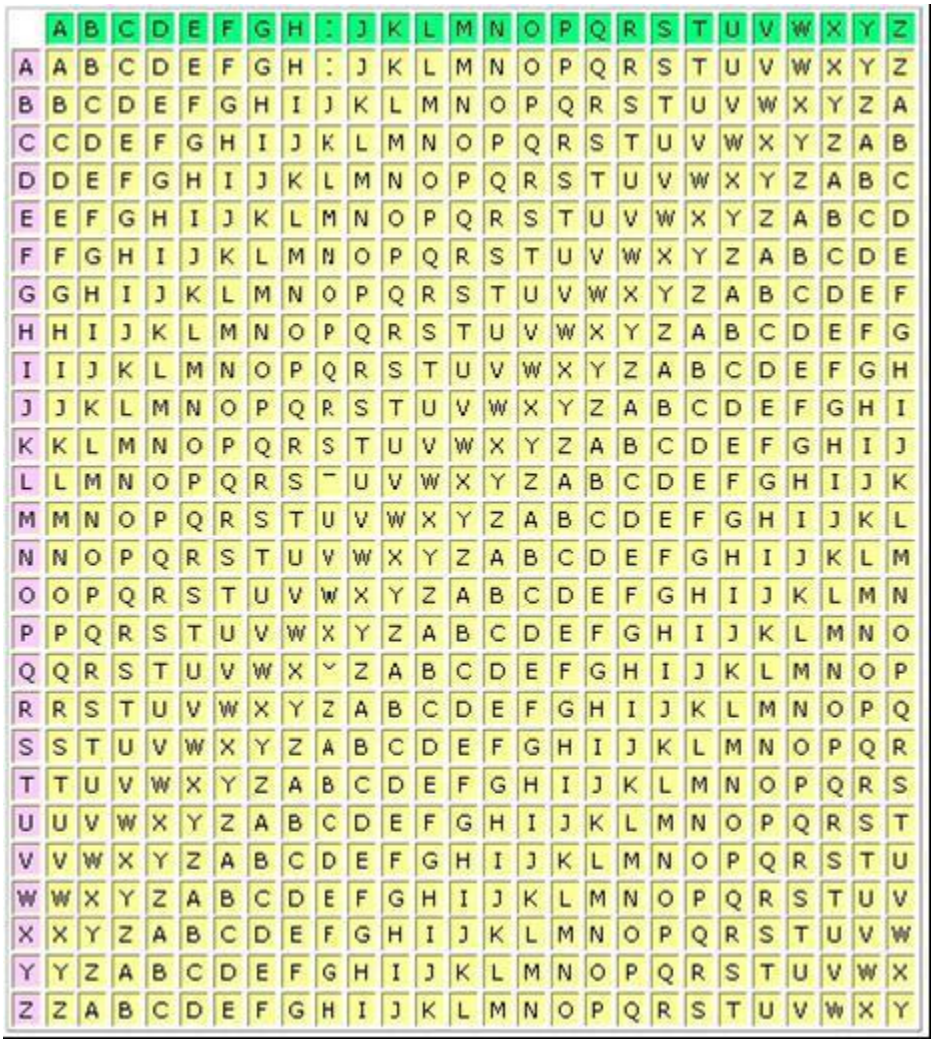
Principe

La lettre de la clé est dans la colonne la plus à gauche, la lettre du message clair est dans la ligne tout en haut. La lettre chiffrée est à l'intersection des deux.

Exemple 01

On va chiffrer le texte "CHIFFRE DE VIGENERE" avec la clé "BACHELIER" (cette clé est éventuellement répétée plusieurs fois pour être aussi longue que le texte clair).

Clair	C	H	I	F	F	R	E	D	E	V	I	G	E	N	E	R	E
Clé	B	A	C	H	E	L	I	E	R	B	A	C	H	E	L	I	E
Décalage	1	0	2	7	4	11	8	4	17	1	0	2	7	4	11	8	4
Chiffré	D	H	K	M	J	C	M	H	V	W	I	I	L	R	P	Z	I



Exemple 2

Clé : RECTEUR

Texte : RENDEZ VOUS REMAIN AUX FAB

Cela donne alors : RENDEZ VOUS REMAIN AUX FAB

RECTEU RREC TEURRE CTE URR

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.

Mathématiquement, on identifie les lettres de l'alphabet par des nombres de 0 à 25 (A=0, B=1...). Les opérations de chiffrement et de déchiffrement sont des opérations de chiffrement César pour chaque lettre. En désignant la i^{eme} lettre du texte clair par Texte[i], la i^{eme} du chiffré par Chiffré[i], et la i^{eme} 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]).

5.2. Algorithme de Transposition

Le chiffrement par transposition, aussi appelée « les *codes de permutation* » est basé sur les permutations des caractères du message en clair. De ce fait, les caractères sont toujours mais dans un autre ordre, En d'autres termes, les données cryptées sont réarrangées de telle manière qu'elles ne peuvent pas être comprises. Ce procédé est fondé essentiellement sur des matrices d'ordres n x p. (2)

5.2.1. La technique assyrienne

La cryptographie assyrienne est probablement la première preuve de l'utilisation de la cryptographie en Grèce en 600 avant Jésus Christ, afin de dissimuler des messages écrits sur des bandes de papyrus. (14)

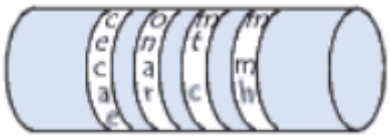


Figure 5. Bande papyrus

La technique consistait à :

- enrouler une bande de papyrus sur un cylindre appelé **scytale** ;

- écrire le texte longitudinalement sur la bandelette ainsi enroulée (le message dans l'exemple ci-dessus est « comment ça marche »).

Lorsque le message est développé, il n'est plus compréhensible ("cecaeonar mt cm mh"). Le destinataire n'a besoin que de cylindres de même diamètre pour pouvoir déchiffrer le message. En fait, un briseur (il existait des casseurs à l'époque !) pouvait déchiffrer le message en essayant successivement des cylindres de différents diamètres. Cela signifie que la méthode est statistiquement déchiffrable (il suffit de séparer les caractères un par un d'une distance donnée).

5.2.2. Transposition simple par colonnes

Dans la matrice, le message en clair est écrit horizontalement et le message chiffré est obtenu en lisant la matrice verticalement. De ce fait, l'ordre de la matrice représente la clé de chiffrement K.

Exemple

K = 4 x 4

M = INFORMATIQUES

Donc nous avons une matrice de 4 lignes et 4 colonnes, X est pour compléter la matrice.

1	2	3	4
I	N	F	O
R	M	A	T
I	Q	U	E
S	X	X	X

D'où le message chiffré est C = IRISNMQXFAUXOTEX

5.2.3. Transposition complexe par colonnes

Dans la transposition complexe par colonnes, le nombre de colonnes (p) de la matrice est fixé par le nombre de caractères d'une clé K (tous les caractères sont différents les uns aux autres), le nombre de ligne (n) dépendra de la longueur du message en clair. Le classement dans l'ordre alphabétique des caractères de la clé permet de fixer le séquencement de la lecture des colonnes de la matrice.

Exemple

K = GATS, p = 4

M = INFORMATIQUES

1	2	3	4
I	N	F	O
R	M	A	T
I	Q	U	E
S	X	X	X

Classement par ordre alphabétique des caractères de K

A	G	S	T
N	I	O	F
M	R	T	A
Q	I	E	U
X	S	X	X

Message chiffré est C = NMQX IRIS OTEX FAUX

6. Cryptographie Moderne

Les cryptosystèmes moderne utilisent des algorithmes de calcul complexes et ont de longues clés cryptographiques pour répondre aux objectifs de la cryptographie.

Dans les premiers jours de la sécurité, les professionnels de la sécurité ont estimé que la meilleure façon de garder l'algorithme de chiffrement sûr (sécurisé) était de cacher (masquer) les détails de l'algorithme de l'extérieur. Les cryptosystèmes modernes ne se reposent pas sur le secret de leurs algorithmes. En fait, les algorithmes de la plupart des systèmes cryptographiques sont largement disponibles au public dans une documentation correspondante (accompagnante) et sur Internet. Cela améliore effectivement la sécurité des algorithmes en les ouvrants au public. Au lieu de s'en remettre à des algorithmes secrets, les cryptosystèmes modernes se fondent sur le secret d'une ou plusieurs clés cryptographiques.

6.1. Cryptographie symétrique

La cryptographie symétrique utilise la même clé pour les processus de codage et de décodage ; cette clé est le plus souvent appelée "secrète" car toute la sécurité de l'ensemble est directement liée au fait que cette clé n'est connue que par l'expéditeur et le destinataire. La cryptographie symétrique se caractérise par une grande rapidité. (15)

Cependant, dans les années 1940, Claude Shannon a démontré que pour qu'un système à clé secrète soit complètement sécurisé, il devait utiliser une clé au moins aussi longue que le message chiffré. (16) (17)

6.1.1. Principe de base

Un expéditeur et un destinataire souhaitant communiquer de manière sécurisée à l'aide du cryptage conventionnel doivent s'entendre (convenir) d'une clé et ne pas la divulguer.

