

Caractérisation d'une attaque logicielle sur un système informatique

Lexique

- Tout acte sur un système visant à nuire à ses propriétés peut être qualifié de ***malveillant***.
- Tout acte de ***malveillance*** sur un système (ou encore attaque sur un système) comme une combinaison d'actions qui ciblent ce système et dont l'intention est de nuire au moins à l'une des propriétés de disponibilité, d'intégrité ou de confidentialité.
- Un système informatique est la composition de deux systèmes, l'un matériel et l'autre logiciel vérifiant les propriétés suivantes:
 - le système logiciel appartient à l'état du système matériel ;
 - la structure du système matériel est apte à interpréter le système logiciel.

Action des attaques

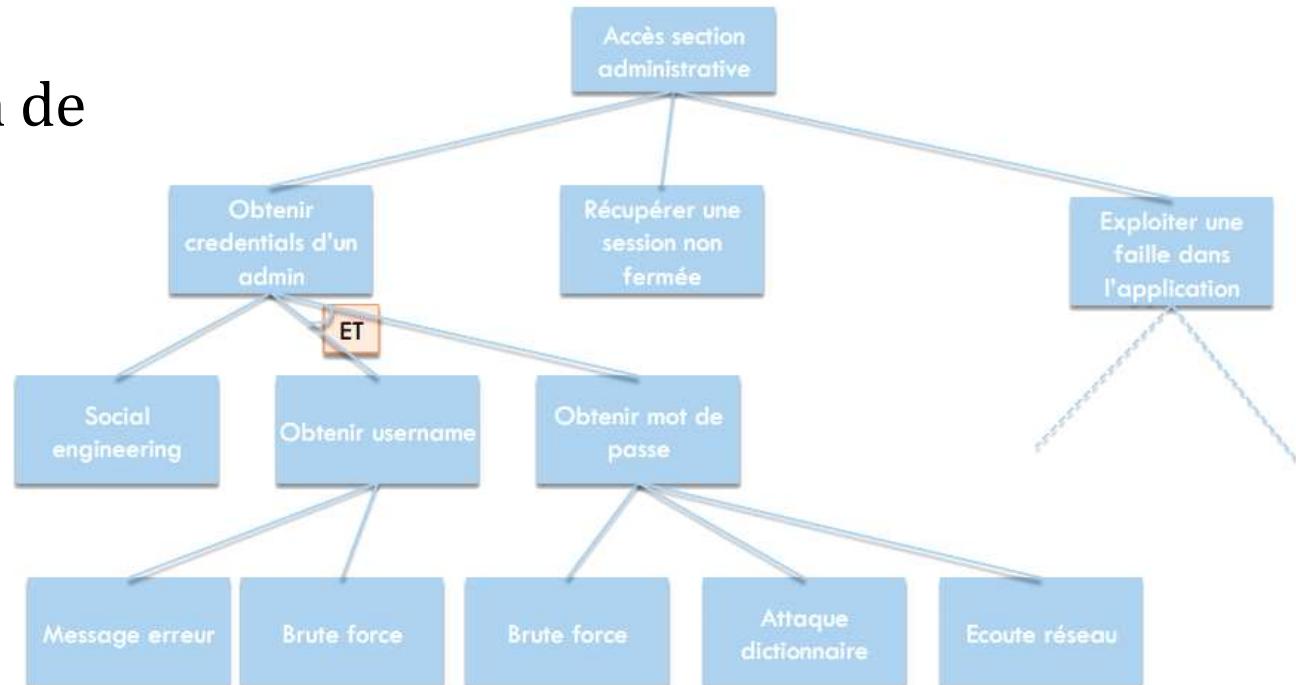
- une attaque impactant un système informatique, agit :
 - soit sur l'état du système matériel et de même agit sur la structure ou l'état du système logiciel ;
 - soit sur la structure du système matériel.
- Si l'on se restreint aux attaques logiques, une action sur la structure du système matériel n'est généralement possible que si ce dernier est un système adaptable par logiciel (par exemple, un système à base de FPGA).
- Quoique des fois, une attaque logique peut entraîner indirectement une altération de la structure du système matériel.
 - Phénomènes physiques liés à la température, aux champs électromagnétiques, etc.

Le problème de l'attaquant

- Une attaque logique sur un système peut être vue comme l'expression de la solution d'un problème particulier contenant un objectif de malveillance sur un système ou sur un autre système en relation avec ce dernier.

Le problème de l'attaquant

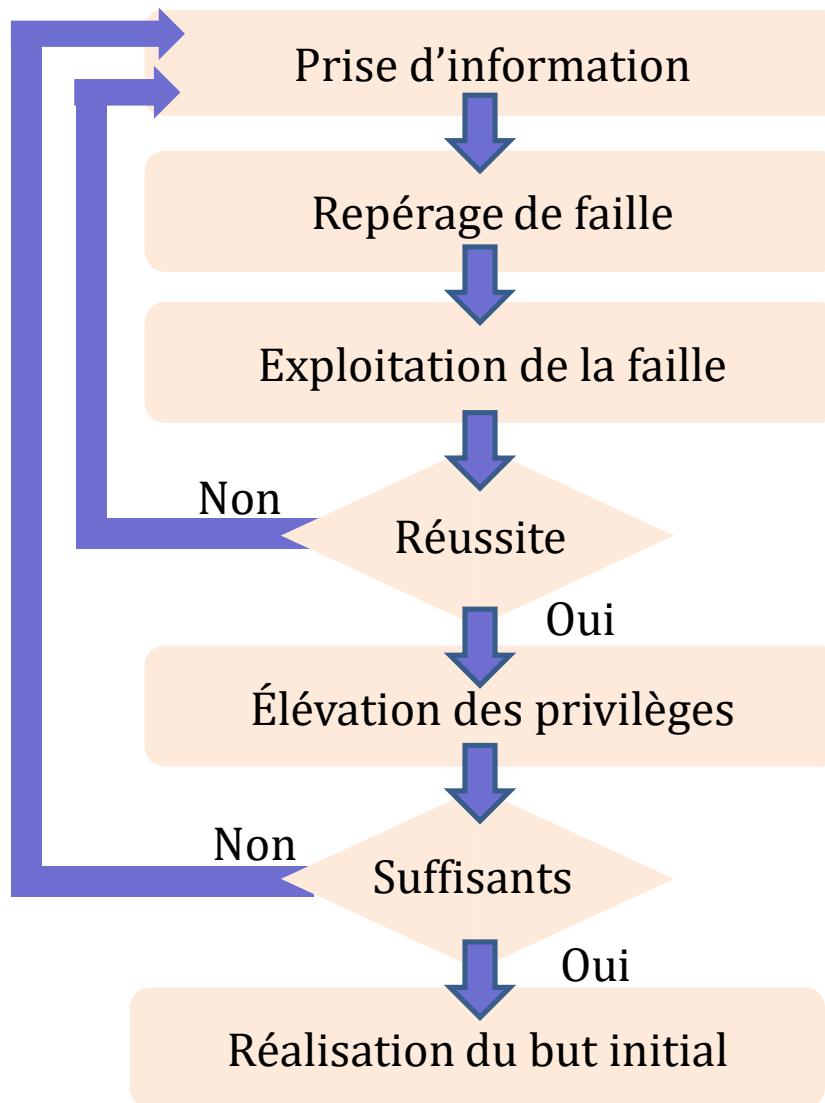
Décomposition en de multiples sous-problèmes (par exemple Arbres d'attaques, Bruce Schneier dans Dr. Dobb's Journal).



Le problème de l'attaquant

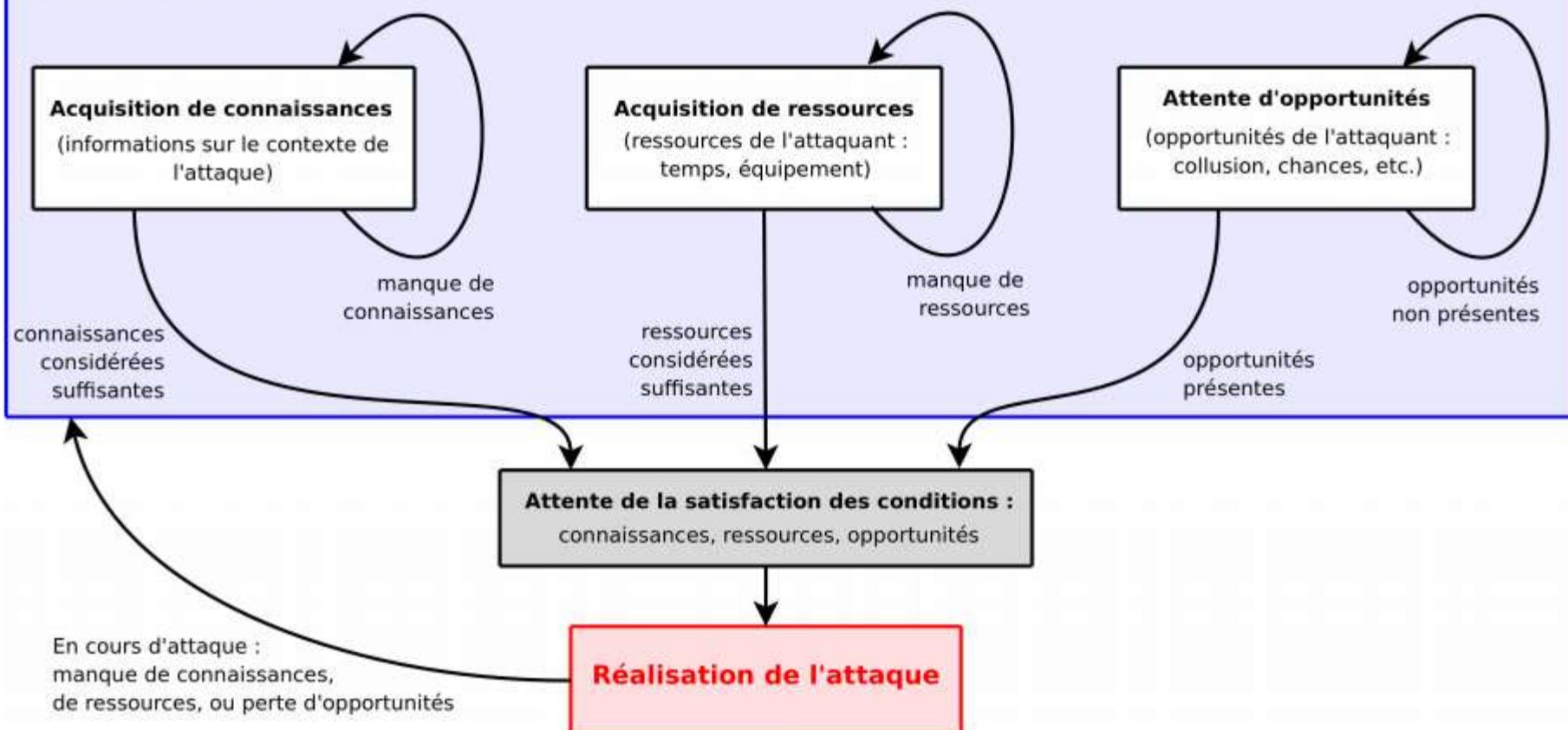
- Objectifs
 - récupérer des informations contenues dans ou transitant par le système ; (**interception**)
 - introduire des informations fallacieuses dans le système ; (**modification**)
 - bloquer l'accès au système, c'est-à-dire provoquer un déni de service (DoS) (la notion de système incluant ici le réseau) ; (**saturation**)
 - détourner des services du systèmes ;
 - prendre le contrôle du système afin de l'utiliser pour espionner ou pour le transformer en serveur de contenus illégaux ; (**usurpation d'identité**)
 - rebondir vers d'autres systèmes, auquel cas il sert uniquement de point de transfert vers une autre cible.

Algorithme de l'attaquant



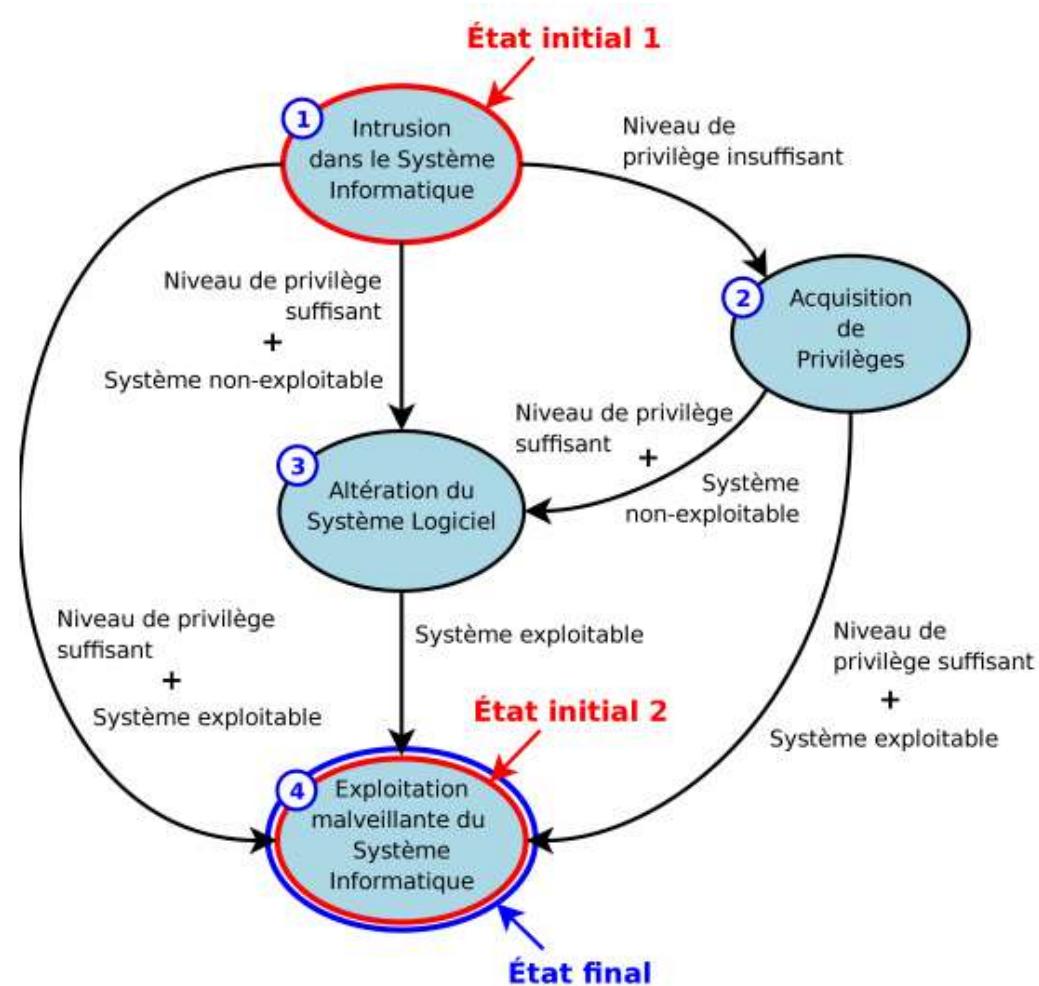
Résolution du problème de l'attaquant

Préparation de l'attaque



Modèle générique d'attaque logique

- Il est possible de décomposer toute attaque à partir d'un nombre restreint de blocs sémantiques
- Nœuds du graphe : différents blocs sémantiques, les étapes fondamentales d'une attaque
- Une solution est ainsi caractérisée par le chemin qui débute par l'un des deux états initiaux et se termine à l'état final
- L'attaque qui correspond à cette solution, est dite **réussie** dès lors que tous les nœuds du chemin emprunté sont franchis avec **succès**.



Intrusion dans le système informatique

- Une intrusion dans un système informatique est le fait pour un système externe d'y pénétrer alors que sa présence n'y est pas autorisée.
- Il s'agit dans le cas de la sécurité, d'un système malveillant appelé "attaquant"
- Modélités :
 - **Usurper l'identité d'un utilisateur légitime du système** (vol d'identifiant et de mot de passe, attaque par dictionnaire, etc.)
 - **exploiter une faille du système logiciel** qui lui permet d'accéder aux services du système informatique
- Exemple
 - Le point de départ d'une attaque qui installe un *sniffer* sur un système informatique complexe. L'attaquant souhaite alors récupérer par exemple des mots de passe d'utilisateurs légitimes du système. Bien que le travail du *sniffer* ne requiert aucune intrusion sur le système (il effectue une écoute passive), son installation nécessite une intrusion préalable.