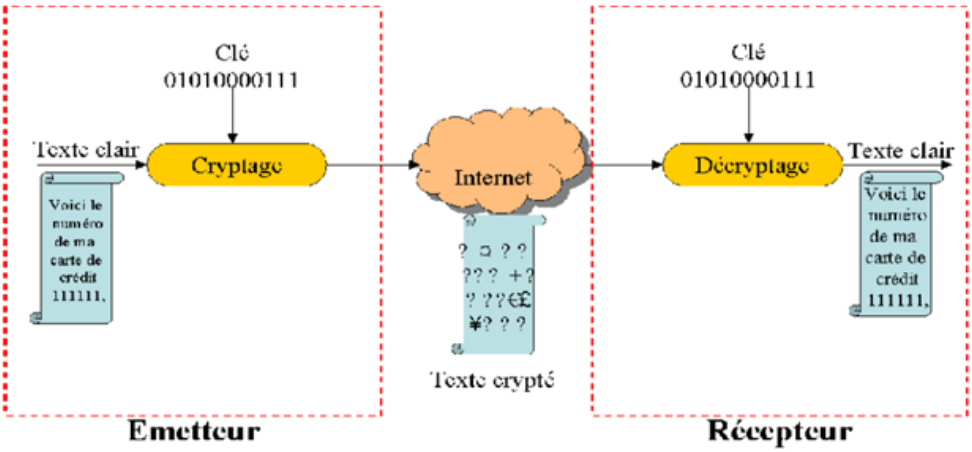


Figure 6. Cryptographie symétrique

6.1.2. Algorithme DES

Le Data Encryption Standard DES (standard de chiffrement de données) a été publié en 1977 et est devenue le premier algorithme de chiffrement à petite clé privée (56 bits) publié. Le DES consiste en un réseau Feistel à 16 tours. Le message à chiffrer est découpé en blocs de 64 bits et chaque bloc est découpé en deux sous-blocs de 32 bits.

Le cahier des charges était le suivant :

- L'algorithme repose sur une clé relativement petite, qui sert à la fois au chiffrement et au déchiffrement ;
- L'algorithme doit être facile à implémenter (côté logiciel et matériel) et doit être très rapide ;
- Le chiffrement doit avoir un haut niveau de sûreté, uniquement lié à la clé, et non à la confidentialité de l'algorithme. (17) (18)

6.1.2.1. 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 (les bits : 8, 16, 24, 32, 40, 48, 56, 64 sont des bits de détection d'erreur). Ce système 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). 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 sécurité) ;
- 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 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.

DES utilise huit tables de permutation (S-boxes), qui font l'objet de nombreuses controverses quant à leur contenu. Une vulnérabilité incorporée intentionnellement par les concepteurs a été suspectée. Ces rumeurs ont été dissipées au début des années 1990 avec la découverte de la cryptanalyse différentielle, qui a montré que les tables étaient bien conçues.

6.1.2.2. Les avantages

Le cryptage traditionnel a un avantage majeur. Sa rapidité le rend particulièrement adapté à l'envoi de grandes quantités de données. Le chiffrement symétrique est répandu et caractérise par une grande rapidité (chiffrement à la volée) propose des implémentations logicielles (Krypto Zone, firewalls logiciels type firewall-1, et VPN-1 de check point) et matériels (carte dédiées, processeurs cryptos 8 à 32 bits, algorithmes câblés…). (16) (17)

6.1.2.3. Les faiblesses

Ces systèmes exigent que l'émetteur et le destinataire connaissent la clé. Une faiblesse inhérente au système est la transmission de cette clé entre les parties prenantes, ils devront faire confiance à une tierce personne ou un moyen de communication sécurisé, toute personne interceptant la clé lors d'un transfert peut ensuite lire, modifier et falsifier toutes les informations cryptées ou authentifiées avec cette clé.

De la norme de cryptage de données DES au code secret de Jules César, la distribution des clés reste le problème majeur du cryptage conventionnel. (Autrement dit, comment faire parvenir la clé à son destinataire sans qu'aucune personne ne l'intercepte). Les moyens à déployer pour garantir la distribution sécurisée des clés entre les correspondants sont très onéreux, ce qui constitue un inconvénient supplémentaire. (16) (17)

6.1.3. Algorithme AES

Dans les années 1990, il est devenu évident que la norme de chiffrement DES la plus utilisée n'était plus en mesure de faire face aux évolutions techniques. Une nouvelle norme de chiffrement était nécessaire. Le successeur de l'algorithme de Rijndael, nommé d'après le nom de ses développeurs Vincent Rijmen et Joan Daemen, s'est imposé comme le successeur (une procédure qui a fait l'objet d'un appel d'offre datant de 1997) pour sa sécurité, sa flexibilité et ses performances et qui a été certifiée par le NIST en tant que norme de chiffrement avancée (AES) à la fin de l'année 2000.

AES divise également le texte brut à chiffrer en blocs. Ainsi, ce cryptosystème est, à l'instar de DES, basé sur le chiffrement de bloc. La norme prend en charge les clés 128, 192 et 256 bits. Cependant, au lieu de blocs 64 bits, AES utilise des blocs beaucoup plus grands de 128 bits qui sont encodés en plusieurs cycles consécutifs, à l'aide d'un réseau de permutation de substitution (SPN). Le successeur DES utilise également une nouvelle clé ronde pour chaque cycle de chiffrement, qui est dérivée récursivement de la clé initiale et liée au bloc de données à chiffrer en utilisant XOR.

Le processus de chiffrement peut être divisé en quatre étapes :

1. **Expansion des clés** : comme DES, AES utilise une nouvelle clé de ronde dans chaque boucle de chiffrement. Ceci est dérivé de la clé initiale par récusions. La clé initiale est étendue à une longueur qui vous permet de mapper le nombre requis de touches rondes de 128 bits. Chaque clé ronde est donc basée sur une sous-section de la clé initiale étendue. Le nombre de touches rondes nécessaires est le nombre de tours de chiffrement (R) y compris le tour final plus une touche ronde pour le tour préliminaire

2. (nombre de touches rondes = R + 1).

3. **Phase préliminaire :** lors de la ronde préliminaire, le bloc d'entrée 128 bits est transféré dans une table bidimensionnelle (tableau) et relié à la première clé ronde à l'aide de XOR (Key Addition). Le tableau contient 4 lignes et 4 colonnes. Chaque cellule contient donc un octet (8 bits) du bloc à chiffrer.

4. **Rondes de chiffrement** : le nombre de rondes de chiffrement dépend de la longueur de clé utilisée, 10 rondes pour AES128, 12 rondes pour AES192 et 14 rondes pour AES256 :

    a. **Sous-octets** : les sous-octets sont une substitution monoalphabétique. Chaque octet du bloc à chiffrer est remplacé par un équivalent en utilisant une S-Box.
    b. **Les rangées de quarts** : dans le contexte de la transformation ShiftRow, les octets dans les cellules du réseau (voir le tour préliminaire) sont déplacés cycliquement vers la gauche.
    c. **MixColumns** : avec MixColumns, l'algorithme AES inclut une transformation dans laquelle les données sont fusionnées dans les colonnes du tableau. Cette étape est basée sur un nouveau calcul de chaque cellule individuelle. Pour cela, les colonnes de la matrice sont soumises à une multiplication matricielle et les résultats sont liés par XOR.
    d. **KeyAddition** : à la fin de chaque cycle de chiffrement, un autre KeyAddition a lieu. Comme dans le tour préliminaire, il est basé sur un lien XOR entre le bloc de données et la touche ronde courante.

5. **Final Round :** le dernier round est le dernier round de chiffrement. Contrairement aux cycles précédents, il ne contient pas de transformations MixColumns et ne comprend donc que les opérations SubBytes, ShiftRows et KeyAddition. Le résultat du dernier tour est le texte secret.

Le décryptage des données cryptées AES est basé sur un investissement dans des algorithmes de cryptage. Cela implique une séquence d'étapes plus les opérations ShiftRow, MixColumns et SubBytes.

AES est certifié **hautement sécurisé** grâce à son algorithme. À ce jour, aucune attaque pratique n'est connue. Les attaques par force brute sont inefficaces en raison de la longueur de clé d'au moins 128 bits. De plus, des opérations telles que ShiftRows et MixColumns garantissent un mixage optimal des bits : dans le résultat, chaque bit dépend de la clé. De plus, le cryptosystème impressionne par sa simplicité d'implémentation et sa grande vitesse. AES est utilisé comme norme de chiffrement pour WPA2, SSH et IPSec ainsi que comme algorithme de chiffrement pour les archives de fichiers compressés telles que 7-Zip ou RAR.

Cependant, les données chiffrées AES ne sont protégées contre l'accès par des tiers que si la clé reste secrète. Étant donné la même clé est utilisée pour le chiffrement et le déchiffrement, le cryptosystème est affecté par le problème de distribution des clés comme toute autre méthode symétrique. L'utilisation sécurisée d'AES est donc limitée aux domaines d'application qui ne nécessitent pas d'échange de clés ou qui le permettent via un canal sécurisé.

Cependant, la communication cryptée sur Internet nécessite que les données soient cryptées sur un ordinateur et décryptées sur un autre. Des cryptosystèmes asymétriques sont ici établis, permettant l'échange sécurisé de clés symétriques ou de fonctions sans échange de clés partagées.

6.2. Cryptographie Asymétrique

La cryptographie asymétrique à clé publique est apparue pour la première fois en 1976 avec la publication d'un ouvrage sur la cryptographie par Whitfield Diffie et Martin Hellman, c'est méthode de chiffrement qui s'oppose à la cryptographie symétrique.

Dans un tel cryptosystème, les clés existent en paires d'où l'appellation bi-clés :

- Une clé publique pour le chiffrement ;
- Une clé secrète pour le déchiffrement.

L'utilisateur d'un cryptosystème asymétrique, choisit une clé aléatoire (la clé privé), à partir de cette clé et en appliquant la fonction à sens unique il calcule la clé publique qu'il diffuse au travers d'un canal non sécurisé.

Lorsqu'une personne désire lui envoyer un message il lui suffit de chiffrer ce dernier à l'aide de la clé publique. Le destinataire sera en mesure de déchiffrer le message à l'aide de sa clé privé.

Ce système est basé sur une fonction facile à calculer dans un sens (appelé fonction à trappe à sens unique) et mathématiquement très difficile à inverser sans la clé privée appelé trappe. (17) (18)

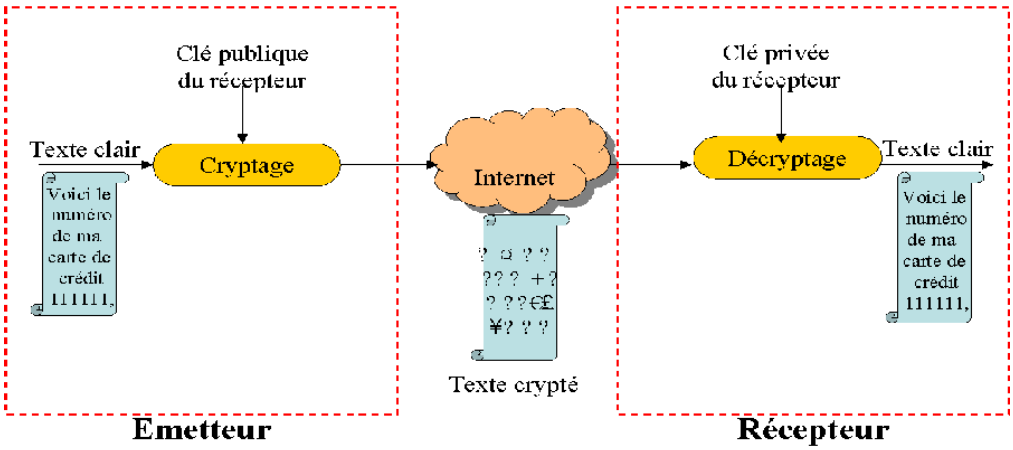


Figure 7. Cryptographie asymétrique

6.2.1. Algorithme RSA

Le premier système à clé publique solide à avoir été inventé et le plus largement utilisé aujourd'hui est le système RSA. Publié en 1977 par Ron Rivest, Adi Shamir et Leonard Adleman au Massachusetts Institute of Technology (MIT), est basé sur la difficulté de factoriser de grands nombres, et la fonction à sens unique utilisée est la fonction "puissance".

RSA a été breveté par le Massachusetts Institute of Technology (MIT) en 1983 aux États-Unis. Le brevet a expiré le 21 septembre 2000. (12)

6.2.1.1. Fonctionnement de 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 hypothèse importante est que le déchiffrement est "calculatoirement impossible" avec la clé publique. En particulier, de reconstruire la clé privée à partir de la clé publique. Si la confidentialité doit être maintenue), ne le permettez pas. Le chiffrement RSA est souvent utilisé pour transmettre des clés de chiffrement symétriques. Cela permet à l'échange de se dérouler de façon confidentielle. (12)

Ils utilisent la congruence entière et le petit théorème de Fermat pour obtenir une fonction de compromis secret unidirectionnel 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).

6.2.1.2. Les étapes de l'algorithme

Départ :

· Il est facile de fabriquer de grands nombres premiers p et q (+- 100 chiffres)

· Étant donné un nombre entier n = p*q, il est très difficile de retrouver les facteurs p et q

1) Création des clés

· La clé secrète : 2 grands nombres premiers p et q

· La clé publique : n = p*q ; un entier e premier avec (p-1) (q-1)

2) Chiffrement : le chiffrement d'un message M en un message codé C se fait suivant la transformation suivante :

$C = M^e \bmod n$

3) Déchiffrement : il s'agit de calculer la fonction réciproque

$M=C^d \bmod n$ tel que $e.d = 1 \bmod [(p-1)(q-1)]$

Exemple

Chiffrer BONJOUR

1) Alice crée ses clés :

· La clé secrète : p = 53, q = 97 (Note : en réalité, p et q devraient comporter plus de 100 chiffres !)

· La clé publique : e = 7 (premier avec 52*96), n = 53*97 = 5141

2) Alice diffuse sa clé publique (par exemple, dans un annuaire).

3) Bob ayant trouvé le couple (n, e), il sait qu'il doit l'utiliser pour chiffrer son message. Il va tout d'abord remplacer chaque lettre du mot BONJOUR par le nombre correspondant à sa position dans l'alphabet :

B = 2, O = 15, N = 14, J = 10, U = 21, R = 18

BONJOUR = 2 15 14 10 15 21 18

4) Ensuite, Bob découpe son message chiffré en blocs de même longueur représentant chacun un nombre plus petit que n. Cette opération est essentielle, car si on ne faisait pas des blocs assez longs (par exemple, si on laissait des blocs de 2 chiffres), on retomberait sur un simple chiffre de substitution que l'on pourrait attaquer par l'analyse des fréquences.

BONJOUR = 002 151 410 152 118

5) Bob chiffre chacun des blocs que l'on note B par la transformation $C = B^e \bmod n$ (où C est le bloc chiffré) :

$C1 = 2^7 \bmod 5141 = 128$

$C2 = 151^7 \bmod 5141 = 800$

$C3 = 410^7 \bmod 5141 = 3761$

$C4 = 152^7 \bmod 5141 = 660$

$C5 = 118^7 \bmod 5141 = 204$

On obtient donc le message chiffré C : 128 800 3761 660 204

6.3. Fonctions de Hachage

6.3.1. Principe

Les fonctions de hachage sont des fonctions qui compriment une entrée de longueur arbitraire pour produire un résultat de longueur fixe. Si les fonctions de hachage répondent à d'autres exigences supplémentaires, elles deviennent un outil très puissant dans la conception des techniques de protection de l'authenticité et de l'intégrité des informations. (19)]

6.3.2. Définition

Une fonction de hachage H (1) est un algorithme déterministe et efficace, qui prend comme entrée une donnée (un message) binaire M de taille quelconque, et produit à la sortie un haché h de taille fixe (en général entre 128 et 512 bits). Cet haché, appelé aussi "condensé" ou "empreinte" doit dépendre de tous les bits du message, et il est utilisé comme représentant comprimé de celui-ci. (20)

$$H : \{0, 1\}^* \rightarrow \{0, 1\}^n, h = H(M). \qquad (1)$$

Les fonctions de hachage ont été inventées dans les années 1950 (21)]. Celles-ci sont très utiles en informatique car les empreintes digitales peuvent être utilisées pour identifier les messages sous une forme plus compacte.

6.3.3. Applications des fonctions de hachage

Les fonctions de hachage sont utilisées par de nombreux systèmes cryptographiques, à la fois en cryptographie symétrique et en cryptographie asymétrique. Nous présentons ci-dessous une liste non exhaustive de ces systèmes.

6.3.3.1. Code d'authentification de message

Les codes d'authentification de message (ou MAC) sont des empreintes d'une donnée qui dépendent d'une clé secrète k, et qui ne peuvent être calculées qu'en connaissant k (c'est l'équivalent symétrique d'une signature). En ce sens, ils peuvent être vus comme des fonctions de hachage à clé secrète.

Un MAC doit être difficile à contrefaire, c'est-à-dire qu'un attaquant ne doit pas pouvoir calculer de MAC sans connaître la clé. Une façon simple pour construire un MAC est de faire rentrer le message et la clé dans une fonction de hachage : MACk(M) = F(k|M) ; cette construction s'appelle Secret-Prefix MAC, elle est sûre si la fonction de hachage se comporte comme une fonction aléatoire, puisqu'on ne peut pas prévoir la valeur de la fonction en un point en connaissant seulement sa valeur en d'autres points. Cependant, des fonctions très utilisées comme MD5 ou SHA-1 ne peuvent pas être utilisées de cette façon pour construire un MAC. Pour construire un MAC à partir d'une fonction de hachage, on utilise donc des constructions plus complexes, comme HMAC (22). HMAC repose sur l'utilisation de deux constantes p1 et p2, et sur une fonction de hachage H. Le MAC d'un message M avec une clé K est défini par (23) :

$$MAC_k(M) = H(k \oplus p_1 | H(k \oplus p_2 | M))$$

6.3.3.2. Signature électronique (Hash-and-Sign)

Les schémas de signature sont peut-être l'application la plus importante des fonctions de hachage. Cela permet aux utilisateurs de signer des messages avec leur clé privée (24)]. N'importe qui peut vérifier la validité de cette signature à l'aide de la clé publique correspondante.