Acquisition de privilèges

- Étape associée à l'état dans lequel se trouve un attaquant lorsque les privilèges dont il dispose sur un système informatique sont insuffisants pour la réalisation de son objectif
- L'attaquant tente d'acquérir les privilèges qui lui sont nécessaires au travers de failles existantes dans le système informatique dans lequel il a pénétré
- Exemple :
 - un attaquant ayant usurpé l'identité d'un utilisateur légitime du système peut avoir besoin des privilèges d'un administrateur du système pour réaliser son objectif. Il peut également nécessiter des privilèges encore supérieurs comme ceux associés à un noyau de système d'exploitation

Altération du système logiciel

- Étape associée à l'état dans lequel se retrouve l'attaquant lorsque le système dans lequel il a pénétré ne dispose pas de la structure adéquate à la réalisation de son objectif. Toutefois, l'attaquant dispose des privilèges nécessaires pour y remédier Il s'agit alors pour l'attaquant, de modifier la structure et/ou l'état du système logiciel afin que ce dernier soit apte à la réalisation de l'objectif de l'attaquant.
- Une altération du système logiciel par un attaquant peut être vue comme l'ajout (ou encore l'installation) d'un ou de plusieurs sous-systèmes malveillants au système logiciel, soit de l'altération d'informations qui impactent la sécurité du système
- Ces altérations se groupent en 4 catégories :
 - Programmes autoreproducteurs
 - Infections informatiques
 - Rootkit
 - Altération d'informations qui impactent la sécurité du système

Programmes autoreproducteurs

- Définition d'un programme autoreproducteur (Fred Cohen et Leonard Adleman)

“Un *virus* est une séquence de symboles qui, interprétée dans un environnement donné (adéquat), modifie d'autres séquences de symboles dans cet environnement, de manière à y inclure une copie de lui-même, cette copie ayant éventuellement évolué.”

- Cette définition est assez générale pour regrouper l'ensemble des programmes autoreproducteurs
- Deux classes de programmes autoreproducteurs :
 - les virus
 - les vers

Virus

- Un virus est un segment de programme qui, lorsqu'il s'exécute, se reproduit en s'adjoignant à un autre programme. L'analogie avec les virus biologiques tient à leur comportement
 - un virus biologique ne peut se reproduire par lui même ; pour se reproduire, il doit modifier le code génétique des cellules qu'il infecte pour que les cellules ainsi modifiées produisent des copies du virus ;
 - De même un virus informatique ne peut être activé (et donc se reproduire) que par l'exécution d'un programme porteur du virus



Types de virus

- **Exécutables** : « virus de remplacement » (écrasent des programmes avec eux-mêmes), « virus parasites » (s'attachent à des programmes, en tête, en queue, ou dans des cavités).
- **Résidents** : caché en RAM en permanence.
Redirection de trap, attendre un exec etc.
- **Boot sector** : exécutés avant même le chargement du système. Point de départ fréquent des virus résidents.
- **Pilotes** : chargés « officiellement » par le système, exécutés en mode noyau.
- **Macros / scripts** : VB dans Office, ELisp dans Emacs etc. Transmission par mail croissante. Peu déqualifications requises.
- **Source** : contaminent les sources plutôt que les exécutables.

Propagation des virus

- Freewares, sharewares sur le web
- Floppies, zips, clés USB etc.
- Internet, LAN etc.
- Mails news (attachements, carnets d'adresses)
- Plugins pour les navigateurs
- etc.

Exemple de virus

- **1986 > Brain : le premier virus diffusé**
 - Premier virus informatique à avoir attaqué massivement des ordinateurs, Brain **contamine les disquettes 5 pouces 1/4 uniquement** .
- **2004 > Cabir traque les terminaux mobiles**
 - Utilisant la connexion Bluetooth pour se répandre, **Cabir** reste cependant un virus à but expérimental. Il **cible le très répandu système d'exploitation Symbian OS**, équipant un smartphone sur deux dans le monde à l'époque(source : Gartner).
- **Love Bug**
 - Love Bug est probablement le plus célèbre des virus. En feignant d'être un billet doux, il a joué sur la curiosité des utilisateurs, se propageant en quelques heures sur l'ensemble de la planète.
 - La version originale du virus envoie un mail avec l'objet “I LOVE YOU” et le texte “kindly check the attached love letter coming from me”. **L'ouverture de la pièce jointe permet au virus de s'exécuter**. Si vous avez installé Microsoft Outlook, le virus s'en sert pour tenter de s'expédier lui-même à l'ensemble des contacts de votre carnet d'adresses. Il peut aussi se diffuser tout seul vers d'autres utilisateurs de forums de discussion, subtiliser des renseignements sur l'utilisateur infecté et écraser certains fichiers.

Remède anti-virale

- Scanners : comparaison de tous les exécutables avec une base de données. Mises à jour régulière.
 - Recherche floue (plus coûteuse, risque de fausses alarmes).
 - Préserver les dates originales des fichiers infectés, leur longueur (compression), se différencier des bases de données (encryption)
 - La procédure de déchiffrement ne peut pas être encryptée
 - Virus polymorphes (moteur de mutation)
- Vérificateurs d'intégrité : calcul (puis comparaison) de sommes de contrôle à partir d'un état sain
 - Écraser les sommes avec les nouvelles
 - Encrypter les sommes (avec une clé externe)
- Vérificateurs de comportement : antivirus résidents. Travail difficile.
- Détection d'intrusion

Antivirus

- Un bon antivirus
 - détecte, identifie et éradique les virus connus
 - gère les virus inconnus qui utilisent des techniques connues
 - ne génère pas (ou peu) de fausses alertes et ne nécessite pas d'intervention de l'utilisateur
 - s'exécute sans ralentir le système
 - s'installe et se désinstalle facilement et facile à configurer est pré-configurer de façon pertinente
- En pratique, c'est de plus en plus difficile même si, , ca pourrait être mieux :
 - les éditeurs d'antivirus "s'opposent" à la recherche dans le domaine
 - Les produits privilégient la rapidité d'exécution à la sûreté
 - l'ingénierie sociale permet de contourner les outils
 - l'utilisateur final ne sait pas réagir à une alerte
- L'état du marché est assez déprimant :
 - les meilleurs ne sont pas les plus connus (Avast/Alwil Software, AVG/Grisoft, FSecure, Kaspersky/KAV, NOD32)
 - Unix/Linux très en retard sur Windows (mais moins de virus).
 - Existe Anti-virus pour linux pour protéger des ordinateurs: Avg, AntiVir, BitDefender,...

Antivirus



List of 10 Best Antivirus Software For 2022

- 01 Norton Security Premium
- 02 AVG Internet Security
- 03 Trend Micro Maximum Security
- 04 WebrootSecure AnywhereAntivirus
- 05 ESET NOD32 Antivirus
- 06 Kaspersky Anti-Virus
- 07 Bitdefender Antivirus Plus 2022
- 08 Comodo Antivirus
- 09 Windows Defender Security Center
- 10 Avira Antivirus

Contrôle d'intégrité

- Vérifier que le système est conforme à ce qu'il devrait.
- Essentiellement une technique : prise d'empreintes et comparaisons :
 - on prend une empreinte de chaque fichier (système, pas forcément les images, quoique...) du système sain
 - on met à jour la base d'empreintes à chaque modification légitime
 - une modification illégitime est détectable par changement de l'empreinte
- Comment savoir que le système est sain lorsqu'on prend/met à jour une empreinte ? Méthodologie : utiliser une machine de référence, isolée et coupée du réseau sauf lors des mises à jour.
- Sécurité de la base d'empreintes : certains virus modifient la base d'empreinte des antivirus les plus répandus pour cacher leur infection.

Détection d'intrusions

- Très proche des antivirus :
 - même techniques de détection (mêmes limites)
 - contrôle d'intégrité (mêmes difficultés)
 - limites fortes en terme de ressources système : l'analyse du réseau nécessite de surveiller simultanément des milliers de connexions sur des centaines de protocoles différents

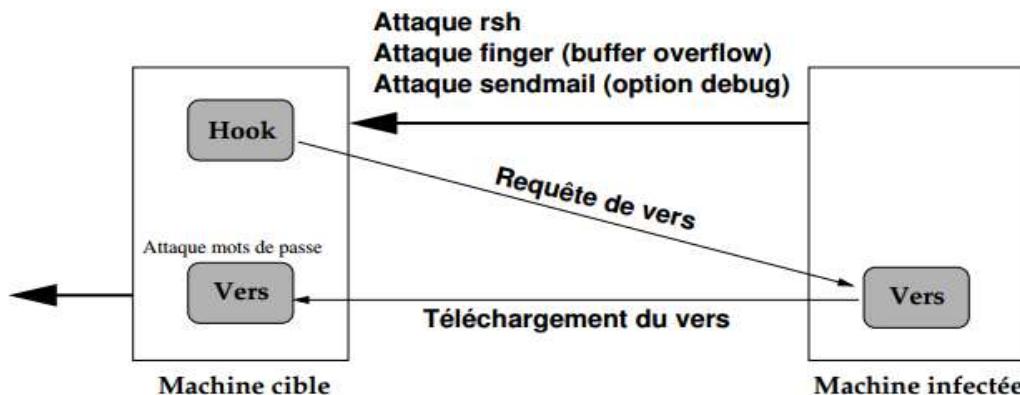
Vers



- Un vers (worm en anglais) est un programme autonome qui se propage sur les réseaux, se reproduit et s'exécute à l'insu des utilisateurs normaux.
- Ses caractéristiques sont
 - Consommation (voire épuisement) des ressources systèmes.
 - Dissémination à travers les réseaux informatiques.

Exemple de vers

- **1988 > Morris Worm contamine Internet**
- Lancé le soir du 2 Novembre 1988, détecté pour cause de DoS, solutions proposées le 3, neutralisé en quelques jours.
- Le virus est en fait l'un des tout premiers vers écrits pour Internet. Lancé en 1988 par son créateur, Robert Tappan Morris, depuis le MIT où il étudie, **le virus montre rapidement la vitesse de propagation élevée qu'offre Internet** aux créateurs de virus.
- Le ver s'appuie sur des **vulnérabilités présentes sur certains systèmes Unix où les mots de passe se révèlent relativement simples à forcer**. Il s'installe alors sur le système et prend des ressources machines à l'utilisateur. Le virus pouvant se réinstaller sur un système déjà contaminé, il cause d'importantes perturbations.



Infections informatiques (1/2)

- Une infection informatique est le résultat de l'installation dans un système d'information, à l'insu du ou des utilisateurs, d'un programme à caractère offensif, en vue de porter atteinte à la sécurité de ce système
- **Bombe logique** : Une bombe logique est une infection qui attend un évènement (date, action, données particulières, etc.) appelé en général "gâchette" pour exécuter sa fonction offensive.
- **Cheval de Troie** : Un Cheval de Troie est directement issue de la mythologie grecque, plus précisément de la guerre de Troie.
 - Un cheval de Troie, depuis lors désigne un artifice quelconque qui entraîne une cible à "inviter" un ennemi dans un endroit sécurisé.
 - Dans le contexte de la sécurité informatique, un cheval de Troie, désigne un système réalisant une fonction a priori souhaitée par un utilisateur mais qui réalise une fonction pour l'attaquant. Il vise de part sa nature à leurrer sa victime.

Bombe logique

- Le passage informatique à l'an 2000 (couramment appelé **bug de l'an 2000**, ou bogue de l'an 2000) a suscité de sérieuses inquiétudes à cause de problèmes de conception et de programmation portant sur le format de la date dans les mémoires des ordinateurs et par conséquent dans les matériels informatiques, ainsi que dans les logiciels.
- Dans de nombreux programmes liés à des bases de données ou à des fichiers avec procédures de tris internes à l'exécution, il manquait les deux chiffres 19 correspondant au siècle, de sorte qu'au passage de 1999 à 2000, de nombreux dysfonctionnements devaient se produire dans ces traitements informatiques aboutissant à l'année 1900 au lieu de 2000.
- Les ordinateurs n'étaient ni conçus matériellement ni programmés pour passer à l'an 2000 et ils affichaient quel qu'en soit le moyen — imprimante ou écran — 1900 à la place de l'année 2000 en cours, qu'il aient des logiciels compilés ou interprétés. Seuls les ordinateurs n'inscrivant une année que sous 2 caractères n'étaient pas concernés (par ex. inscription d'une date limite de consommation sur un objet alimentaire).

Infections informatiques (2/2)

- **Porte dérobée** : Une porte dérobée (Backdoor/Trapdoor) est un moyen de contourner les mécanismes de contrôle d'accès. Il s'agit d'une faille du système de sécurité qui provient soit d'une faute de conception (volontaire ou accidentelle), soit d'une altération du système durant sa phase opérationnelle.
- **Spyware** : Un spyware est une infection dont l'objectif est de divulguer à un attaquant des informations issues du système infecté.
- **Adware** : Un adware est une infection dont l'objectif est d'envoyer à l'utilisateur du système infecté de la publicité ciblée. Pour cela, il espionne les habitudes de l'utilisateur.

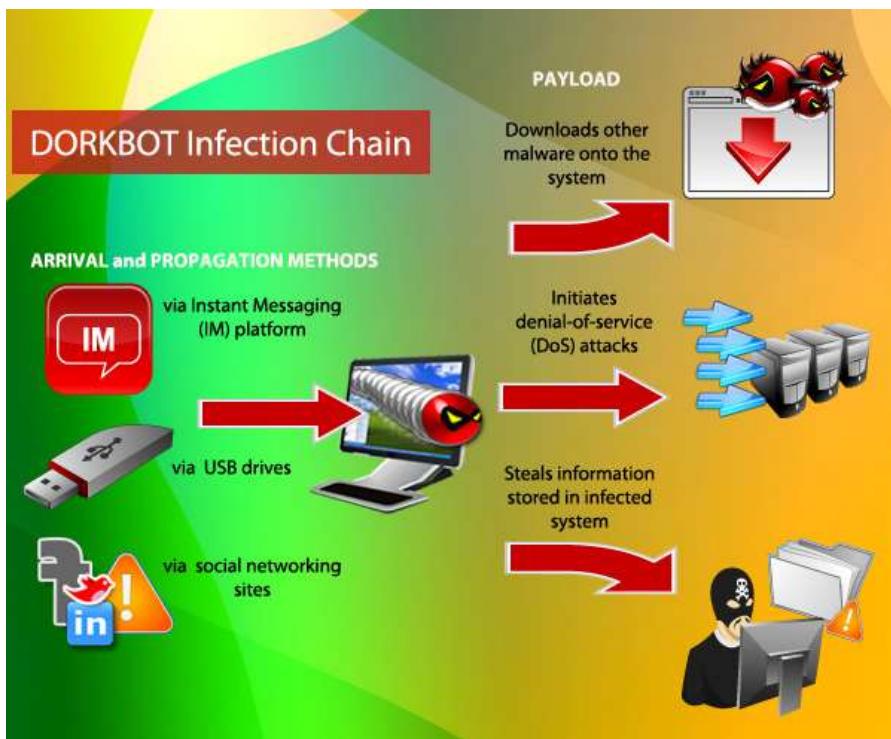
Cheval de Troie

- **Dorkbot, le virus de Skype (Chat) (2015)**



- Le Cheval de Troie [Dorkbot](#) est un parfait exemple de virus "social" [Dorkbot](#). Ce malware est une variante du ver Dorkbot, un *ransomware* qui passe par une porte dérobée, s'infiltre sur votre PC et bloque tout accès à vos données personnelles. Tout rentrera dans l'ordre si vous versez la somme de 200\$ dans les prochaines 48 heures...
- Le **mode de contamination** est toujours le même. L'un de vos contacts vous interpelle sur Skype pour rire de votre nouvelle photo de profil. Le message original en anglais est le suivant:

Cheval de Troie



Rootkit

- Un rootkit est un système parasite permettant à un attaquant de maintenir dans le temps un contrôle frauduleux sur un système informatique.
- L'accent est mis sur le maintien du contrôle du système infecté. Ainsi, un rootkit est caractérisé notamment par son invisibilité, sa robustesse et son pouvoir de nuisance.

Rootkit

- **Sirefef, le rootkit qui transforme votre PC en zombie (Botnet)**
- **Sirefef**, connu aussi sous le nom de [ZeroAccess](#) ou encore **Max++**, est un virus dont le but principal est de transformer en zombie le PC infecté en le reliant à un nœud d'un immense réseau d'ordinateurs (botnet).
- Par le biais du réseau infecté, l'objectif est de gagner de l'argent en cliquant sur des annonces ou d'installer de faux antivirus moyennant rétribution pour des nettoyages immédiats et soit-disant miraculeux.
- L'utilisateur ne remarque rien, mais l'utilisation de la connexion est constante, et peut atteindre plus de 32 Go par mois (l'équivalent de 45 films). Et dans le pire des cas, le PC zombie peut participer à des actions de cyber guerre.

Altération d'informations qui impactent la sécurité du système

- Ces altérations peuvent s'appliquer à la fois à la structure du système et à son état. Nous donnons ci-dessous une liste non exhaustive de telles actions :
 - suppression de mécanismes de protection ;
 - inhibition des mécanismes de l'audit ;
 - suppression des journaux d'activités du système ;
 - etc.

Outils de protection

- antivirus : protège des virus, vers, etc.
- pare-feux : contrôle les connexions réseaux
- anti-espion : cherche les spywares, backdoors, rootkits, etc. ⇒ comme antivirus
- contrôle d'intégrité : vérifie si le système a été altéré
- détecteur d'intrusions : cherche la présence d'un intrus

Outils de cryptographie et d'authentification

Authentification

Cryptographie symétrique

Cryptographie asymétrique

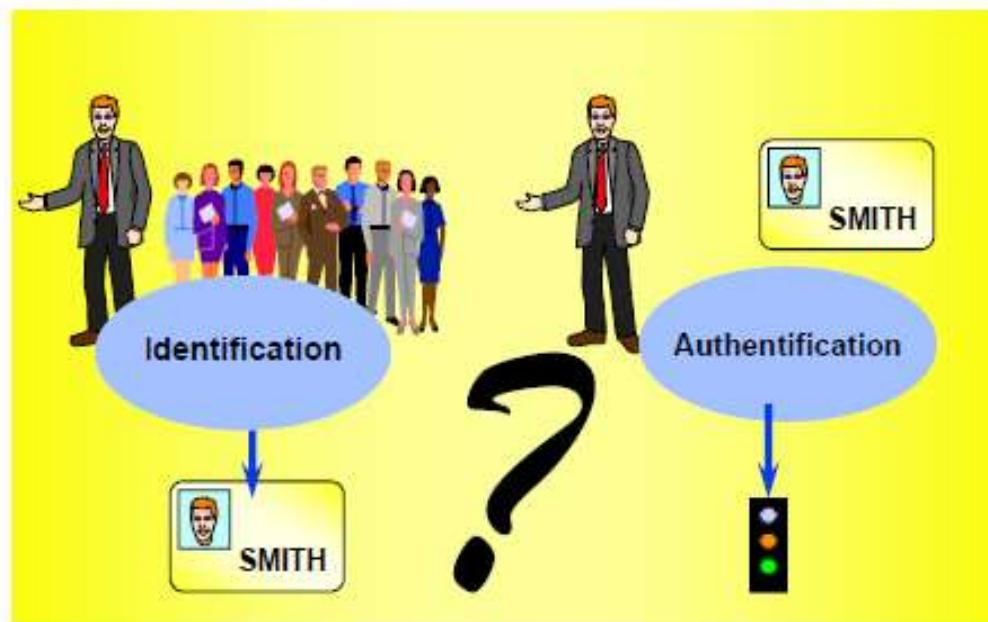
Hachage

Signature numérique

Certificats

Qu'est ce qu'une authentification?

- **S'authentifier**, c'est apporter la preuve de son **identité**. Mais sur quels principes se base cette notion d'identification ?
- L'authentification intervient après une phase d'identification qui consiste à établir l'identité annoncée par l'utilisateur. On parle souvent de la phase d'identification – authentification ou I/A.



Phases d'authentification

- Identification : Identité numérique \leftrightarrow Personne
 - Chaque personne ayant accès au système d'information doit se voir attribuer un **IDENTIFIANT UNIQUE**
- L'authentification = Challenge
 - L'authentification repose toujours sur un challenge lancé par le système de destination à l'intention du client afin que celui-ci prouve son identité



Dupond

➤ Mot de passe, clef privée, empreinte digitale, ...



= 1 identifiant



- La réussite du challenge valide l'association client \leftrightarrow Identifiant



Mon mot de passe
est toto



Bases de l'authentification

- Il existe plusieurs moyens pour s'authentifier :
 - Le secret (ex : mot de passe)
 - L'objet (ex : jeton)
 - Le caractère (ex : biométrie)
 - Le savoir faire (ex : signature manuscrite)

La combinaison de plusieurs méthodes d'authentification permet de mettre en place des solutions **d'authentification fortes**.



Mots de passe

- Le mot de passe reste le mécanisme de déverrouillage le plus répandu car il ne nécessite pas d'ajout de dispositif technique.
- Les risques principaux liés à l'utilisation du mot de passe sont sa **divulgation** et sa **faiblesse**.
 - Il faut que son domaine d'utilisation soit le plus restreint possible pour limiter le risque de divulgation.
 - Il est préférable d'associer à chaque usage un mot de passe différent.
 - Un bon mot de passe doit être construit de manière à ne pas être trop sensible aux techniques de craquage connues.
 - Ex: 1 heure pour craquer un mot de passe de 8 caractères alphabétiques [A-Z] Vs 1 mois pour craquer un mot de passe de 10 caractères alphabétiques ou un mot de passe de 8 caractères alphanumériques [A-Za-z0-9].

Mots de passe

- Il existe des moyens techniques pour fabriquer et retenir des mots de passe forts.
 - La méthode phonétique : utiliser les sons de chaque syllabe pour fabriquer une phrase facile à retenir. Par exemple, la phrase « j'ai acheté huit cd pour cent euros cet après-midi » deviendra ght8CD%E7am
 - La méthode des premières lettres : Cette méthode consiste à garder les premières lettres d'une phrase (citation, paroles de chanson, ...) en veillant à ne pas utiliser que des minuscules. Par exemple, la citation « un tiens vaut mieux que deux tu l'auras » deviendra 1TvmQ2tl'@

Problèmes liés aux mots de passe

- L'utilisation de mots de passe requiert leurs gestion (création, Modification, Déblocage, Suppression)
 - Problème : stockage, transfert sur le réseau, rejet (réutilisation frauduleuse), mémorisation difficile pour les utilisateurs, gestion pour l'administrateur
- La plupart des systèmes sont configurés de manière à bloquer temporairement le compte d'un utilisateur après un certain nombre de tentatives de connexion infructueuses.
 - Un pirate peut difficilement s'infiltrer sur un système de cette façon.
 - En contrepartie, un pirate peut se servir de ce mécanisme d'auto-défense pour bloquer l'ensemble des comptes utilisateurs afin de provoquer un déni de service.
- La plupart des mots de passe des systèmes sont stockés de manière chiffrée (cryptée) dans un fichier ou une base de données.
 - Lorsqu'un pirate obtient un accès au système et détient ce fichier, il lui est possible de tenter de casser le mot de passe de l'utilisateur.

Craquage du mot de passe

- Craquage par force brute
 - Il consiste à tester tous les mots de passe.
 - Plus il existe de combinaisons possibles pour former un mot de passe, plus le temps moyen nécessaire pour retrouver ce mot de passe sera long.
 - Rajouter un caractère à un mot de passe alphanumérique double le temps nécessaire pour le retrouver.
- Craquage par dictionnaire
 - L'attaque par dictionnaire consiste à tester une série de mots issus d'un dictionnaire. (dictionnaire des noms/marques, ...)
- Craquage hybride combinant force brute et dictionnaire
- Attaque par rejet : réutilisation du mot de passe.

Moyens de craquages des mots de passe

- Les moyens permettant au pirate d'obtenir les mots de passe des utilisateurs :
 - **Les key loggers** : Ce sont des logiciels qui sont installés sur le poste de l'utilisateur, permettent d'enregistrer les frappes de claviers saisies par l'utilisateur.
 - **L'ingénierie sociale**: consiste à exploiter la naïveté des individus pour obtenir des informations. Un pirate peut obtenir le mot de passe d'un individu en se faisant passer par un administrateur du réseau ou bien à l'inverse appeler l'équipe de support en demandant de réinitialiser le mot de passe en prétextant un caractère d'urgence.
 - **L'espionnage**: observation de l'utilisateur.

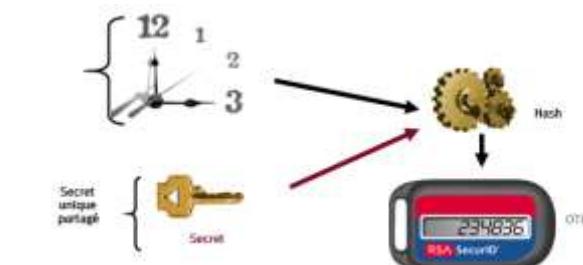
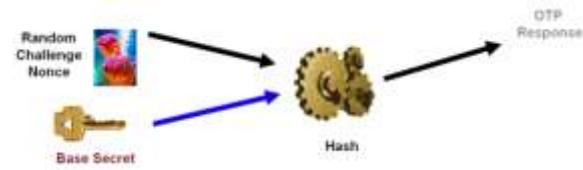
Mot de passe unique

- Concept de l'OTP (One Time Password) est que le mot de passe est utilisable une seule fois
 - Protection contre l'oubli du mot de passe
 - Protection contre le rejeu
 - Non prédictible si l'algorithme est gardé secret
- L'OTP peut être basé sur l'utilisation des jetons/authentifieurs/ *Token*



Techniques de l'OTP/authentif. par objet

- Asynchrone (challenge/réponse)
 - Envoi d'un challenge par le serveur, l'utilisateur possède une calculatrice qui transforme le challenge en un mot de passe qu'il saisit. Le serveur fait la même opération et compare.
- Synchrone dépendant du temps
 - Le mot de passe est fonction du temps et est généré à intervalle régulier. Le challenge (utilisé dans le mode asynchrone) est en fait la date et l'heure.
- Synchrone indépendant du temps
 - Utilisation d'un compteur interne à la calculatrice incrémenté à chaque utilisation. Le serveur est synchronisé sur ce compteur et n'accepte pas de code antérieur au compteur



Exemples d'OTP /authentif. par objet

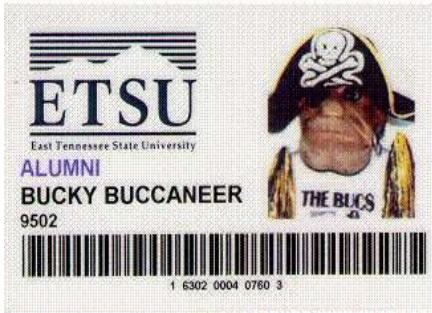
- La liste à biffer ou TAN (Transaction Authentication Number). Il s'agit de rentrer un OTP provenant d'une liste de codes fournie (banque).
- *Matrix card authentication* ou authentification à carte matricielle. Il s'agit de rentrer un OTP provenant d'une carte matricielle fournie.
- Utilisation des SMS: l'utilisateur reçoit un OTP directement sur son téléphone portable.

The diagram illustrates two examples of two-factor authentication:

- TAN List:** A screenshot of a login page titled "Welcome to Any Bank". It shows fields for "User Name" (AnyUser) and "Password" (represented by five asterisks). A "Submit" button is below the password field. A large blue arrow points downwards from this screen to the next screen.
- Matrix Card Authentication:** A screenshot of a challenge-response interface. It displays "Current IdentityGuard#:" followed by the number 12345678. Below it, a "Challenge" section shows a 3x3 grid of letters: A2, C4, E2. To the right, a "Response" section shows a 3x3 grid of numbers: 9, 2, [empty], 2, 4, [empty]. A red arrow points from the "E2" challenge cell to the first empty response cell. Another red arrow points from the "2" in the challenge grid to the second empty response cell. A third red arrow points from the "4" in the challenge grid to the third empty response cell. A "Submit" button is at the bottom.
- Matrix Card:** A screenshot of a 5x10 grid labeled "ANY BANK" with columns A through J and rows 1 through 5. The grid contains numerical values. Red arrows point from the highlighted challenge cells (A2, C4, E2) to the corresponding cells in the matrix grid. The matrix grid also has some cells circled in red.

Authentification par objets

- Cartes d'identité (sans mot de passe)
 - Code barre, Carte magnétique, puce, clé électronique
 - Problèmes : Copie, vol, liaison lecteur↔ Serveur, rejeu



- Cartes d'identité + mot de passe
 - Authentification double facteur
 - Problèmes :
 - Liaison lecteur ↔ Serveur, rejeu
 - En cas copie ou de vol le pirate doit « simplement » résoudre un problème de mot de passe

Saisie de temps à l'aide
d'une clé électronique RFID

L'authentification par caractère : la biométrie

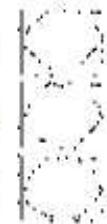
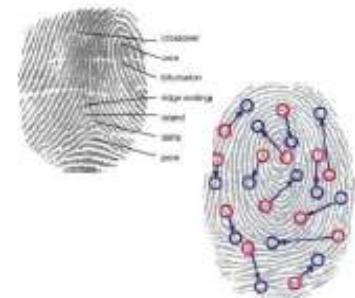
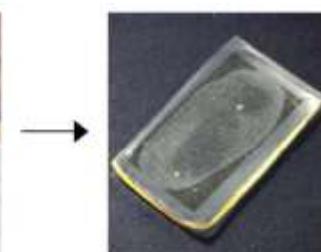
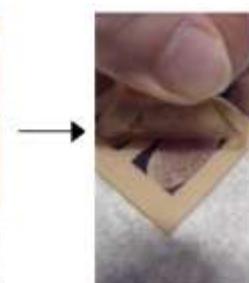
- La biométrie est la technologie qui mesure les caractéristiques physiques d'un individu, que ce soit ses empreintes digitales, la forme de son visage ou encore son ADN. Les attributs biométriques doivent posséder les caractéristiques suivantes :
 - **L'unicité**: chaque attribut biométrique doit varier énormément d'une personne à l'autre au point que l'ensemble des variations rend cet attribut unique.
 - **La robustesse**: un attribut biométrique devrait être permanent tout au long de la vie d'une personne.
 - **La quantifiabilité**: cet attribut doit être mesurable et quantifiable.
 - **L'acceptabilité par la population** (les empreintes digitales sont souvent associées à la criminalité).
 - **L'universalité** : quelle proportion de la population a cet attribut ?

La biométrie



Toutes caractéristiques humaines

- Reconnaissance Iris
- Empreintes digitales ou de la mains
- Reconnaissance des lèvres
- Cartographie des veines
- Reconnaissance de visages
- Reconnaissance d'oreille (et échos du conduit auditif)
- Spectre d'absorption de la peau
- Voie
- ADN

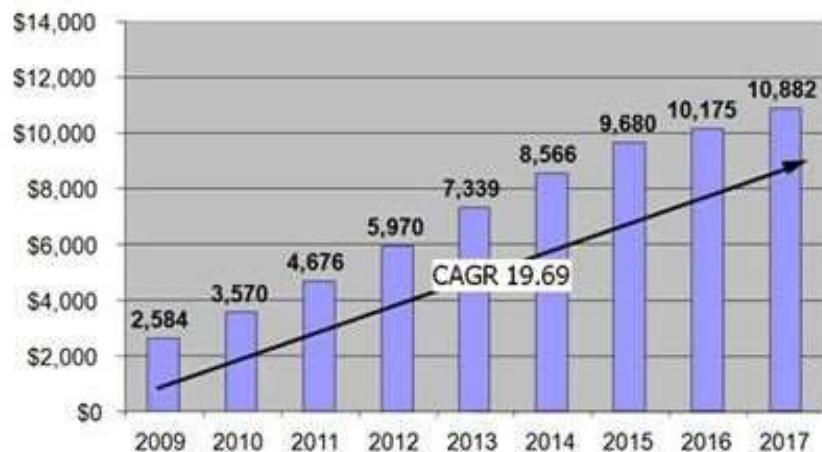


La biométrie

ACUITY
MARKET INTELLIGENCE

Global Market Growth

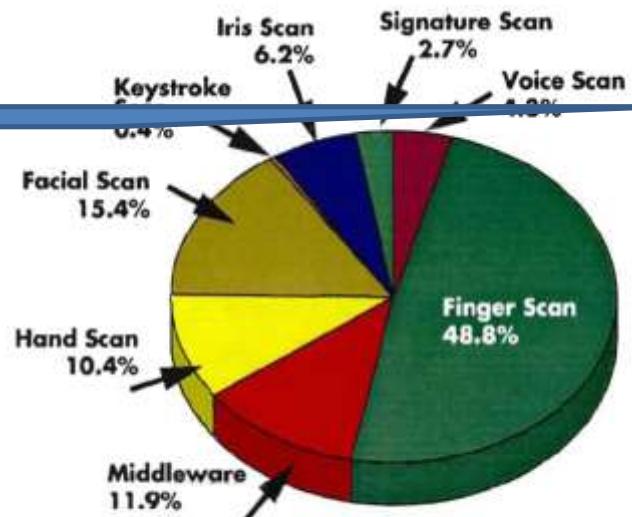
Biometrics industry Revenues 2009 – 2017
(USD \$M)



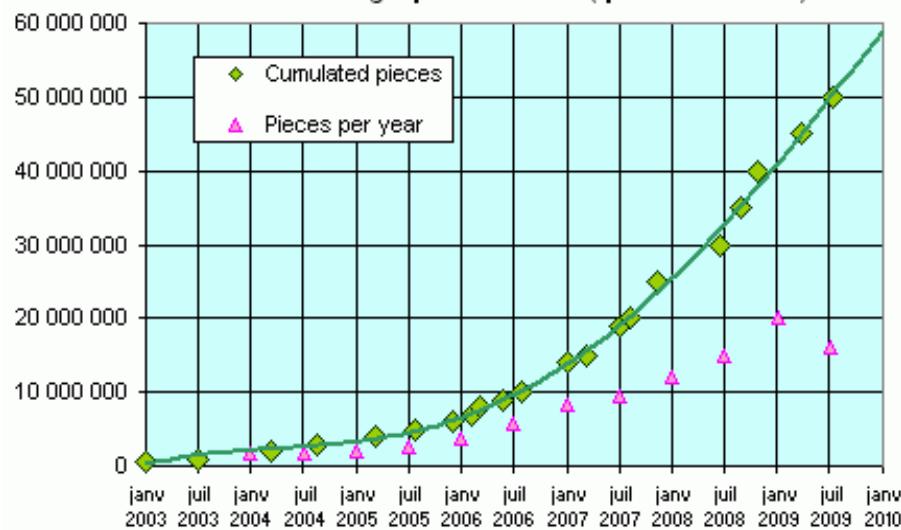
Graph 2.1

October 7, 2009

9



Authentec fingerprint sensors (press releases)



Sources: International Biometrie Group, New York, NY; 1.212.809.9491 2010.