Les schémas de signature tels que RSA et ElGamal permettent d'authentifier les messages, mais ils appartiennent à la cryptographie asymétrique et nécessitent donc des calculs complexes et coûteux. Par conséquent, dans certains cas, le calcul prend trop de temps lorsqu'il est appliqué à des messages très longs. En pratique, au lieu d'appliquer le schéma de signature directement au message long, on applique la signature à un hachage du message. Par conséquent, l'opération de signature est effectuée avec de petits identifiants et est peu coûteuse.

Si on veut que le signataire ne puisse pas répudier ses signatures (la même signature pour plusieurs massages), il faut que la fonction de hachage soit résistante aux collisions.

Il est théoriquement possible de construire des mécanismes de signature permettant de signer des données de taille quelconque, cependant de telles constructions seraient très lentes. Pour pouvoir signer des données de taille quelconque, la solution la plus répandue consiste donc à leur appliquer une fonction de hachage, puis d'appliquer la fonction de signature S sur l'empreinte obtenue (voir la Figure 8).

La signature d'un message M est alors S(H(M)).

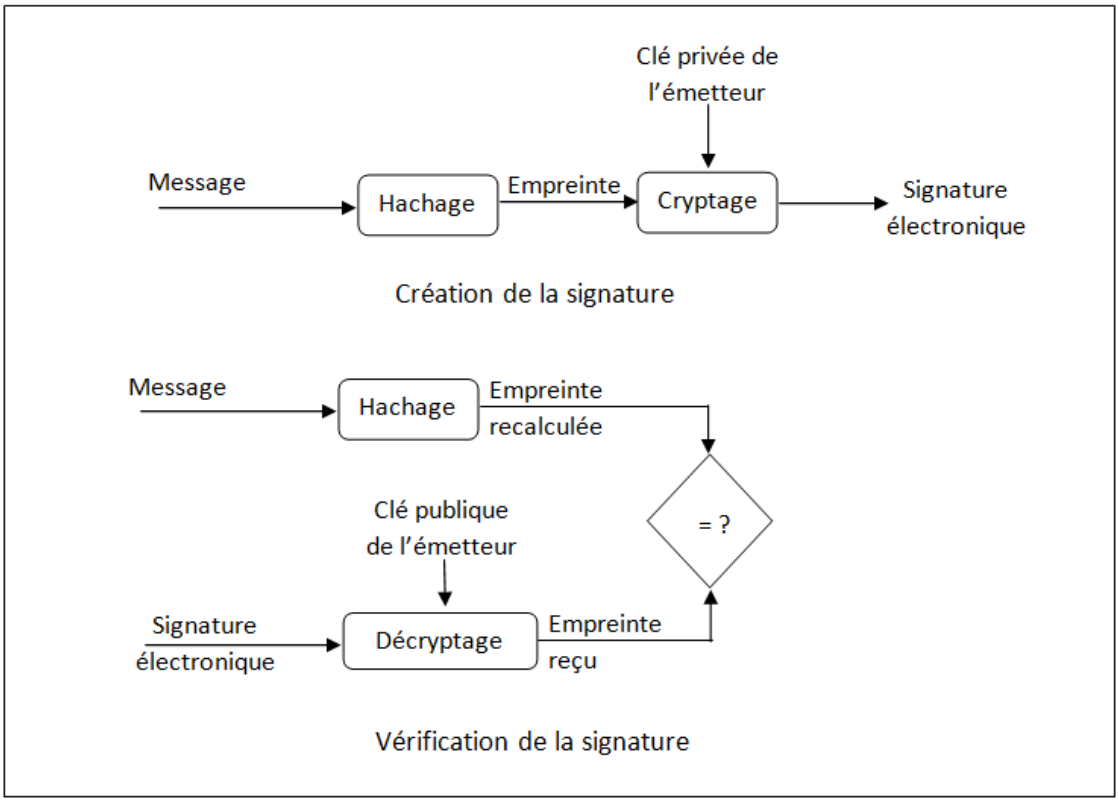


Figure. 8 Schéma de signature électronique

6.3.3.3. Génération de nombres pseudo-aléatoires

Les propriétés statistiques des fonctions de hachage et leur caractère à sens unique peuvent être exploités dans des contextes de génération de nombres aléatoires (25)]. En effet, de nombreux mécanismes cryptographiques nécessitent la génération de nombres aléatoires : génération de clés, signature électronique, chiffrement asymétrique, génération de valeurs d'initialisation pour le chiffrement symétrique.

Les nombres ainsi générés doivent être tirés uniformément et être imprédictibles pour un attaquant. Les nombres aléatoires sont générés à partir de secrets stockés en mémoire et/ou de phénomènes physiques aléatoires (seed). La traduction en chaînes de bits de phénomènes physiques aléatoires peut conduire à des chaînes non uniformément distribuées. Dans ce cas l'utilisation de fonctions de hachage permet d'extraire l'entropie de ces chaînes, c'est-à-dire d'obtenir des suites de bits indépendants et uniformément distribués.

D'autre part, le caractère non inversible des fonctions de hachage permet de dériver des nombres pseudo-aléatoires à partir de secrets sans les compromettre. Ces fonctions permettent également de mettre à jour les valeurs secrètes, afin que leur compromission éventuelle

n’affecte pas leurs valeurs passées. Le schéma défini par Barak et Halevi dans (26), fait un tel usage des fonctions de hachage.

On peut facilement construire un générateur pseudo-aléatoire à partir d’une fonction de hachage. Par exemple, si x est le "seed" du générateur, on peut utiliser F(x|0), F(x|1), F(x|2). Cette suite pseudo-aléatoire peut être utilisée comme suite générée dans un schéma de chiffrement par flux.

Certains protocoles de communication reposent entièrement sur la cryptographie symétrique. Les équipements qui communiquent possèdent dans ce cas un secret partagé S, défini avant leur déploiement. Afin de limiter l’impact de la compromission des clés utilisées pour protéger des communications, il est nécessaire de renouveler régulièrement leurs valeurs. Une manière de procéder consiste à calculer la $i^{ème}$ clé k_i comme le haché du secret S et d’un compteur i.

6.3.3.4. Stockage de mots de passe

De nombreux systèmes informatiques authentifient leurs utilisateurs grâce à la connaissance de mots de passe. Un inconvénient potentiel de cette méthode est que si les mots de passe des utilisateurs sont gardés en mémoire sur le système et si un attaquant parvient à accéder à ces mots de passe, il peut s’authentifier sous l’identité de n’importe quel utilisateur. Ce problème peut être résolu à l’aide de fonctions de hachage.

Les fonctions de hachage sont utilisées pour éviter de stocker des mots de passe en clair. Ainsi, quand on se connecte sur un ordinateur, la machine calcule un haché du mot de passe, et le compare au haché préalablement connu. Ceci permet d’éviter de stocker le mot de passe en clair, et si la machine est compromise, l’attaquant ne pourra pas retrouver les mots de passe des utilisateurs (voir la Figure 9). Cette méthode semble sûre puisqu’une fonction de hachage est à sens unique. Ainsi, même si un attaquant mettait la main sur les hachés des mots de passes stockées sur la machine, il lui serait pratiquement impossible de retrouver le mot de passe d’accès. (27)

Cependant, cette méthode présente de graves failles de sécurité qui peuvent être corrigées. Supposons que deux utilisateurs aient le même mot de passe. Ensuite, les hachés de mot de passe sont les mêmes. Si un attaquant voit cela, il peut deviner que le mot de passe est un mot du dictionnaire plutôt qu'une séquence aléatoire de symboles. Ainsi, il peut trouver un moyen d'accéder au compte que vous souhaitez en essayant chaque mot du dictionnaire (attaque du dictionnaire). Pour contrer ces types d'attaques, on ajoute généralement une séquence aléatoire appelée sel. Il s'agit, par exemple, de la date et de l'heure auxquelles le mot de passe a été attribué. Cette séquence aléatoire est stockée en clair dans la mémoire du système. Un hash est donc un hash du mot de passe concaténé avec une composante aléatoire pour créer deux hash complètement différents (par effet d'avalanche).

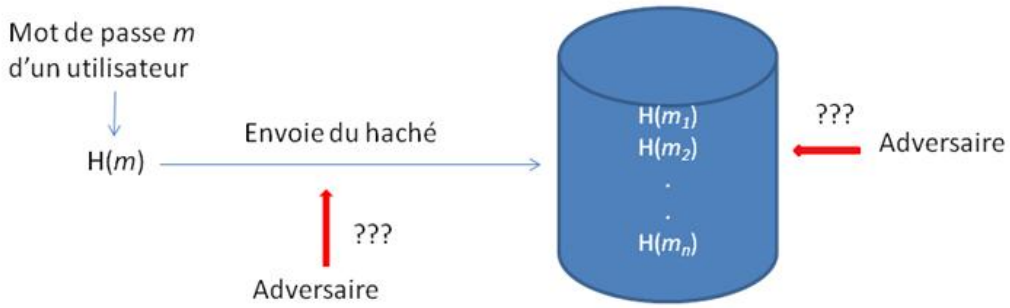


Figure 9. Stockage sécurisé de mots de passe.