Authentification par savoir faire

- Il s'agit d'une reconnaissance de ce que l'utilisateur sait faire. Par exemple authentification par signature ou par geste.



Outils de cryptographie et d'authentification

Authentification

Chiffrement symétrique

Chiffrement asymétrique

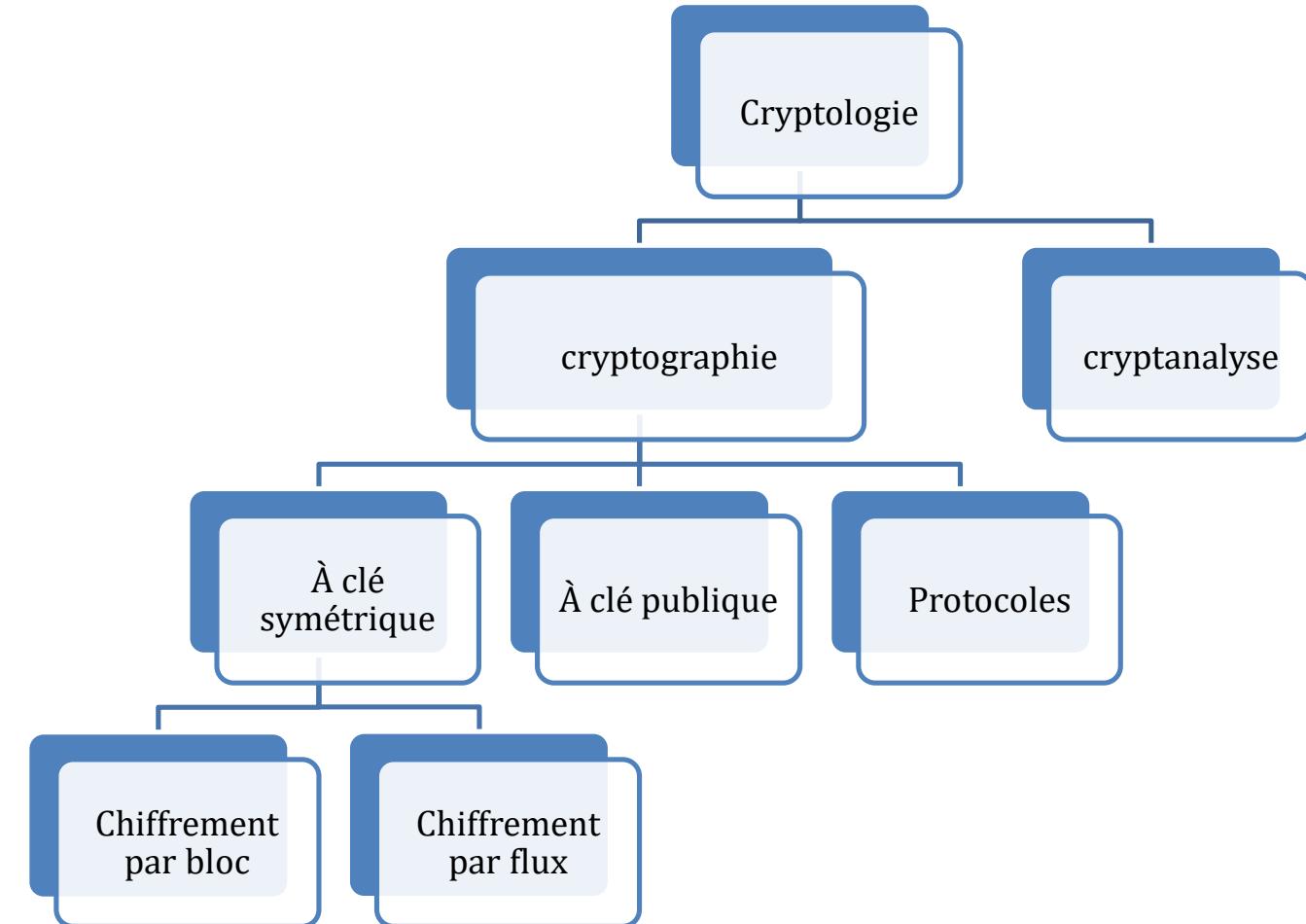
Intégrité d'un message

PKI

Terminologie

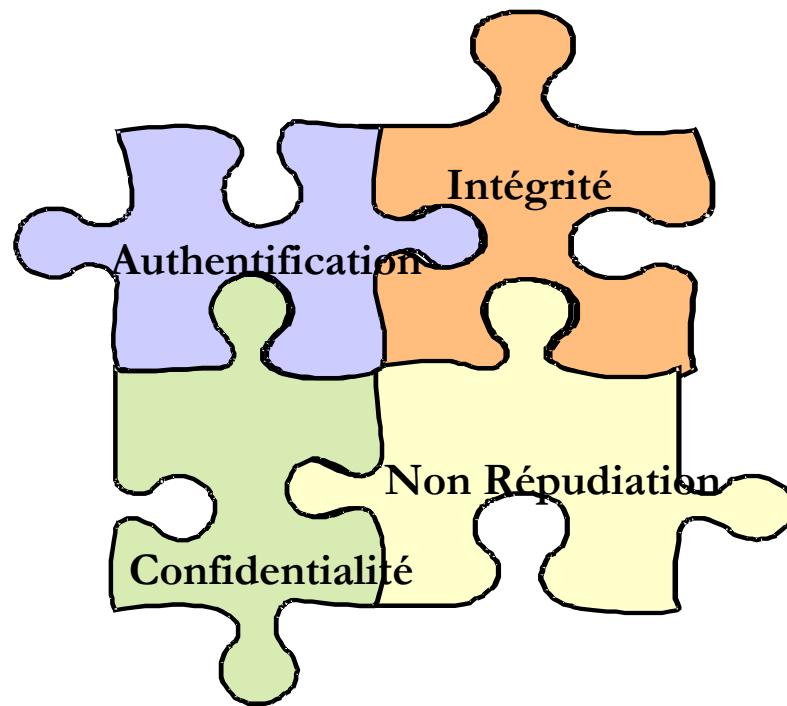
- **Chiffrement** : transformation à l'aide d'une clé de chiffrement d'un message en clair en un message incompréhensible si on ne dispose pas d'une clé de déchiffrement (en anglais encryption) ;
- **Chiffre** : anciennement code secret, par extension l'algorithme utilisé pour le chiffrement ;
- **Cryptogramme** : message chiffré ;
- **Décrypter** : retrouver le message clair correspondant à un message chiffré sans posséder la clé de déchiffrement (terme que ne possèdent pas les anglophones, qui eux « cassent » des codes secrets) ;
- **Cryptographie** : étymologiquement « écriture secrète », devenue par extension l'étude de cet art (donc aujourd'hui la science visant à créer des cryptogrammes, c'est-à-dire à chiffrer) ;
- **Cryptanalyse** : science analysant les cryptogrammes en vue de les décrypter ;
- **Cryptologie** : science regroupant la cryptographie et la cryptanalyse.

Classification



Définition de la cryptographie

- Science mathématique permettant d'effectuer des opérations sur un texte intelligible afin d'assurer une ou plusieurs propriétés de la sécurité de l'information.



Définition d'un crypto-système

- Un crypto-système est décrit par cinq uplets (P, C, K, E, D) :
 - « P » est un ensemble fini de textes clairs (Plain text)
 - « C » est un ensemble fini de textes cryptés (Cypher text)
 - « K » est l'espace de clés (*key space*); c'est un ensemble fini de clés possibles.
 - Pour chaque $k \in K$, il existe une fonction chiffrement $e_k \in E$, et une fonction de déchiffrement correspondante $d_k \in D$
 - Les fonctions $e_k : P \rightarrow C$ et $d_k : C \rightarrow P$ doivent satisfaire : $d_k(e_k(x)) = x$ pour chaque $x \in P$

Définition de la cryptanalyse

- Principes et méthodes permettant de trouver un message clair à partir d'un message crypté sans connaissance de la clé.
- Attaques classifiées selon le type de connaissance disponible pour l'intrus (cryptanalyst).
- Connaissant $C=E(P,K)$ mais pas K , l'objectif est de trouver P ou K .
- Types d'attaques de cryptanalyse:
 - Texte chiffré uniquement: uniquement C et E sont connus par l'intrus
 - Texte clair connu: Uniquement E , C , et quelques paires de messages clairs/cryptés avec K , sont connus par l'intrus
 - Texte clair choisi: E , C , sont connus, et P a été choisi par l'intrus.
 - ...

Les principes de Kerckhoffs

- Le linguiste hollandais Auguste Kerckhoffs Van Nieuwenhof dans son traité *La cryptographie militaire a énoncé les principes suivants qui sont les axiomes de la cryptographie*
 - Le système doit être matériellement, sinon mathématiquement, indéchiffrable;
 - Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi;
 - La clé doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants;
 - Il faut qu'il soit applicable à la correspondance télégraphique;
 - Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes;
 - Enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.

Un peu d'histoire

- Le premier « document » chiffré connu remonte à l'Antiquité. Il s'agit d'une tablette d'argile, retrouvée en Irak, et datant du XVI^e siècle av. J.-C. Un potier y avait gravé sa recette secrète en supprimant des consonnes et en modifiant l'orthographe des mots.
- **La technique grecque:** Entre le X^e et VII^e siècle av. J.-C. Technique de chiffrement par transposition, c'est-à-dire reposant sur le changement de position des lettres dans le message, en utilisant un bâton de diamètre déterminé appelée scytale. On enroulait en hélice une bande de cuir autour de la scytale avant d'y inscrire un message. Une fois déroulé, le message était envoyé au destinataire qui possédait un bâton identique, nécessaire au déchiffrement.



Un peu d'histoire

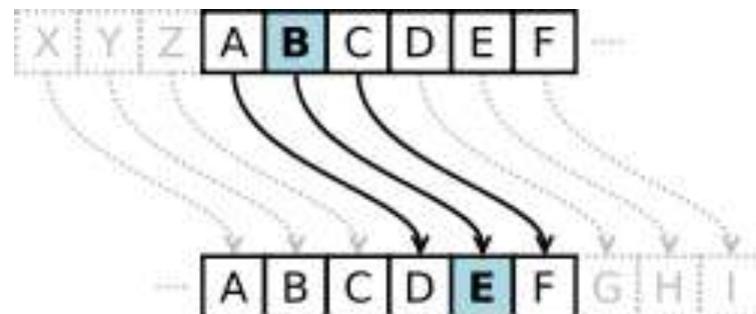
- **La technique des Hébreux:** À partir du V^e siècle av. J.-C., l'une des premières techniques de chiffrement est utilisée dans les textes religieux par les Hébreux qui connaissent plusieurs procédés.
Le plus connu appelé *Atbash* est une méthode de substitution alphabétique inversée. Elle consiste à remplacer chaque lettre du texte en clair par une autre lettre de l'alphabet choisie de la manière suivante : A devient Z, B devient Y, etc.
- **Nabuchodonosor:** Nabuchodonosor, roi de Babylone, employait une méthode originale : il écrivait sur le crâne rasé de ses esclaves, attendait que leurs cheveux aient repoussé, et il les envoyait à ses généraux. Il suffisait ensuite de raser à nouveau le messager pour lire le texte. Il s'agit toutefois de stéganographie (communication invisible) à proprement parler et non pas de cryptographie : l'information est cachée et non pas codée. On remarque dans ce procédé une certaine fiabilité : en effet l'interception du message par un tiers est tout de suite remarquée.

Un peu d'histoire

- **Les premiers « vrais » systèmes de cryptographie:** Il faut attendre -200 pour voir apparaître les premiers « vrais » systèmes de cryptographie. Ce sont essentiellement des chiffrements par substitution.
- Il existe différents types de substitutions :
 - *mono-alphabétique* : remplace chaque lettre du message par une autre lettre de l'alphabet
 - *poly-alphabétique* : utilise une suite de chiffres mono-alphabétiques (la clé) réutilisée périodiquement
 - *polygrammes* : substitue un groupe de caractères dans le message par un autre groupe de caractères

Un peu d'histoire

- **Le code de César:** Le code de César est la méthode cryptographique, par substitution mono-alphabétique, la plus ancienne (I^{er} siècle av. J.-C.).
- Cette méthode est utilisée dans l'armée romaine et bien qu'elle soit beaucoup moins robuste que la technique Atbash, la faible alphabétisation de la population la rend suffisamment efficace.
- Méthode de chiffrement: Son système est simple, il consiste à décaler les lettres de l'alphabet d'un nombre n . Par exemple, si on remplace A par D ($n=3$), on remplace B par E, C par F...
- Le texte que nous souhaitons coder étant le suivant : « décaler les lettres de l'alphabet »
- Le texte codé est alors : « ghfdohu ohv ohwwuhv gh o'doskdehw »



Un peu d'histoire

- **Le code de César:**
- Malheureusement, on comprendra que ce système est très peu sûr, puisqu'il n'y a que 26 lettres dans l'alphabet donc seulement 25 façons de chiffrer un message avec le code de César(on ne peut substituer une lettre par elle-même). Pourtant sa simplicité conduit les officiers sudistes à le réemployer durant la guerre de Sécession. L'armée russe en fit de même en 1915.
- Un système connu et pourtant Le code de César a été utilisé sur des forums internet sous le nom de ROT13 (rotation de 13 lettres ou A→N...). Le ROT13 n'a pas pour but de rendre du texte confidentiel, mais plutôt d'empêcher la lecture involontaire (d'une réponse à une devinette, ou de l'intrigue d'un film, etc.). Son utilisation est simple : il suffit de re-chiffrer un texte, codé en ROT13, une deuxième fois pour obtenir le texte en clair.

Un peu d'histoire

- Le **chiffre de Vigenère** est un système de chiffrement, élaboré par Blaise de Vigenère (1523-1596), diplomate français du XVI^e siècle.
- C'est un système de substitution poly-alphabétique ou de chiffrement poly-alphabétique. Cela signifie qu'il permet de remplacer une lettre par une autre qui n'est pas toujours la même, contrairement au code de César ou à ROT13 qui se contentaient d'utiliser la même lettre de substitution. C'est donc un système relativement plus « solide » que ces deux systèmes.
- **Principe:** Ce chiffrement introduit la notion de clé. Une clé se présente généralement sous la forme d'un mot ou d'une phrase. Pour pouvoir chiffrer notre texte, à chaque caractère nous utilisons une lettre de la clé pour effectuer la substitution. Évidemment, plus la clé sera longue et variée et mieux le texte sera chiffré. Il faut savoir qu'il y a eu une période où des passages entiers d'œuvres littéraires étaient utilisés pour chiffrer les plus grands secrets. Les deux correspondants n'avaient plus qu'à avoir en leurs mains un exemplaire du même livre pour s'assurer de la bonne compréhension des messages.

Un peu d'histoire

- Table de Vigenère:

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

Un peu d'histoire

- Le **chiffre de Vigenère**: Pour chaque lettre en clair, on sélectionne la colonne correspondante et pour une lettre de la clé on sélectionne la ligne adéquate, puis au croisement de la ligne et de la colonne on trouve la lettre chiffrée. La lettre de la clé est à prendre dans l'ordre dans laquelle elle se présente et on répète la clé en boucle autant que nécessaire.
 - clé : MUSIQUE
 - texte : j'adore écouter la radio toute la journée
 - j'adore écouter la radio toute la journée
 - M USIQU EMUSIQU EM USIQU EMUSI QU EMUSIQU
 - | | | |
 - | | | Colonne O, ligne I : on obtient la lettre W.
 - | | | Colonne D, ligne S : on obtient la lettre V.
 - | | | Colonne A, ligne U : on obtient la lettre U.
 - Colonne J, ligne M : on obtient la lettre V.

Un peu d'histoire

- Le texte chiffré est alors :
V'UVWHY IOIMBUL PM LSLYI XAOLM BU NAOJVUY.
- 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.
 - V'UVWHY IOIMBUL PM LSLYI XAOLM BU NAOJVUY
 - M USIQU EMUSIQU EM USIQU EMUSI QU EMUSIQU
 - | |||
 - | || Ligne I, on cherche W: on trouve la colonne O.
 - | | Ligne S, on cherche V: on trouve la colonne D.
 - | Ligne U, on cherche U: on trouve la colonne A.
 - Ligne M, on cherche V: on trouve la colonne J.

Exercice d'application

En utilisant le chiffre de Vigenère:

- a) Chiffrez le message suivant en utilisant le mot clé "UNIVERSITE":

"CET EXERCICE EST FACILE"

- b) Déchiffrez le message suivant, en utilisant le mot clé "ETUDIANT" :

" FKUYW CRLX MLHA BVXR "

Exercice d'application

- Lester Hill, mathématicien cryptographe (1891-1961) propose en 1929 un nouveau type d'algorithme de chiffrement. Son idée n'est plus de coder lettres par lettres, mais de coder simultanément des groupes de ***m*** lettres! Bien sûr, plus ***m*** est grand, plus les analyses statistiques deviennent difficiles!
- D'abord, nous remplaçons chaque lettre par son ordre dans l'alphabet -1 : A devient 0, B devient 1,..., Z devient 25. On groupe les nombres ainsi obtenus par ***m*** (prenons par exemple m=2).
- Pour chaque bloc de ***m*** nombres à coder $x_1x_2\dots x_m$, on calcule le texte codé en effectuant des combinaisons linéaires (ici $m=2$) :

$$y_1=ax_1+bx_2$$

$$y_2=cx_1+dx_2$$

Exercice d'application

- Si a, b, c, d sont des entiers, y_1 et y_2 seront aussi des entiers. Pourtant, si l'on souhaite les reconvertir en lettres, il faudrait qu'ils soient compris entre 0 et 25 ce dont on ne peut s'assurer. On les y ramène en prenant leur reste dans la division par 26. Si z_1 et z_2 sont les restes respectifs de y_1 et y_2 dans la division par 26, on peut retransformer z_1 et z_2 en lettres, et obtenir le message codé.
- a) Codez le mot **chiffrer** avec le chiffre de Hill, pour $m=2$, $a=3$, $b=5$, $c=1$ et $d=2$.
- Diviser le texte à coder en blocs de m lettres
 - Convertir les lettres en chiffres
 - Calculer les y_i
 - Calculer les z_i
 - Déduire le mot codé
- b) Est-ce qu'une analyse de fréquence des lettres permet de casser ce code? Expliquez.

Un peu d'histoire

- **Enigma** est une machine électromécanique portable d'origine allemande, faisant appel à des rotors montés sur cylindres pour le chiffrement et le déchiffrement de l'information. Plus précisément, Enigma est une famille de machines, car il en a existé de nombreuses et subtiles variantes. Enigma fut commercialisée en Europe et dans le reste du monde dès le début des années 1920. Elle fut aussi adaptée pour une utilisation par les services militaires et diplomatiques de nombreuses nations.
- Son utilisation la plus fameuse fut celle de l'Allemagne nazie et de ses alliés, avant et pendant la Seconde Guerre mondiale.



Un peu d'histoire

- Bien qu'elle fût considérée avant la Seconde Guerre mondiale comme sûre par ses utilisateurs, les cryptologues britanniques furent, à plusieurs reprises et sur de longues durées, capables de décrypter les messages protégés par ces machines. Les informations obtenues grâce à cette source leur donnèrent un net avantage dans la poursuite de la guerre.
- Enigma chiffre les informations en réalisant le passage d'un courant électrique à travers une série de composants. Le courant est transmis en pressant une lettre sur le clavier. Après sa traversée dans un réseau complexe de fils, une lampe indique la lettre chiffrée.

Critères de sécurité



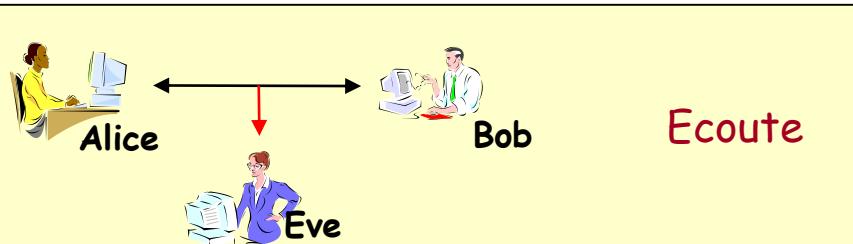
- **Confidentialité (confidentiality)**: garantie que seules les personnes autorisées ont accès aux éléments considérés.
- **Intégrité des données (data integrity)**: garantie que les éléments considérés sont exacts et complets.
- **Authentification (authentication)**: possibilité de vérifier l'identité d'une entité (personne, ordinateur...), afin d'autoriser l'accès de cette entité à des ressources (systèmes, réseaux, applications...).
- **Non-répudiation (non-repudiation)**: la possibilité de vérifier que l'envoyeur et le destinataire sont bien les parties qui disent avoir respectivement envoyé ou reçu le message.

Critères de sécurité

- Autres aspects importants
 - **Disponibilité (availability)**: garantie que ces éléments considérés sont accessibles au moment voulu par les personnes autorisées.
 - **Anonymat (privacy)**: garantie que l'identité et/ou la localisation de l'entité reste(nt) confidentielle(s).
- Un outil fondamental pour assurer la sécurité est la **cryptographie**
- La cryptographie et la cryptanalyse sont des outils importants pour assurer la confidentialité d'une information (stockée ou transmise), son intégrité (toute modification est détectable), et l'identification de son origine (l'émetteur peut être identifié).

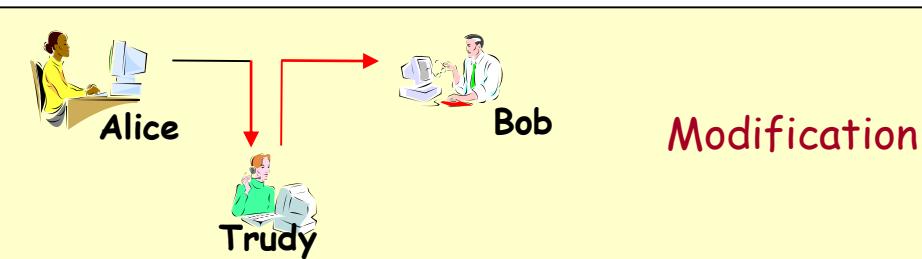
Types d'attaques

Passive

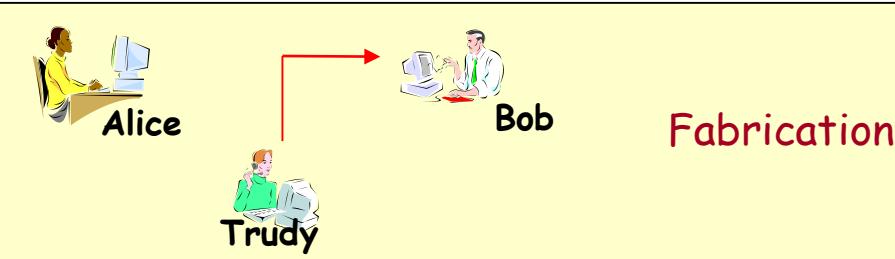


Ecoute

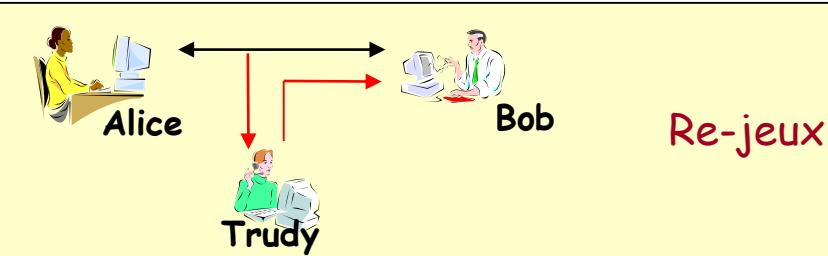
Active



Modification

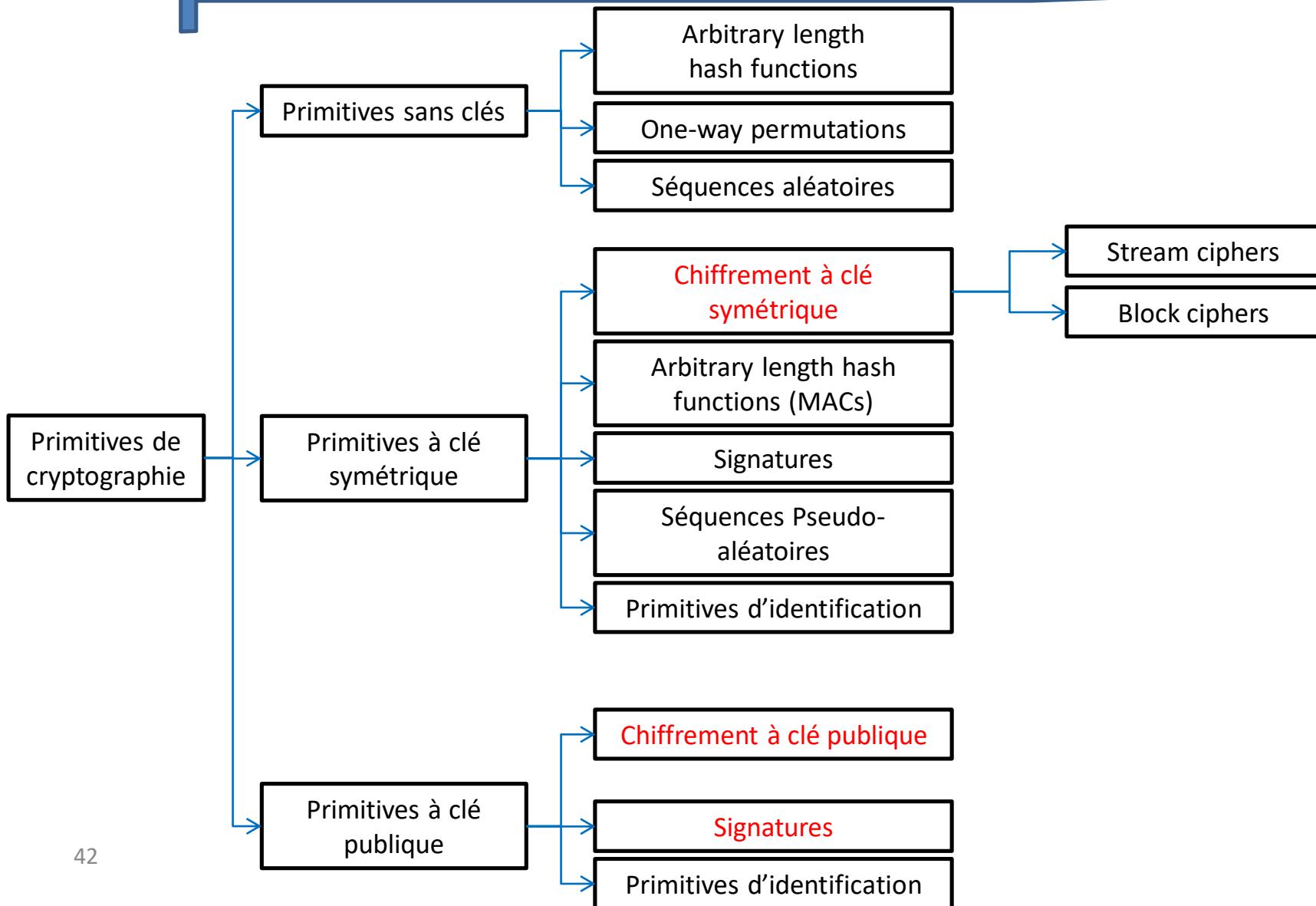


Fabrication

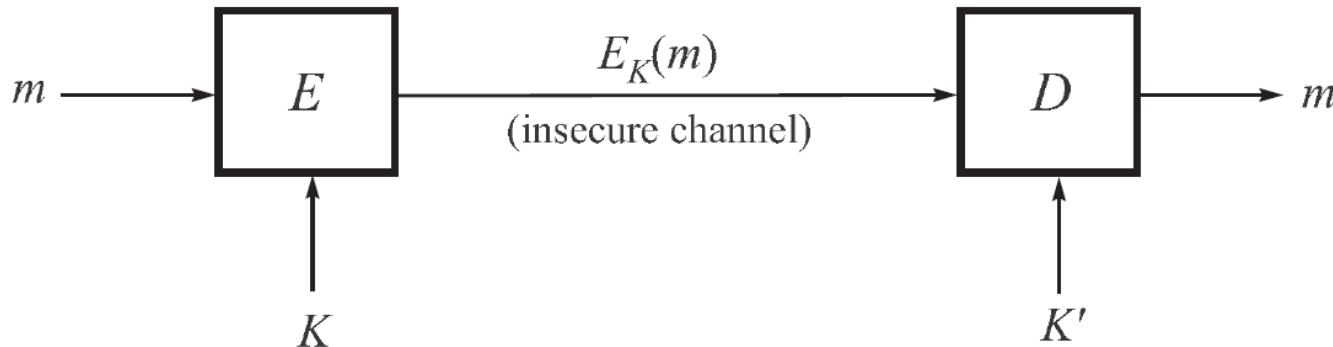


Re-jeux

Primitives Cryptographiques

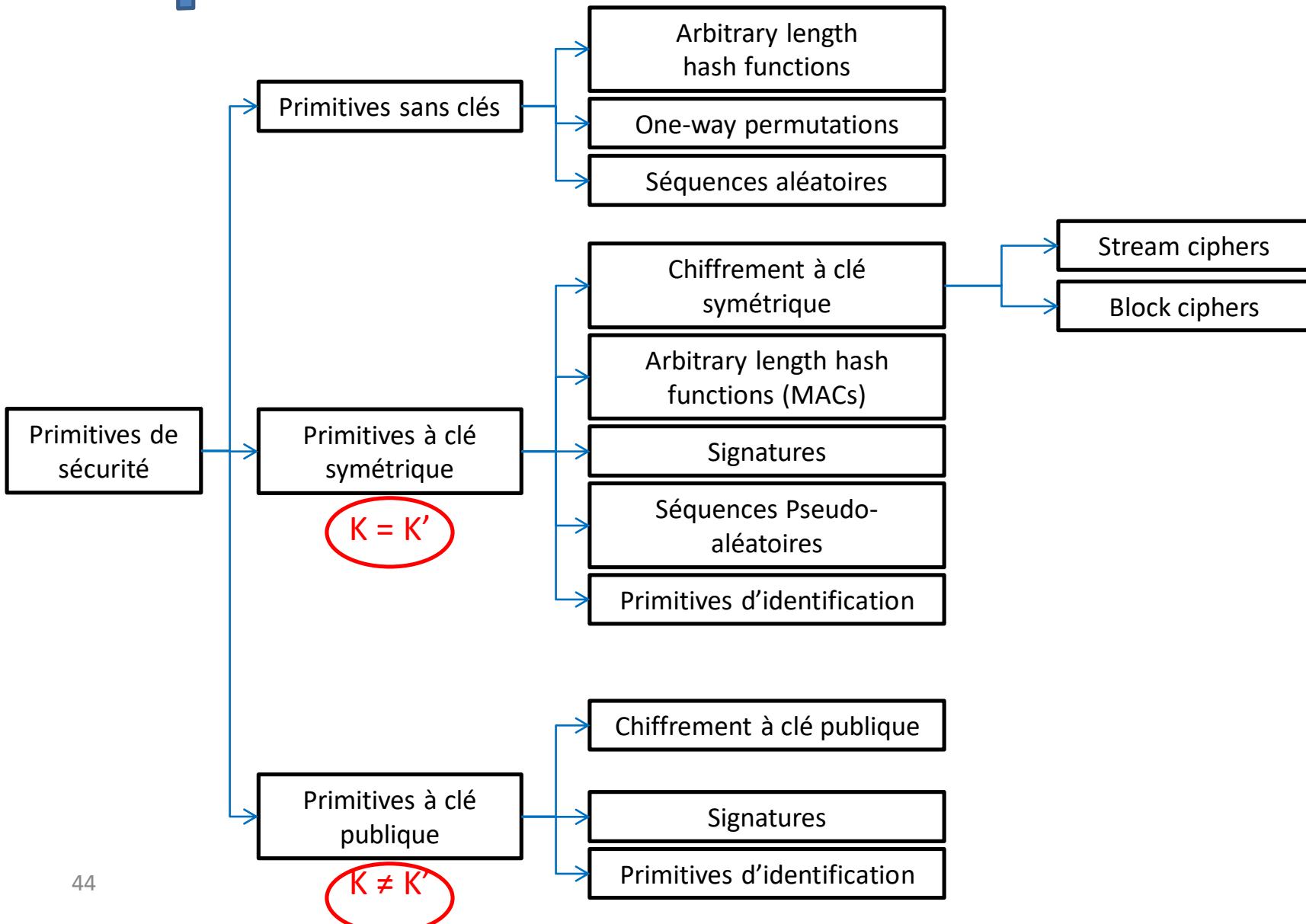


Chiffrement



- Le nœud **S** envoie le message **m** au noeud **R** via un canal non sécurisé.
- **S** chiffre **m** en utilisant un algorithme de chiffrement **E** et une clé de chiffrement **K**.
 - $E_K(m)$ est le texte chiffré (ciphertext)
 - **m** est le texte en clair (plaintext)
- **R** décrypte le texte chiffré avec un algorithme de déchiffrement **D** et d'une clé de déchiffrement **K'**