6.3.3.5. Schémas d'engagement

Les fonctions de hachage sont aussi utilisées dans certains protocoles pour s'engager à l'avance sur le choix d'une certaine valeur ou même pour confirmer la connaissance d'un certain secret, sans le révéler. Par exemple, le calcul de secret partagé entre deux entités utilise souvent ce genre de techniques. (28)

Une fonction de hachage permet de s'engager sur un message : on dévoile d'abord le haché d'un message, puis on révèle plus tard son contenu. Le haché ne révèle pas d'information exploitable sur le message, mais il garantit qu'on ne peut pas modifier le message après avoir révélé le haché. On peut utiliser un schéma d'engagement pour réaliser des enchères secrètes, par exemple. Ce type d'engagement est aussi utile dans de nombreux protocoles cryptographiques, par exemple, pour que plusieurs participants choisissent une valeur aléatoire sans possibilité de triche : chaque participant s'engage sur une valeur aléatoire, puis les valeurs sont révélées et on calcule leurs hachés pour vérifier.

Un schéma d'engagement permet aussi d'horodater un document, pour garantir qu'il existait à une certaine date. Pour construire un horodatage, ou time-stamp, on enregistre le haché d'un message auprès d'une autorité, et cette autorité peut ensuite certifier la date à laquelle le haché, et donc le message, était connu. Un time-stamp peut servir à prouver l'antériorité d'une découverte. (23)

Pour ce type d'utilisation, on n'a pas forcément besoin de résistance en collision (si on sait fabriquer une collision, il est vrai que les deux messages sont connus au moment où le haché est publié), mais on a besoin d'une notion plus forte que la résistance en pré-image. En effet, il est possible de s'engager sur une valeur particulière, pour laquelle il sera plus facile de modifier le message ultérieurement.

6.3.3.6. Protection des fichiers

Une des façons d'utiliser une fonction de hachage est de considérer l'empreinte d'un document comme un identifiant unique. En effet, une bonne fonction de hachage est résistante aux collisions, i.e. on ne peut pas trouver deux messages ayant la même empreinte (23). On peut utiliser ces identifiants pour vérifier l'intégrité d'un document, ou pour identifier un document si l'empreinte est mieux protégée que le document lui-même. Ceci sert notamment dans certains protocoles de téléchargement pair-à-pair : on obtient le haché du document depuis un serveur central, et le haché sert à vérifier les données reçues depuis les pairs.

6.3.3.7. Authentification par défi/réponse

Les MAC sont souvent utilisés pour construire des protocoles d'authentification. Dans un protocole d'authentification simple, un client veut s'identifier auprès d'un serveur avec qui il partage un mot de passe. Le serveur envoie un message aléatoire appelé défi ou challenge au client, et le client répond avec un MAC du défi, en utilisant le mot de passe comme clé. Cela ne révèle pas d'information utile sur la clé à un adversaire, mais le serveur peut vérifier que le calcul est correct et donc identifier l'utilisateur (20)].

Les protocoles par défi/réponse sont très utilisés en pratique. Par exemple, le mode d'authentification CRAM-MD5 (29), utilisé dans SASL, POP3, IMAP, et SMTP, est un protocole défi/réponse construit avec HMAC-MD5.

6.3.4. Les algorithmes de fonction de hachage

La famille de fonctions de hachage MD-SHA (Messages Digest - Standard Hash Algorithm) est la norme pour les fonctions de hachage cryptographique depuis de nombreuses années, mais la plupart de ces fonctions ont été cryptanalysées.

Le principal avantage des fonctions de hachage MD-SHA est leur rapidité dans une implémentation logicielle. En effet, seules des opérations très simples et bien supportées par les microprocesseurs 32 bits sont utilisées pour le hachage (additions modulaires, fonctions booléennes, rotations et décalages) (20). Dans ce qui suit nous allons présenter les principales fonctions de cette famille.

6.3.4.1. Algorithme MD5

La fonction MD5 a été conçue par Rivest (30). Cette fonction est largement utilisée et est toujours disponible aujourd'hui dans plusieurs systèmes et protocoles. Cela représente une amélioration par rapport à MD4. La fonction de compression (représentant 16 étapes) a été étendue d'un tour et les étapes de base ont également changé.

La fonction de compression de MD5 a donc un autre tour par rapport à MD4. Il s'agit d'une nouvelle fonction booléenne et de constantes définies à chaque étape. Le principe général reste le même : des hachés de taille n = 128 bits pour un état interne de r = 4 registres de w = 32 bits chacun, initialisé avec la variable de chaînage d'entrée :

$$A_{-3} = h_0, \quad A_{-2} = h_3, \quad A_{-1} = h_2, \quad A_0 = h_1$$

À chaque exécution de la fonction de compression, 16 mots de message sont traités, durant 4 tours de 16 étapes chacun (c'est-à-dire 64 étapes en tout). L'expansion de message reste très simple : pour chaque tour j, une permutation j de l'ordre des mots de message est définie (0 représente l'application identité). Ainsi, nous avons pour la $k^{ème}$ étape du tour j, avec

$$0 \leq j \leq 3 \text{ et } 0 \leq k \leq 15 : \; W_{j\times 16+k} = M_{\pi_j(k)}$$

Comme pour MD4, on peut observer que puisque l'expansion de message est une permutation par tour des mots du message d'entrée M, chaque mot de M sera utilisé une fois pour chaque tour. L'expansion de message de MD5 est donc similaire, sauf que les permutations ont été légèrement modifiées.

Pour chaque étape i le registre cible A_{i+1} est mis à jour par la fonction f_j, qui dépend du tour j auquel i appartient :

$$A_{i+1} = f_j(A_i, A_{i-1}, A_{i-2}, A_{i-3}, W_i, K_i, s_i)$$
$$= A_i + (A_{i-3} + F_j(A_i, A_{i-1}, A_{i-2}) + W_i + K_i)^{<<<s_i}$$

Où K_i sont des constantes prédéfinies pour chaque étape, les s_i sont des valeurs de rotation prédéfinies pour chaque étape, et les fonctions F_j sont des fonctions booléennes définies pour chaque tour et prenant 3 mots de 32 bits en entrée. À la fin des 64 étapes, les mots de la sortie de la fonction de compression sont calculés par :

$$h'0 = A_{61} + A_{-3}, \quad h'_1 = A_{64} + A_0, \quad h'_2 = A_{63} + A_{-1}, \quad h'_3 = A_{62} + A_{-2}$$

6.3.4.2. Algorithme SHA-1

SHA-1 a été publiée en 1995 par le NIST pour remplacer la fonction SHA-0 (31)]. Malgré ses vulnérabilités connues qui conduisent à son remplacement progressif par SHA-2, elle reste probablement aujourd'hui la fonction la plus utilisée dans la pratique. Nous décrivons maintenant les détails de son fonctionnement.

SHA-1 est construite avec l'algorithme d'extension de domaine de Merkle-Damgård. Sa fonction de compression permet le traitement de blocs de message de 512 bits et de variables de chaînage de 160 bits. Elle est construite à partir d'une permutation paramétrée en mode de Davies-Meyer, où l'opération XOR est remplacée par 5 additions modulo 2^{32} entre les 5 registres de 32 bits d'entrée de la variable de chaînage et les 5 registres de sortie de la permutation. Le bloc de message, divisé en 16 mots de 32 bits (M_0, ...,M_{15}) passe par une expansion linéaire permettant d'obtenir 80 mots de 32 bits, de la manière suivante (25) :

$$\forall i \in 0, ..., 15, W_i = M_i$$

$$\forall i \in 16, ..., 79, W_i = (W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16}) << 1$$

Cette rotation de 1 bit vers la gauche dans l'expansion de message constitue une des différences entre SHA-0 et SHA-1. A partir de la variable de chaînage d'entrée, on initialise 5 registres A, B, C, D et E.

$$A_0 = H_0^{i-1}, \quad B_0 = H_1^{i-1}, \quad C_0 = H_2^{i-1}, \quad D_0 = H_3^{i-1}, \quad E_0 = H_4^{i-1}$$

La permutation paramétrée de SHA-1 se décompose en 4 tours de 20 étapes. Chaque étape est définie par les opérations suivantes :

$$A_{i+1} = (A_i << 5) \oplus \phi_i(B_i, C_i, D_i) \oplus E_i \oplus W_i \oplus k_i$$

$$B_{i+1} = A_i$$

$$C_{i+1} = B_i^{<<<30}$$

$$D_{i+1} = C_i$$

$$E_{i+1} = D_i$$

Ou k_i est une constante dépendante du numéro de l'étape et $\emptyset_i$est la juxtaposition de 32 fonctions identiques de 3 bits vers 1 bit. Une étape de la fonction de compression de SHA-1 est décrite dans la figure 10. La nouvelle valeur de la variable de chainage est définie par :