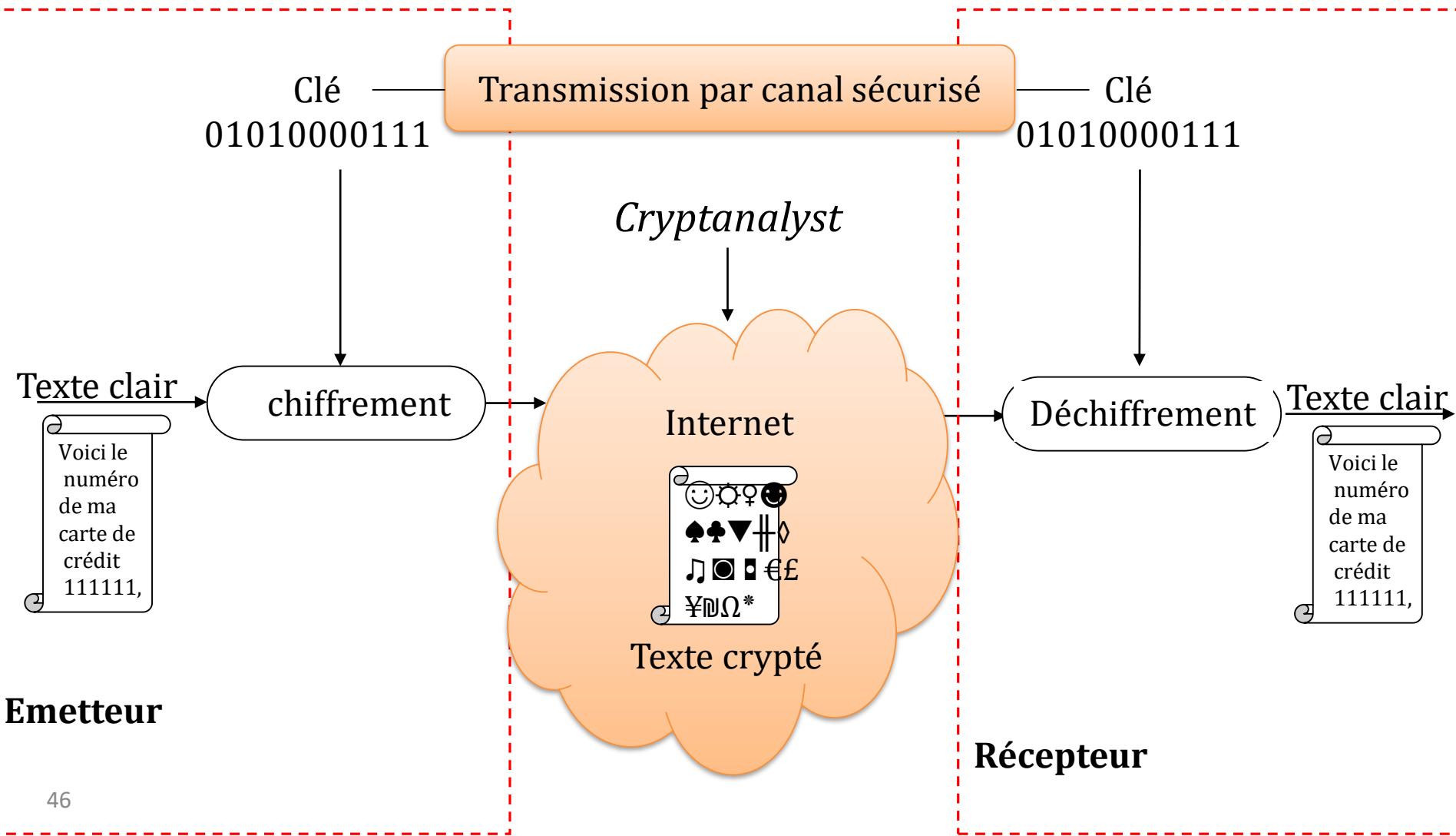
Primitives Cryptographiques



Principe du chiffrement symétrique

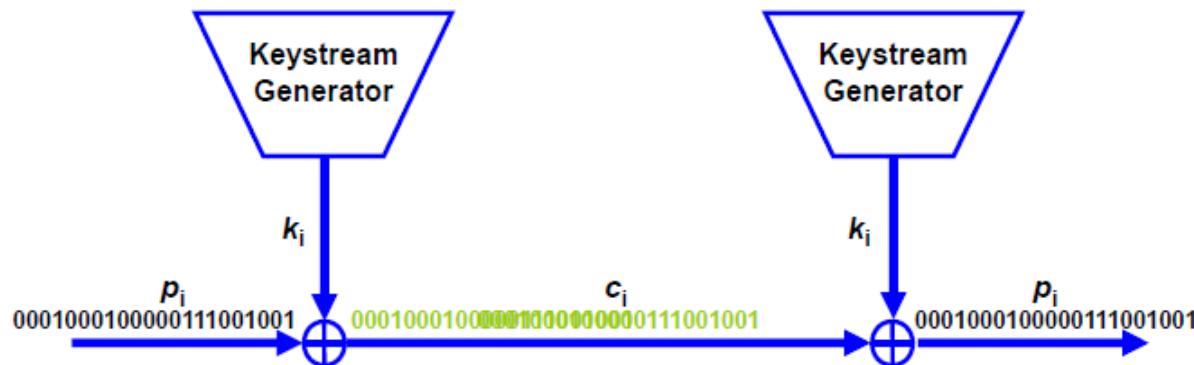
- Aussi nommé chiffrement à clé secrète, le chiffrement symétrique utilise la **même clé**, tenue secrète, pour le **chiffrement et le déchiffrement**.
 - Le chiffrement s'effectue en additionnant (fonction XOR) la clé au message coupé en blocs et en effectuant plusieurs permutations, ceci fait plusieurs fois.
- Le chiffrement symétrique utilise des canaux sécurisés pour échanger la clé secrète (IPSec) (par exemple : un algorithme de chiffrement à clé publique comme RSA)
- Algorithmes
 - DES (Data Encryption Standard), Triple-DES, AES (Advanced Encryption Standard),
 - IDEA (Internationale Data Encryption Algorithm), Blowfish
- ⇒ Simples à mettre en œuvre, l'exécution peut être très rapide

Principe du chiffrement symétrique



Principe du chiffrement symétrique

- Chiffrement par flux (Stream Cipher)
 - Principe: Traite les éléments d'entrée de façon continue, produisant à la fois un élément de sortie (crypté).
 - La clé est aussi longue que le stream de données.
 - Mode adapté pour la communication en temps réel: Pas besoin d'attendre l'arrivée du block entier
 - Implémenté en général sur des supports hardware



Stream Ciphers

- Le texte en clair est combiné avec un *keystream* (un flot de clés de chiffrement pseudo-aléatoire), généralement en utilisant l'opération exclusive-or (xor).
- 1 time pad:

Plaintext	1	1	0	1	0	0	0	1	1	0	1	0	1	0	0	1
Keystream	0	1	1	1	1	1	0	0	0	0	1	1	0	1	1	1
Ciphertext	1	0	1	0	1	0	0	1	1	1	0	0	0	1	1	1

(a) Encryption

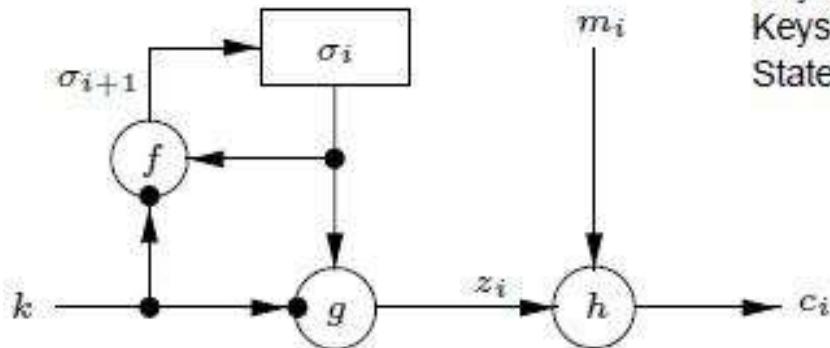
Ciphertext	1	0	1	0	1	0	0	1	1	1	0	0	0	1	1	1
Keystream	0	1	1	1	1	1	0	0	0	0	1	1	0	1	1	1
Plaintext	1	1	0	1	0	0	0	1	1	0	1	0	1	0	0	1

(b) Decryption using an identical keystream

Chiffrement par flux

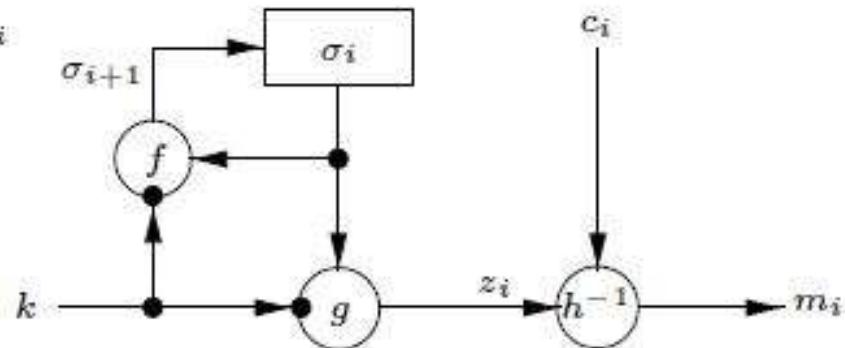
- Un *synchronous stream cipher* est un schéma de chiffrement dans lequel le keystream est généré indépendamment du texte en clair (plaintext) et du texte chiffré (ciphertext).
 - σ_0 : état initial (initial state)
 - f : La fonction Next-state
 - G : La fonction qui produit le keystream z_i

(i) Encryption



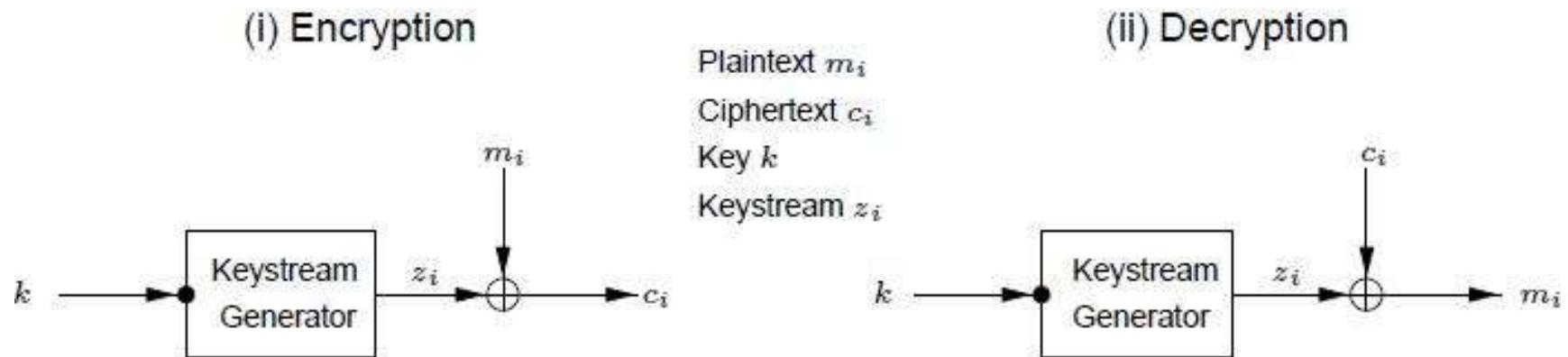
Plaintext m_i
Ciphertext c_i
Key k
Keystream z_i
State σ_i

(ii) Decryption



Chiffrement par flux

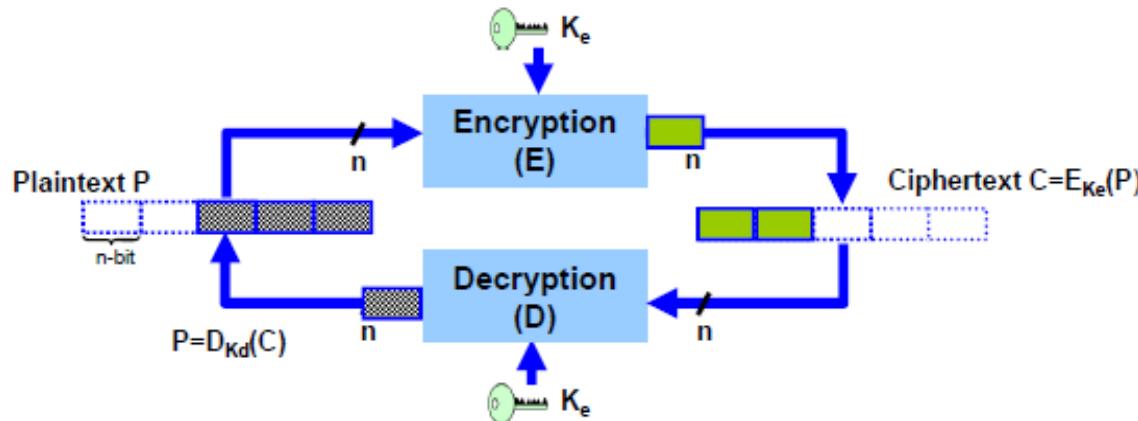
- La plupart des schémas de chiffrement par flux proposés dans la littérature sont *additifs*
- Définition: Un *binary additive stream cipher* est un synchronous stream cipher dans lequel le keystream, le texte en clair et le texte chiffré sont des données binaires, et le résultat de la fonction h est une fonction XOR.



Principe du chiffrement symétrique

- Chiffrement par bloc

- Principe: Le texte est divisé en différents blocks de taille fixe. Un block est traité à la fois, produisant un block de données cryptées.
- le block doit être entièrement disponible avant le traitement
- La même fonction et la même clé sont utilisées pour crypter les blocks successifs.
- Implémentation d'une manière logicielle en générale.



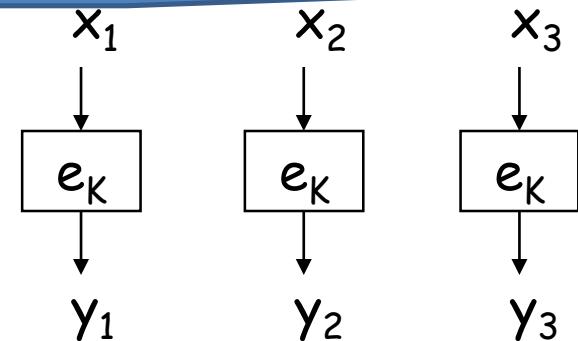
Chiffrement par bloc

- Le **chiffrement par bloc** (en anglais *block cipher*) est une méthode de chiffrements symétrique qui découpe le texte à chiffrer en blocs de taille généralement fixe.
- La taille de bloc est comprise entre 32 et 512 bits
 - Dans le milieu des années 1990 le standard était de 64 bits
 - Depuis 2000 le standard est de 128 bits
- Les blocs sont ensuite chiffrés les uns après les autres.
- Il est possible de transformer un chiffrement de bloc en un chiffrement par flux en utilisant un mode d'opération
 - ECB (chaque bloc chiffré indépendamment des autres)
 - CFB (on chaîne le chiffrement en effectuant un XOR entre les résultats successifs).

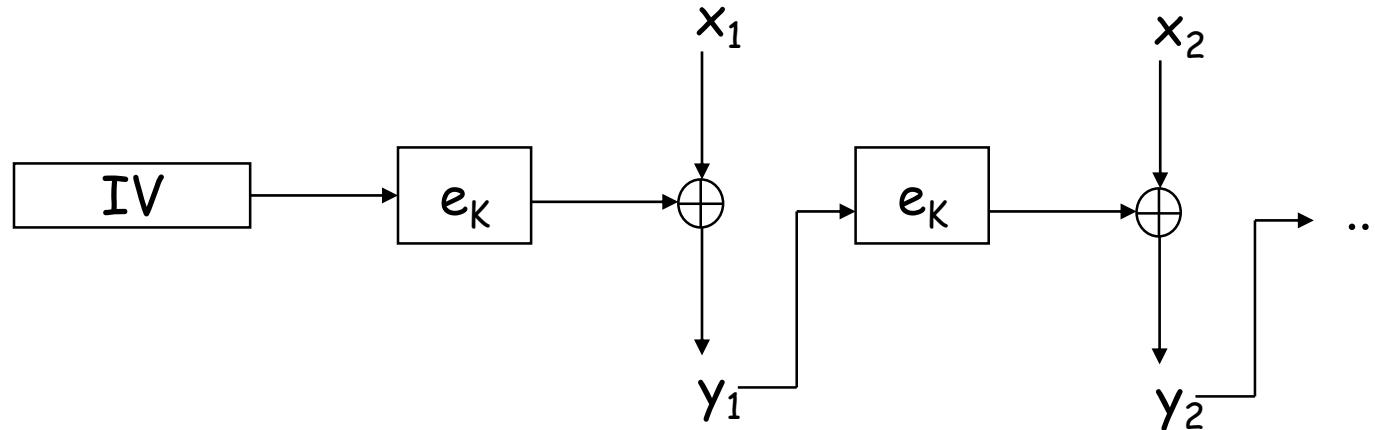
Chiffrement par bloc

Electronic Code Book mode (ECB):

- + : implementation simple
- : un attaquant peut aisément remplacer un block



Cipher FeedBack mode (CFB):

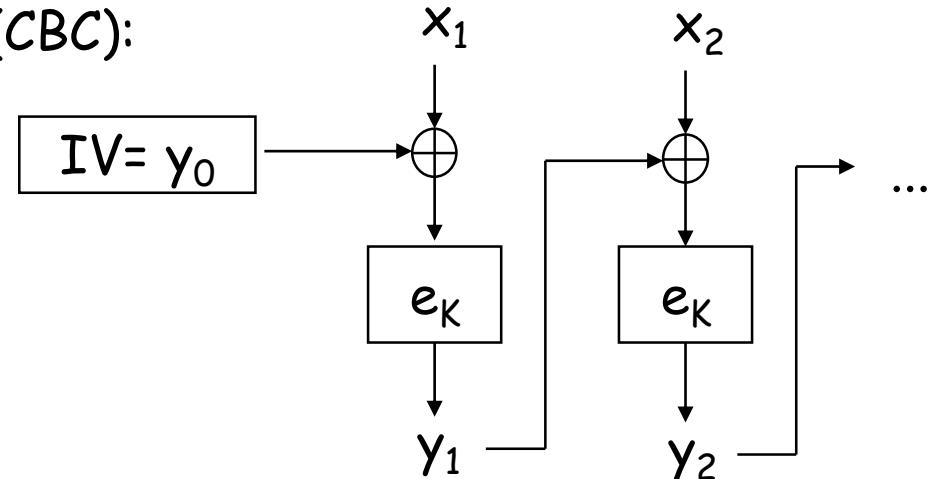


IV: Initial Vector

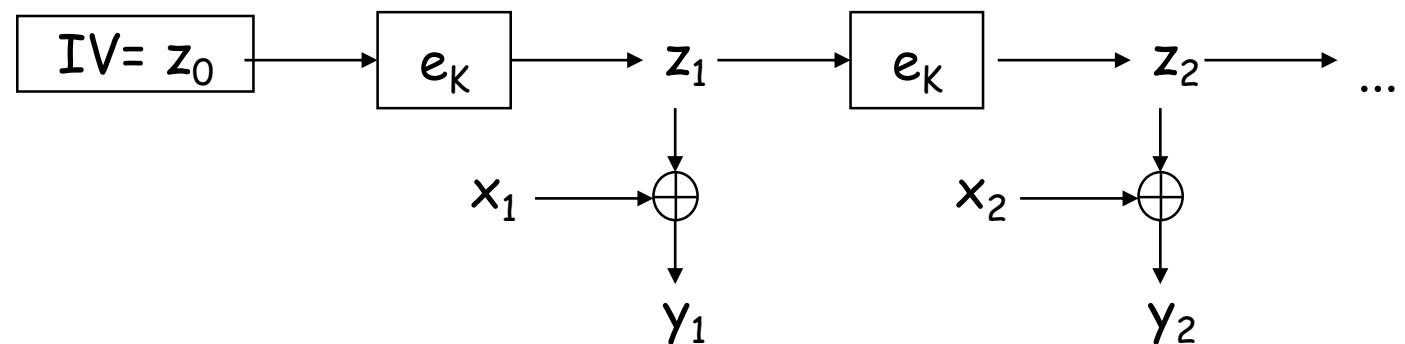
- But: chaque message codé devient unique
- Génération: timestamp ou nombre aléatoire (random number)

Chiffrement par bloc

Cipher Block Chaining mode (CBC):



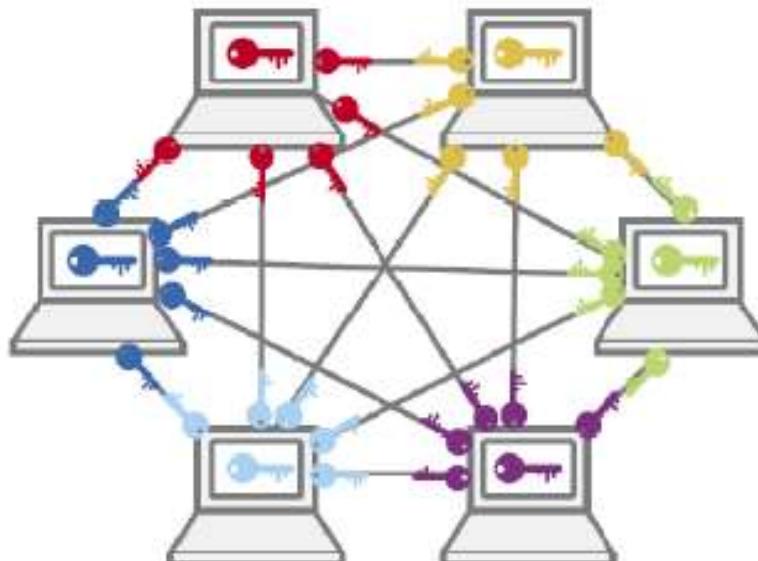
Output FeedBack mode (OFB):



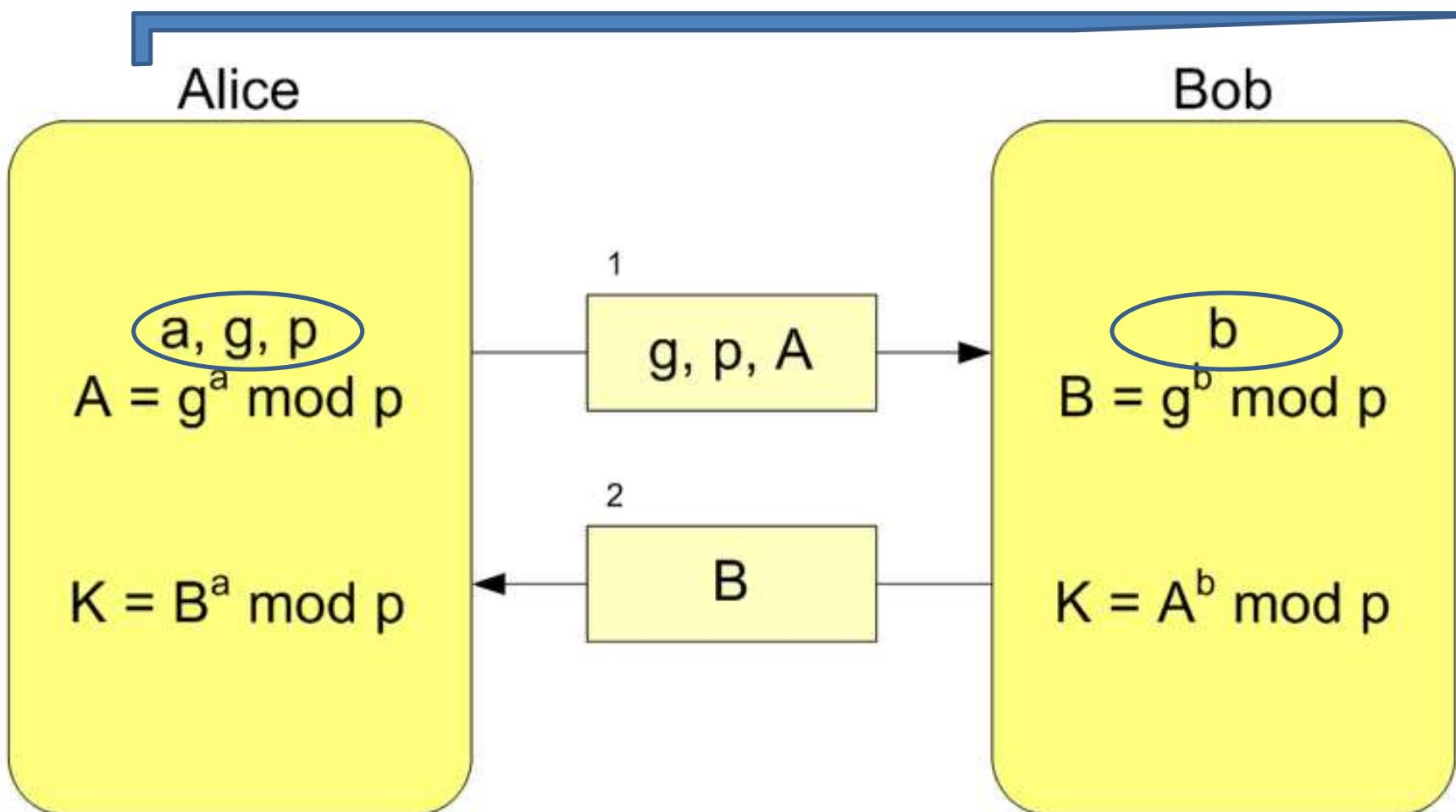
Problème de distribution de la clé

- Pour N utilisateurs, il faut $\frac{N(N-1)}{2}$ clés. Soit alors

Utilisateurs	Clés
100	5,000
1000	500,000

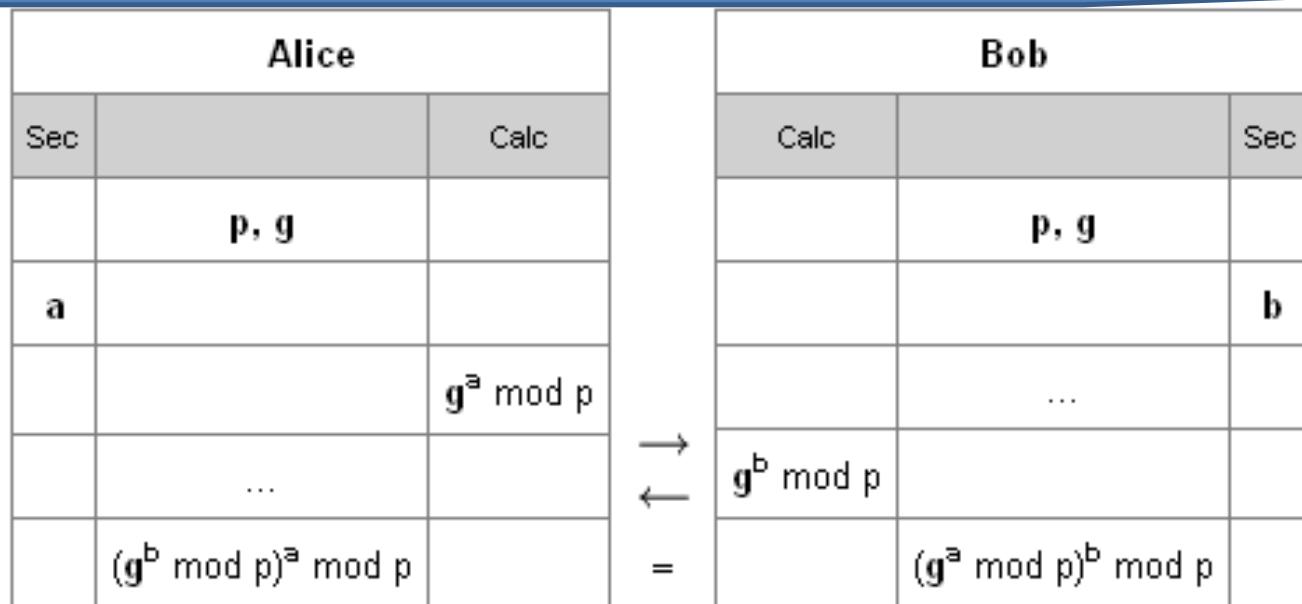


Diffie-Hellman: Principe



$$K = A^b \text{ mod } p = (g^a \text{ mod } p)^b \text{ mod } p = g^{ab} \text{ mod } p = (g^b \text{ mod } p)^a \text{ mod } p = B^a \text{ mod } p$$

Diffie-Hellman: Exemple



- Alice et Bob choisissent un nombre premier $p=23$ et une base $g=3$
- Alice choisit un nombre secret $a=6$
- Elle envoie à Bob la valeur $g^a \text{ [mod } p] = 3^6 \text{ [23]} = 16$
- Bob choisit à son tour un nombre secret $b=15$
- Bob envoie à Alice la valeur $g^b \text{ [mod } p] = 3^{15} \text{ [23]} = 12$
- Alice calcule la clé secrète : $(g^b \text{ [mod } p])^a \text{ [mod } p] = 12^6 \text{ [23]} = 9$
- Bob obtient la même clé qu'Alice : $(g^a \text{ [mod } p])^b \text{ [mod } p] = 16^{15} \text{ [23]} = 9$

Histoire du DES

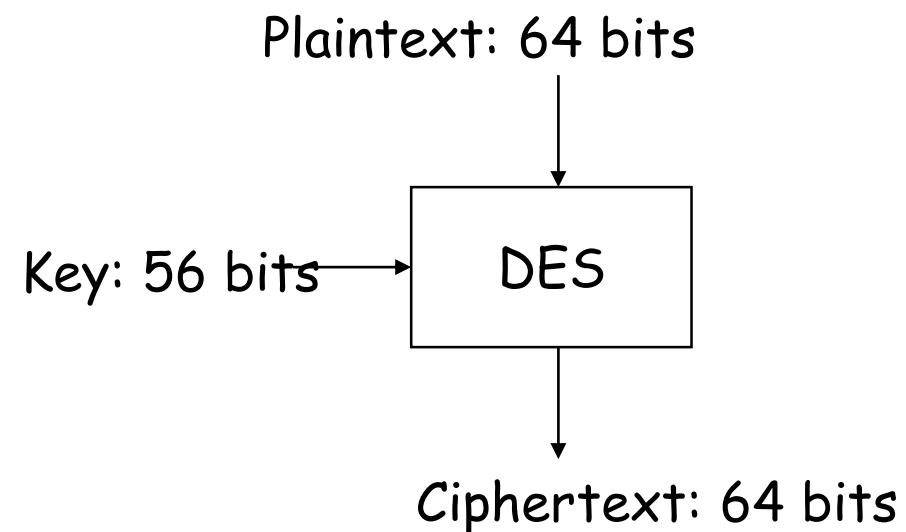
- Jusque dans les années 1970, seuls les militaires possédaient des algorithmes à clé secrète fiables.
- Devant l'émergence de besoins civils, le NBS (*National Bureau of Standards*) lança le 15 mai 1973 un appel d'offres dans le Federal Register (l'équivalent du Journal Officiel américain) pour la création d'un système cryptographique dont :
 - 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, en logiciel et en 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 .
- Les efforts conjoints d'IBM, qui propose Lucifer fin 1974, et de la NSA (*National Security Agency*) conduisent à l'*élaboration du DES (Data Encryption Standard) en 1975*, l'*algorithme de chiffrement le plus utilisé au monde* durant le dernier quart du XXième siècle.

Clé du DES

- La clé du DES est une chaîne de 64 bits (succession de 0 et de 1), dont seuls **56** bits servent réellement à définir la clé.
 - Les bits 8,16,24,32,40,48,56,64 sont des bits de parité (=bits de détection d'erreur)
 - Le 8ème bit est fait en sorte que sur les 8 premiers bits, il y ait un nombre impair de 1. Par exemple, si les 7 premiers bits sont 1010001, le 8ème bit est 0. Ceci permet d'éviter les erreurs de transmission
- Il y a donc pour le DES 2^{56} clés possibles, soit environ ... 72 millions de milliards de possibilités

Data Encryption Standard (DES)

- C'est *LE* schéma de chiffrement symétrique par bloc
- Développé dans les années 70 par IBM, standard ANSI en 1981 (ANSI X3.92)
- Largement utilisé dans les transactions bancaires
- N'est plus considéré comme suffisamment robuste
- Crypte des blocks de 64 bits en utilisant des clés relativement courtes (taille effective 56-bit).
- Implémentation facile en matériel.
 - Boites transposition P-Box
 - Boites de substitution S-Box
- Principe:



Data Encryption Standard (DES)

- Les étapes de cette élaboration sont restées secrètes, (la conception des S-Boxes).
- Les S-Boxes sont des tables qui définissent des permutations.
- Le message est découpé en blocs de 64 bits.
- Initialisation : permutation de tous les bits formant ce bloc.
- On le coupe en deux parties : L_0 et R_0 .

Principe du DES

- Les grandes lignes de l'algorithme sont :

- **Phase 1 : Préparation - Diversification de la clé**

Le texte est découpé en blocs de 64 bits. On diversifie aussi la clé K, c'est-à-dire qu'on fabrique à partir de K, 16 sous-clés K₁,...,K₁₆ à 48 bits. Les K_i sont composés de 48 bits de K, pris dans un certain ordre

- **Phase 2 : Permutation initiale**

Pour chaque bloc de 64 bits x du texte, on calcule une permutation finie $y=P(x)$. y est représenté sous la forme $y=L_0R_0$, L₀ étant les 32 bits à gauche de y, R₀ les 32 bits à droite.