$$H_i^0 = A_0 \oplus A_{80}$$

$$H_i^1 = B_0 \oplus B_{80}$$

$$H_i^2 = C_0 \oplus C_{80}$$

$$H_i^3 = D_0 \oplus D_{80}$$

$$H_i^4 = E_0 \oplus E_{80}$$

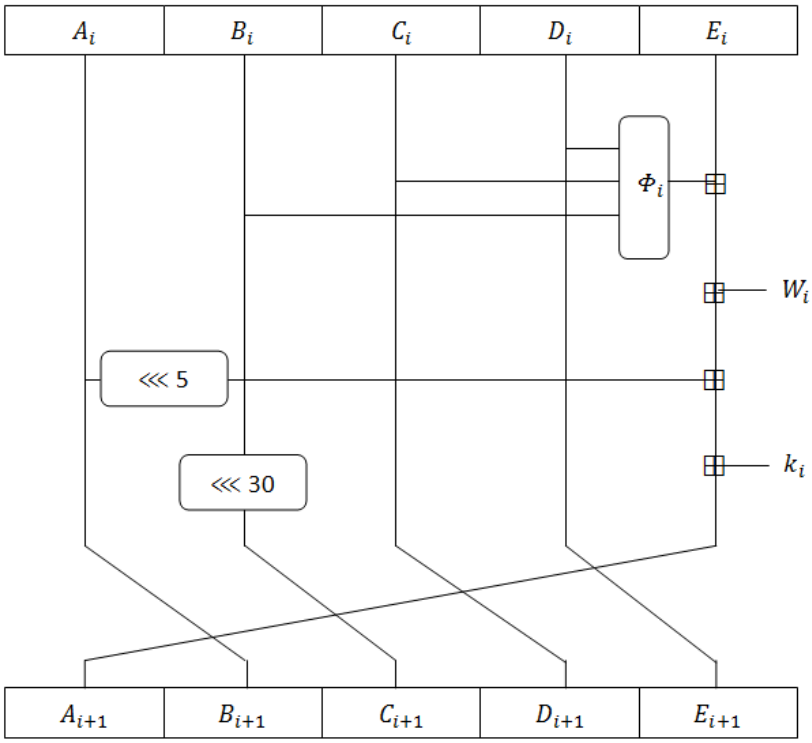


Figure 10. Une étape de la fonction de compression de SHA-1.

6.4. La signature électronique

La signature numérique est un mécanisme qui permet d'authentifier les messages. En d'autres termes, comme une signature sur un document papier, cela prouve que le message a bien été envoyé par l'expéditeur spécifique.

6.4.1. Principe de la signature électronique

La signature électronique repose sur deux familles d'algorithmes, qui seront utilisés de manière complémentaire :

- des algorithmes de chiffrement dit « asymétriques » ou à « clé publique » qui sont RSA et DSA ;
- des fonctions de hachages qui sont MD5 et SHA.

6.4.2. Les objectifs de la signature numérique

Les signatures numériques sont souvent utilisées dans trois objectifs pour prendre en charge les propriétés d'intégrité des données, d'authentification et de non-répudiation. (32)

- L'intégrité des données. Bob peut vérifier que le message d'Alice n'a pas été modifié entre l'envoi et la réception. Toute modification du message produirait une signature complètement différente ;
- L'Authenticité. Tant que la clé privée d'Alice est gardée secrète, Bob peut se servir de sa clé publique pour confirmer que les signatures numériques ont été créées par Alice et personne d'autre ;
- Non-répudiation. Une fois la signature générée, Alice ne pourra pas nier l'avoir appliqué à l'avenir, à moins que sa clé privée ne soit compromise d'un quelconque manière.

6.4.3. Étapes de signature et de chiffrement d'un message

La Figure 11 ci-dessous montre la série d'opérations qu'Alice doit exécuter pour envoyer un message signé et chiffré à Bob. (33)

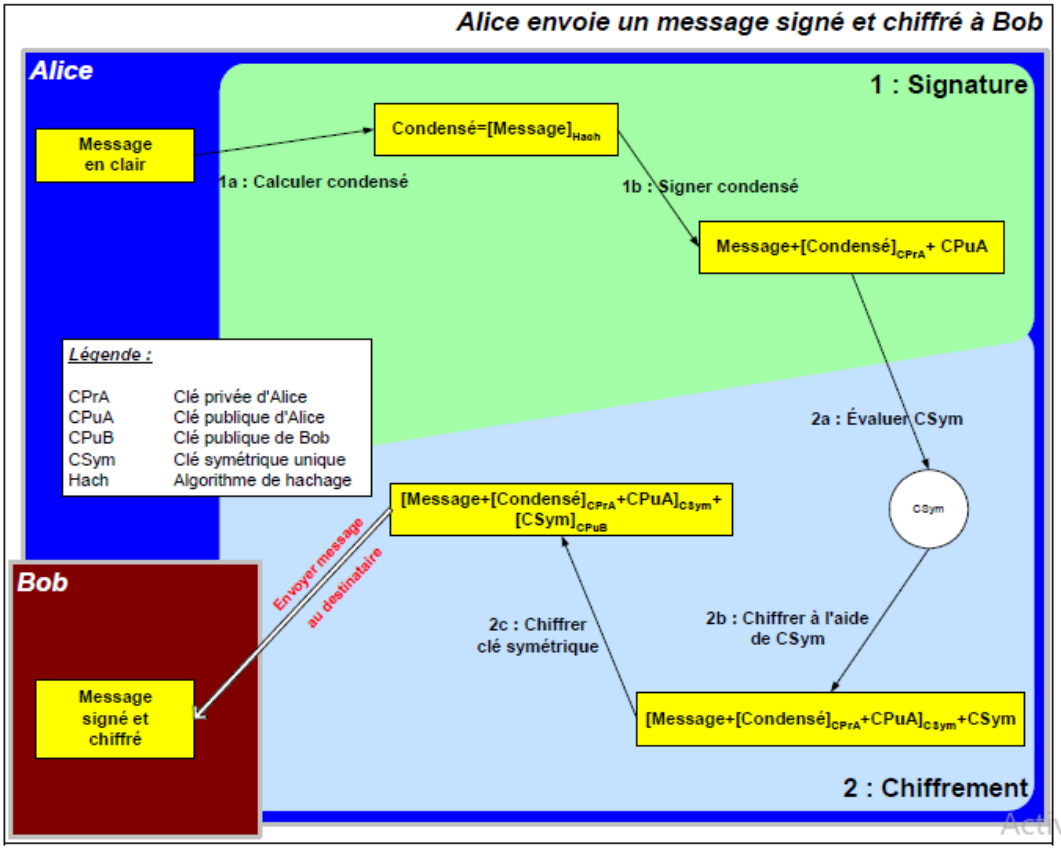


Figure 11. Processus de signature et de chiffrement à l'aide des clés

1) Signature du message. La signature numérique comprend deux étapes :

a) Évaluation du condensé de message. L'objectif principal de l'évaluation d'un condensé est de s'assurer que le message ne sera pas altéré. C'est ce qu'on entend par intégrité du message ;

b) Signature du condensé. Une signature est en fait un chiffrement à l'aide de la clé privée de l'émetteur (Alice dans le cas présent). On retrouve également dans cette signature le nom de l'algorithme de hachage utilisé par l'émetteur. La clé publique de l'émetteur est aussi annexée à la signature. Grâce à ces informations, n'importe qui peut déchiffrer et vérifier la signature à l'aide de la clé publique et de l'algorithme de hachage de l'émetteur. Étant donné les propriétés du chiffrement à clé publique et des algorithmes de hachage, le destinataire a la preuve que :

i. Le condensé a été chiffré à l'aide de la clé privée de l'émetteur ;
ii. Le message est protégé contre toute altération.

2) Chiffrement du message. Le chiffrement comprend les trois étapes suivantes :

a) Création de clés de chiffrement/déchiffrement uniques. Rappelons que les algorithmes de chiffrement et de déchiffrement utilisant des clés asymétriques sont trop lents pour les longs messages. Des algorithmes à clé symétrique sont utilisés car ils sont très efficaces ;

b) Chiffrement du message. La totalité du message (le message proprement dit et la signature) est chiffrée à l'aide de CSym, la clé symétrique évaluée ci-dessus ;

c) Chiffrement de la clé symétrique. CSym est également utilisée par le destinataire pour déchiffrer le message. Elle ne doit donc être accessible qu'au destinataire (Bob). Pour dissimuler CSym à tous sauf au destinataire, il suffit de la chiffrer à l'aide de la clé publique du destinataire. Étant donné que CSym représente un très petit élément d'information comparé au message (qui pourrait être très long), l'inefficacité relative des algorithmes à clé asymétrique devient acceptable.

Il est intéressant de souligner que si Alice voulait envoyer le même message à plusieurs destinataires, Bob et John par exemple, la seule opération supplémentaire qu'elle aurait à exécuter serait de répéter l'étape 2) c) pour John. Par conséquent, le message que Bob et John recevraient prendrait la forme suivante :
$[Message+[Condensé]_{CPrA}+CPuA]_{CSym}+[CSym]_{CPuB}+[CSym]_{CPuJ}$. Notez que Bob et John utiliseront exactement la même CSym pour déchiffrer le message.

6.4.4. Étapes de déchiffrement et de vérification de la signature d'un message

La Figure 12 ci-dessous montre la série d'opérations que Bob doit exécuter pour déchiffrer et vérifier le message envoyé par Alice.

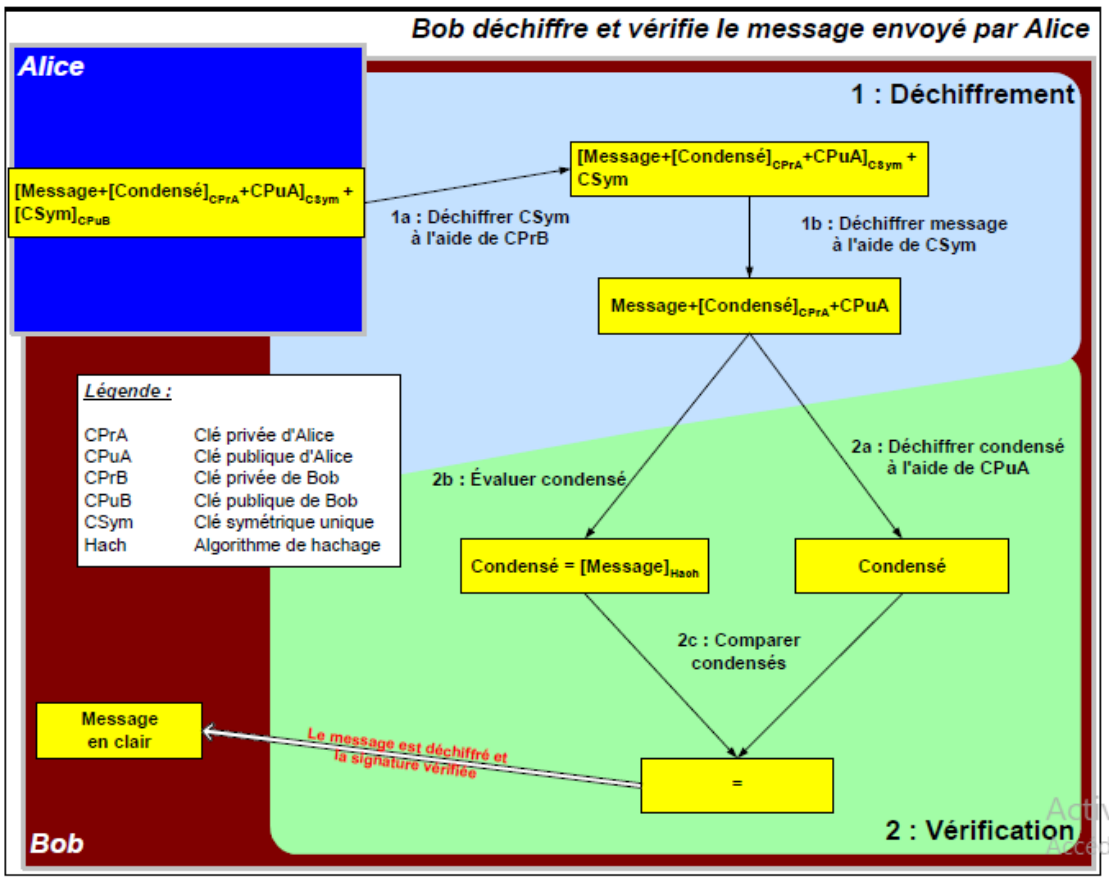


Figure 12. Processus détaillé de déchiffrement et de vérification à l'aide de clés

1) Déchiffrement du message. Le déchiffrement comprend les étapes suivantes :

a) Déchiffrement de la clé symétrique. La clé symétrique unique a été utilisée pour chiffrer le message. Cette clé (CSym) a été chiffrée à l'aide de la clé publique du destinataire (Bob). Seul Bob peut déchiffrer CSym et l'utiliser pour déchiffrer le message ;

b) Déchiffrement du message. Le message (qui comprend le message proprement dit et la signature) est déchiffré à l'aide de CSym.

2) Vérification de la signature. La vérification de signature comprend les trois étapes suivantes :

a) Déchiffrement du condensé de message. Le condensé a été chiffré à l'aide de la clé privée de l'émetteur (Alice). Le condensé est maintenant déchiffré à l'aide de la clé publique de l'émetteur incluse dans le message ;

b) Évaluation du condensé. Étant donné que le hachage est un processus unidirectionnel, autrement dit qu'il est impossible de retrouver le message d'origine à partir du condensé, le destinataire doit réévaluer le condensé en utilisant exactement le même algorithme de hachage que l'émetteur ;

c) Comparaison des condensés. Le condensé déchiffré en a) et le condensé évalué en b) sont comparés. S'ils concordent, la signature est de ce fait vérifiée et le destinataire peut alors avoir la certitude que le message a été envoyé par l'émetteur et n'a pas été altéré. S'ils ne concordent pas, il est possible que :

(i) le message n'ait pas été signé par l'émetteur ;
(ii) le message ait été altéré.

Dans les deux cas, le message doit être rejeté.

6.4.5. Cas d'utilisations

Les signatures numériques peuvent être appliquées à divers types de documents et certificats numériques. En tant que telles, elles ont plusieurs applications. Certains des cas d'utilisation les plus courants incluent : (32)

- Les Technologies de l'information, pour améliorer la sécurité des systèmes de communication Internet ;
- La Finance. Les signatures numériques peuvent être mises en œuvre pour les audits, les rapports de dépenses, les accords de prêt, et bien plus encore ;
- Le Juridique. Signature numérique de tous types de contrats entre entreprises et d'accords juridiques. De même pour les documents gouvernementaux ;
- La Sécurité sociale. Les signatures numériques peuvent agir en tant que prévention contre la fraude des prescriptions et des dossiers médicaux ;
- La blockchain. Les signatures numériques assurent que seulement le propriétaire légitime des fonds est en mesure de signer une transaction pour les transactions (tant que ses clés privées ne sont pas compromises).

6.4.6. Restrictions

Le plus grand défi pour les systèmes de signature numérique repose sur au moins trois exigences :

- L'Algorithme. La qualité des algorithmes utilisés dans un schéma de signature numérique est importante. Cela inclut le choix de fonctions de hachage fiables et de systèmes cryptographiques ;
- Implémentation. Si l'algorithme est bon mais que la mise en œuvre ne l'est pas, le système de signature numérique présentera probablement des défauts ;
- Clé privée. Si les clés privées sont divulguées ou compromises d'une quelconque manière, les propriétés d'authenticité et de non-répudiation seront invalidées. Pour les utilisateurs de crypto-monnaie, la perte d'une clé privée peut entraîner des pertes financières importantes.

6.5. Les certificats numériques

Les certificats numériques sont des documents électroniques servant à vérifier l'identité d'une entité numérique. Cette entité peut être un site web, un développeur de logiciels depuis le site web duquel vous téléchargez un produit ou même une personne avec qui vous désirez entrer dans une communication sécurisée. Les certificats numériques sont indispensables dans le monde moderne du commerce électronique, des services bancaires en ligne, du développement de logiciels et de presque toute sorte de partage d'informations sur Internet.

Certificat numérique ou électronique. C'est le résultat du processus d'établissement de la relation qui existe entre une clé publique, son propriétaire et l'application à laquelle la clé publique a été délivrée. (34)

- Pour une personne il assure son identité ;
- Pour une application, il assure que celle-ci n'as pas été détournée de ses fonctions ;
- Pour un site, il offre la garantie lors d'un accès vers celui –ci que l'on est bien sur le site auquel on veut accéder.

Le format reconnu actuellement est le format X509V3 (35). C'est un petit fichier, qui contient au moins les informations suivantes :

- Le nom de l'autorité (de certification) qui a créé le certificat ;
- Le nom et le prénom de la personne ;
- Son entreprise (CNRS par exemple) ;
- Son service (au CNRS, le nom du laboratoire) ;
- Son adresse électronique ;
- Sa clé publique ;
- Les dates de validité du certificat ;
- Des informations optionnelles ;
- Une signature électronique.

6.5.1. Types de certificats

Le type de certificats électroniques dépend du support qui l'héberge. Ainsi, nous pouvons distinguer trois types de certificats : (34)

- le certificat serveur ;
- le certificat personnel ou client ;
- le certificat IP SEC (Internet Protocol Security) ou VPN.

6.5.1.1. Le certificat serveur

Ce type de certificat est installé sur un serveur. Il est le garant de l'identité du serveur et la sécurité de la session établie par un utilisateur.

Le certificat serveur le plus répandu actuellement est le certificat SSL ou « Security Socket Layer ». Il permet d'assurer l'authenticité d'une URL et de garantir la sécurité des transactions effectuées par les internautes.

6.5.1.2. Le certificat personnel ou certificat client

Un certificat client est stocké dans un ordinateur ou dans conteneur tel qu'une clé USB ou une carte à puce appartenant à un particulier. Similaire à la carte d'identité d'une personne physique, elle permet d'identifier un utilisateur et de définir des droits d'accès à diverses informations partagées sur un réseau.

6.5.1.3. Le certificat IP SEC (Internet Protocol Security) ou VPN

Le certificat IPSEC est quant à lui hébergé sur un équipement réseau. Il a pour but de chiffrer l'ensemble des informations transmises sur les réseaux usant des protocoles internet. Comme les deux précédents certificats, il sert également d'identifiant pour le composant de réseau. Il permet ainsi de rendre privé l'ensemble des flux transitant entre deux équipements réseaux.

6.5.2. Autorités de certification et infrastructure de gestion de clés

Le certificat permet d'établir une relation de confiance entre l'émetteur et récepteur. Mais où peut-on les acquérir ? Qui les crée ? Avec quelles informations ? Comment sont-ils gérés ? (34)

6.5.2.1. Autorité de certification

Une autorité de certification (AC) est un organisme reconnu comme étant compétent pour délivrer des certificats à une population auprès de laquelle elle a toute confiance et en assurer la validité. Elle s'engage sur l'identité d'une personne au travers du certificat électronique qu'elle lui remet. Une autorité de certification est responsable (vis-à-vis de ses clients, mais aussi de toute personne se fiant à un certificat électronique qu'elle a émis) de l'ensemble du processus de certification et, par voie de conséquence, de la validité des certificats qu'elle émet. Par ailleurs, c'est elle qui définit la politique de certification et la fait appliquer.

6.5.2.1.1. Quelle autorité de certification

De même qu'une carte d'identité ou qu'un passeport, un certificat est délivré par une autorité, que l'on qualifie « de certification ».

Mais on est dans le monde de l'Internet où tout est électronique et planétaire, sans frontière ni gouvernement. Ainsi, techniquement, cette autorité peut être n'importe quelle association, société, organisme, individu, ... Actuellement il n'existe pas de gouvernement qui délivre des certificats, ni d'organisations structurées et indépendantes comme celles qui affectent les numéros IP ou les noms de domaine. Par contre de nombreuses sociétés commerciales se sont

déjà lancées dans ce commerce qui s'annonce très lucratif. Ce sont elles qui vendent les certificats.

La confiance que l'on accordera à un certificat va dépendre du sérieux de l'autorité qui l'aura délivré. De plus, on voit très bien le risque encouru par une entreprise ou un organisme dont la carte d'identité des employés aurait été créée par une autorité ni habilitée, ni contrôlée par elle-même. Le choix de l'autorité de certification dans une organisation ou une entreprise est une décision stratégique

6.5.2.2. Infrastructure de gestion de clé

Quelle que soit l'autorité de certification choisie, il faut faire d'autres choix. Comme il existe un circuit de procédures et de vérifications, des personnes habilitées, etc., pour délivrer les cartes d'identité, il faut mettre l'équivalent en place. Il faut ainsi décider qui va recueillir et vérifier les informations données par une personne lorsqu'elle va demander un certificat, suivant quelles procédures, qui va créer le certificat, qui va le lui délivrer, pour quelle durée, où va-t-il être stocké, où va-t-on pouvoir récupérer les certificats d'autres personnes, etc. (34)

Il faut définir ce que l'on appelle une architecture de gestion des certificats. IGC (Infrastructure de Gestion de Clés), et PKI (Public Key Infrastructure) sont les deux sigles les plus connus pour la désigner. Les normes internationales décrivent les différents éléments fonctionnels d'une IGC. En simplifiant, l'architecture est constituée de :

6.5.2.2.1. Autorités d'enregistrement

L'autorité d'enregistrement vérifie l'identité du demandeur, s'assure que celui-ci possède bien un couple de clés privée-publique et récupère la clé publique du demandeur. Elle transmet ensuite ces informations (informations d'identité du demandeur ainsi que sa clé publique) à l'autorité de certification.

6.5.2.2.2. Autorité de certification

Celle-ci reçoit les demandes de création de certificats venant des autorités d'enregistrement. Elle vérifie la validité de la signature des messages reçus, garantie de l'intégrité de la demande et de l'authentification des émetteurs. Elle crée les certificats et signe ces certificats en utilisant sa clé privée. Elle envoie les certificats aux utilisateurs et en parallèle les transmet au service de publication. Une autorité de certification a donc un couple de clé privée-publique pour signer les certificats.

6.5.2.2.3. Service de publication

Celui-ci rend disponible les certificats émis par l'autorité de certification. Il publie aussi la liste des certificats valides et des certificats révoqués. Concrètement ce service peut-être rendu par un annuaire électronique LDAP ou un serveur Web accessibles par l'Internet.

Bibliographie

1. **Laprie, J.C.** *Guide de la sûreté de fonctionnement.* s.l. : Cépaduès, 1996. p. 380.

2. **Grevisse, YENDE RAPHAEL.** SUPPORT DE COURS DE SÉCURITÉ. Butembo, Congo-Kinshasa : s.n., 2018.

3. CHAPITRE 1 Introduction à la sécurité informatique. *Protection et Sécurité des Systèmes Informatiques.* Annaba : s.n., 2020.

4. **FERRAG, Mohamed Amine.** *Sécurité.* Guelma : s.n., 2018.

5. *Le Grand Livre de Securiteinfo.com.* Paris : s.n., 2004.

6. **Shirey, R.** *RFC 2828–Internet security glossary, 2000.* [En ligne] 2003. http://www. faqs.org/rfcs/rfc2828. html.

7. https://www.ncsc.gov/nittf/docs/CNSSI-4009_National_Information_Assurance.pdf. [En ligne]

8. **DOUAS, Youssef CHAIKHI.** *Les Types D'attaques Informatique.* 2010.

9. **Poinsot, Laurent.** Chap. I : Introduction à la sécurité informatique. Paris, France : s.n., 2009.

10. *CrowdStrike.* [En ligne] 2020. https://www.crowdstrike.com/resources/reports/global-attitude-survey-2020/.

11. **Suzanne, Widup., Alex, Pinto., David, Hylender., Gabriel, Bassett., philippe, langlois.** *2021 Data Breach Investigations Report.* 2021.

12. **Grevisse, YENDE Raphael.** *Support de cours de sécurité informatique et crypto.* Butembo : s.n., 2018.

13. *Histoire du chiffrement et de ses méthodes SYNTHÈSE CHRONOLOGIQUE DU CHIFFREMENT À TRAVERS LES ÂGES.* thawte. 2013.

14. Cryptographie-Cryptage par transposition. [En ligne] https://web.maths.unsw.edu.au/~lafaye/CCM/crypto/transpo.htm.

15. **Riguidel, M.** Quelque rappels sur les techniques cryptographiques. 2002.

16. **Ghislaine, L.** *Introduction à la cryptologie.* 1998.

17. **Chassé, Guy.** Cryptographies Mathématiques. 2010.

18. *Origins of cryptology: The Arab contributions.* **Al-Kadi, Ibrahim.A.** 1992, Cryptologia, pp. 97-126.

19. **BELFEDHAL, Alaa Eddine.** Etude et Implémentation des Fonctions de Hachage Cryptographiques Basées sur les Automates Cellulaires. [éd.] Université Djillali Liabès. [Thèse de Doctorat]. Sidi-Bel-Abbès, Algérie : s.n., 2015.

20. **Boura, Christina.** Analyse de fonctions de hachage cryptographiques. [PhD Thesis]. Paris, France : s.n., 2012.

21. **Knuth, D.E.** *Seminumerical Algorithms, The Art of Computer Programming.* 1981. Vol. 2.

22. *Keying hash functions for message authentication.* **Mihir, Bellare., Ran, Canetti., Hugo, Krawczyk.** Santa Barbara : Springer, 1996. 16th Annual International Cryptology Conference. pp. 194-203.

23. **Leurent, Gaëtan.** Construction et Analyse de Fonctions de Hachage. [PhD Thesis]. 2010.

24. **Thomas, Peyrin.** Analyse de fonctions de hachage cryptographiques. [PhD Thesis]. Versailles, Yvelines, France : s.n., 2008.

25. **Fuhr, Thomas.** Conception, preuves et analyse de fonctions de hachage. [PhD Thesis]. Paris : s.n., 2011.

26. *A model and architecture for pseudorandom generation with applications to /dev/random.* **Barak, Boaz., Halevi, Shai.** New York : s.n., 2005. CCS '05: Proceedings of the 12th ACM conference on Computer and communications security. pp. 203-212.

27. **Langlois, Julie.** Une fonction de hachage basée sur la théorie de chaos. 2011.

28. **Peyrin, Thomas.** Analyse de fonctions de hachage cryptographiques. [PhD Thesis]. Yvelines, France : s.n., 2008.

29. **Klensin, J., Catoe, R., Krumviede, P.** Imap/pop authorize extension for simple challenge/response. RFC 2195. [éd.] RFC Editor. United States : s.n., 1997.

30. **Rivest, R.** Rfc 1321 : The md5 message digest algorithm. [éd.] RFC Editor. 1992.

31. **NIST, National Institute of Standards and Technology.** Fips 180 : Secure Hash Standard (SHS). 1993.

32. Qu'est-ce qu'une signature numérique? *Binance Academy.* [En ligne] https://academy.binance.com/fr/articles/what-is-a-digital-signature.

33. *Cryptographie à clé publique et signature numérique. Principes de fonctionnement.* Montréal : s.n., 2002.

34. **Archimbaud, Jean-Luc.** LES PRINCIPES TECHNIQUES DES CERTIFICATS ÉLECTRONIQUES. [éd.] Lavoisier. [Article]. 2003. Vol. 4, pp. 101-110.

35. **Housley, R., Ford, W., Polk, W., Solo, D.** *Internet X.509 Public Key Infrastructure Certificate and CRL Profile.* Network Working Group. 1999.

36. **Grevissse, YENDE Raphael.** *SUPPORT DE COURS DE SÉCURITÉ INFORMATIQUE ET CRYPTO.* Butembo : s.n., 2018.

37. [En ligne] http:// www.webauvages.net.

38. **Thomas, Fuhr.** Conception, preuves et analyse de fonctions de hachage. [PhD Thesis]. Paris, France : s.n., 2011.

39. **Gaëtan, Leurent.** Construction et Analyse de Fonctions de Hachage. [PhD Thesis]. 2010.