- **Phase 3 : Itération**

On applique 16 itération d'une même fonction. A partir de L_{i-1}R_{i-1} (pour i de 1 à 16), on calcule L_iR_i en posant :

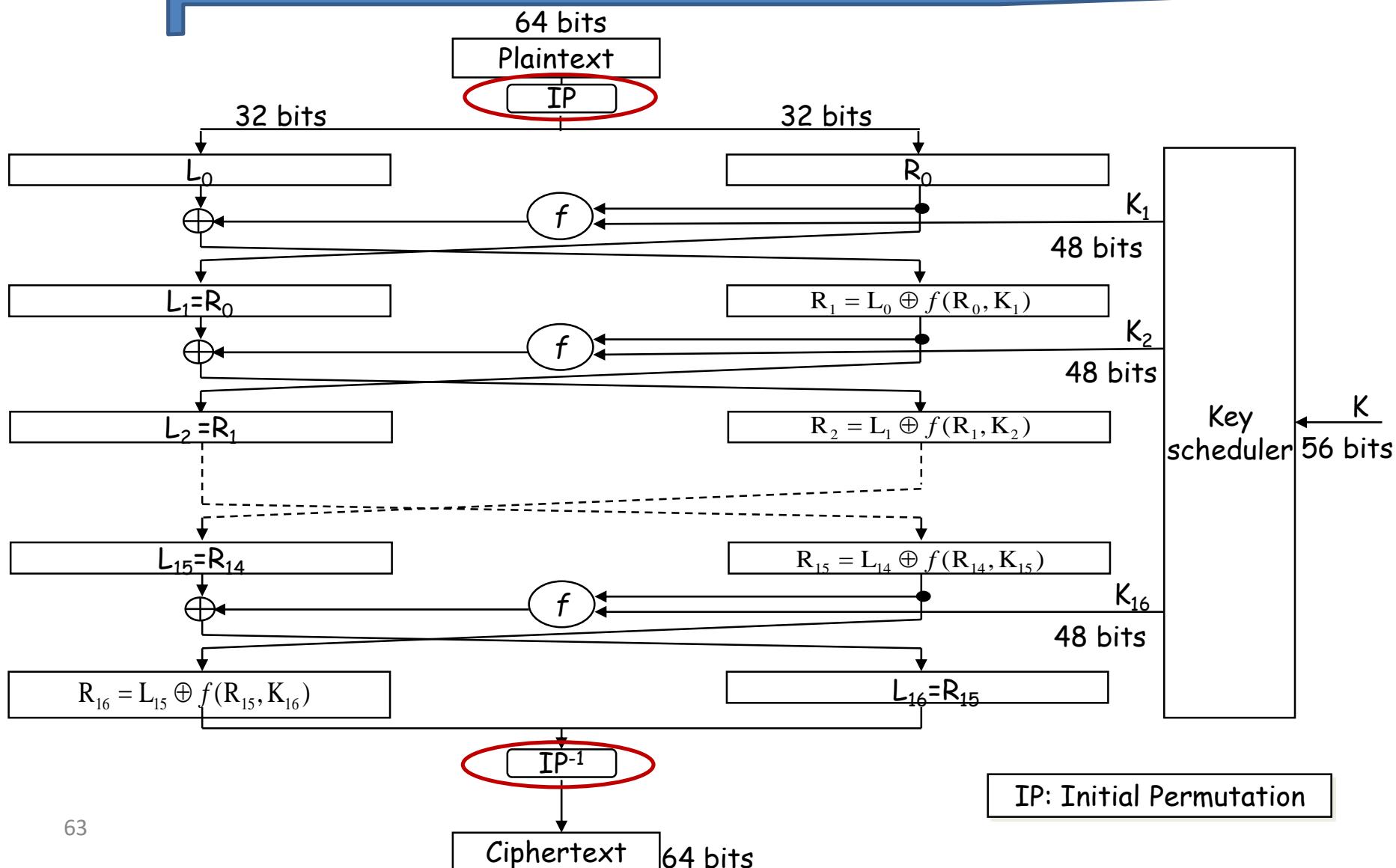
$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \text{ XOR } f(R_{i-1}, K_i)$$

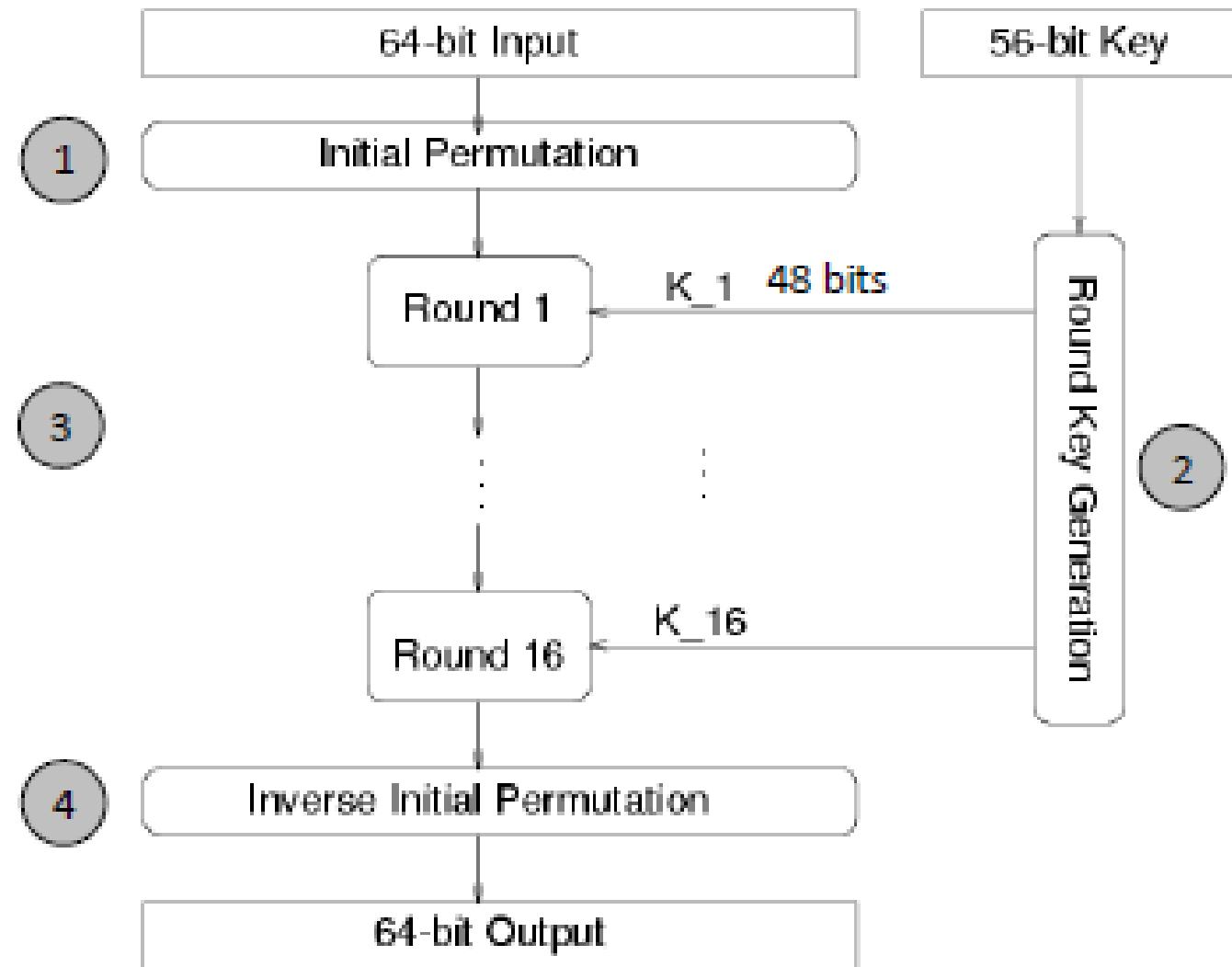
- **Phase 4 : Permutation finale.**

On applique à L₁₆ R₁₆ l'inverse de la permutation initiale. Z=P⁻¹(L₁₆R₁₆) est le bloc de 64 bits chiffré à partir de x.

L'algorithme DES



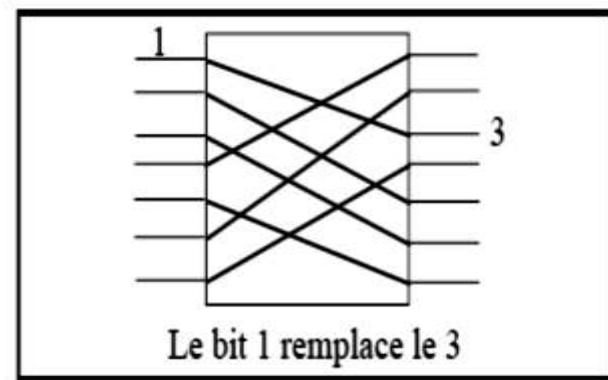
L'algorithme DES



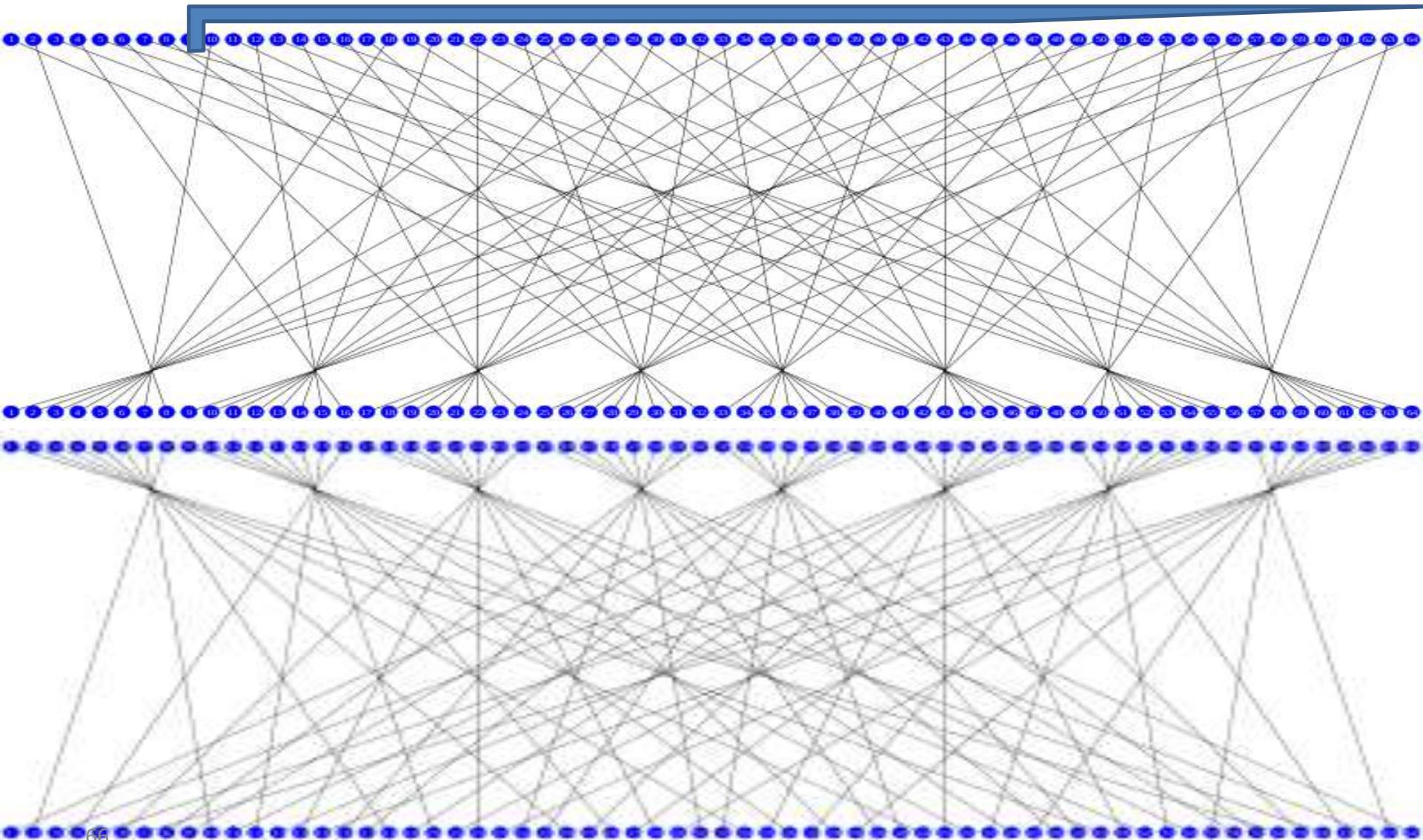
Phase1: Permutation Initiale

IP	IP ⁻¹
58 50 42 34 26 18 10 2	40 8 48 16 56 24 64 32
60 52 44 36 28 20 12 4	39 7 47 15 55 23 63 31
62 54 46 38 30 22 14 6	38 6 46 14 54 22 62 30
64 56 48 40 32 24 16 8	37 5 45 13 53 21 61 29
57 49 41 33 25 17 9 1	36 4 44 12 52 20 60 28
59 51 43 35 27 19 11 3	35 3 43 11 51 19 59 27
61 53 45 37 29 21 13 5	34 2 42 10 50 18 58 26
63 55 47 39 31 23 15 7	33 1 41 9 49 17 57 25

- Comme résultat de la permutation initiale IP, le 58^{ème} bit devient le 1^{er}, le 50^{ème} devient le second, ...
- IP⁻¹ est la fonction inverse de la permutation initiale IP

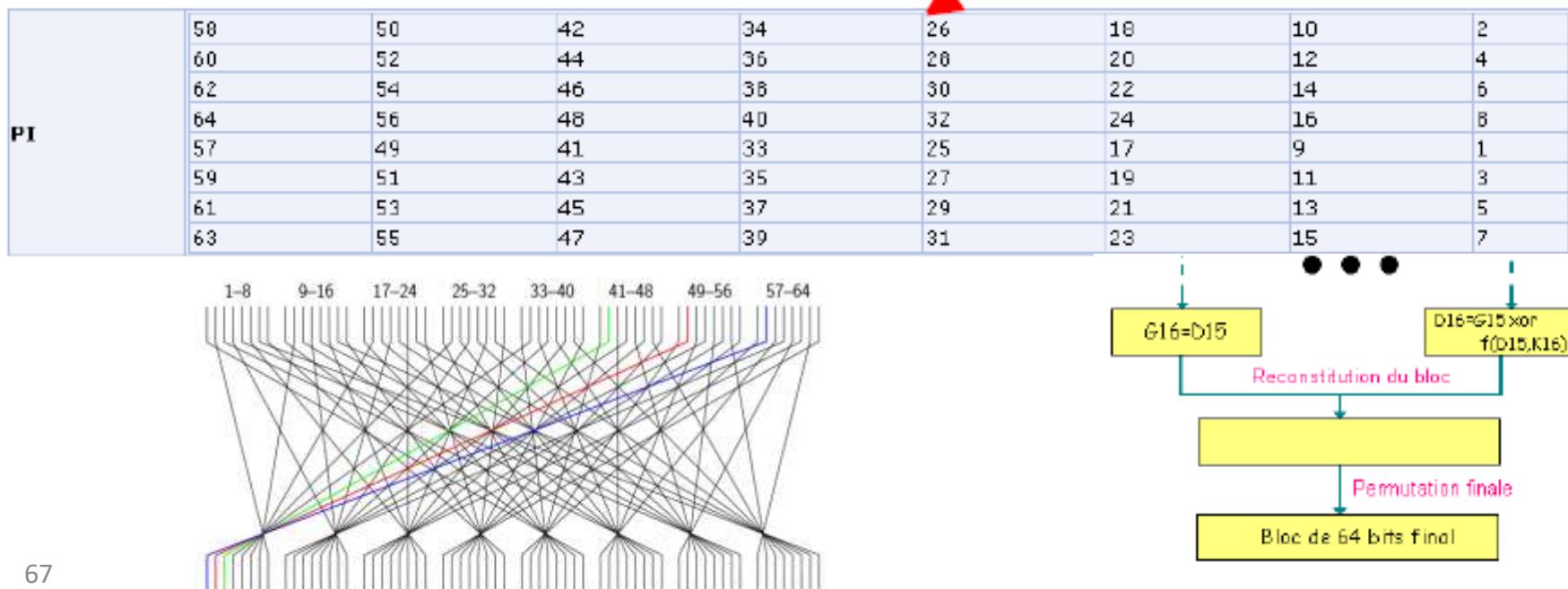


Permutation Initiale



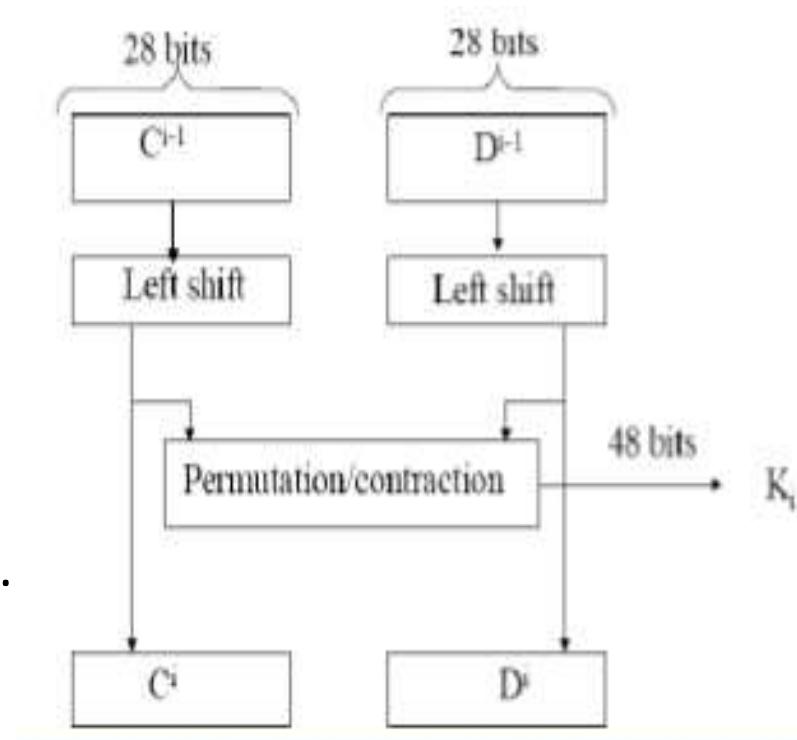
Permutation initiale

Cette matrice de permutation indique, en parcourant la matrice de gauche à droite puis de haut en bas, que le 58^{ème} bit du bloc de texte de 64 bits se retrouve en première position, le 50^{ème} en seconde position et ainsi de suite...



Phase2: Génération des clés

- Les sous-clés (*Round keys*) sont générées à partir de la clé principale de 56 bits :
 - Diviser la clé de 56 bits en deux segments.
 - Rotation de chaque segment par un ou deux bits à gauche (selon le nombre de l'itération).
 - Sélection de 24 bits de chaque segment.



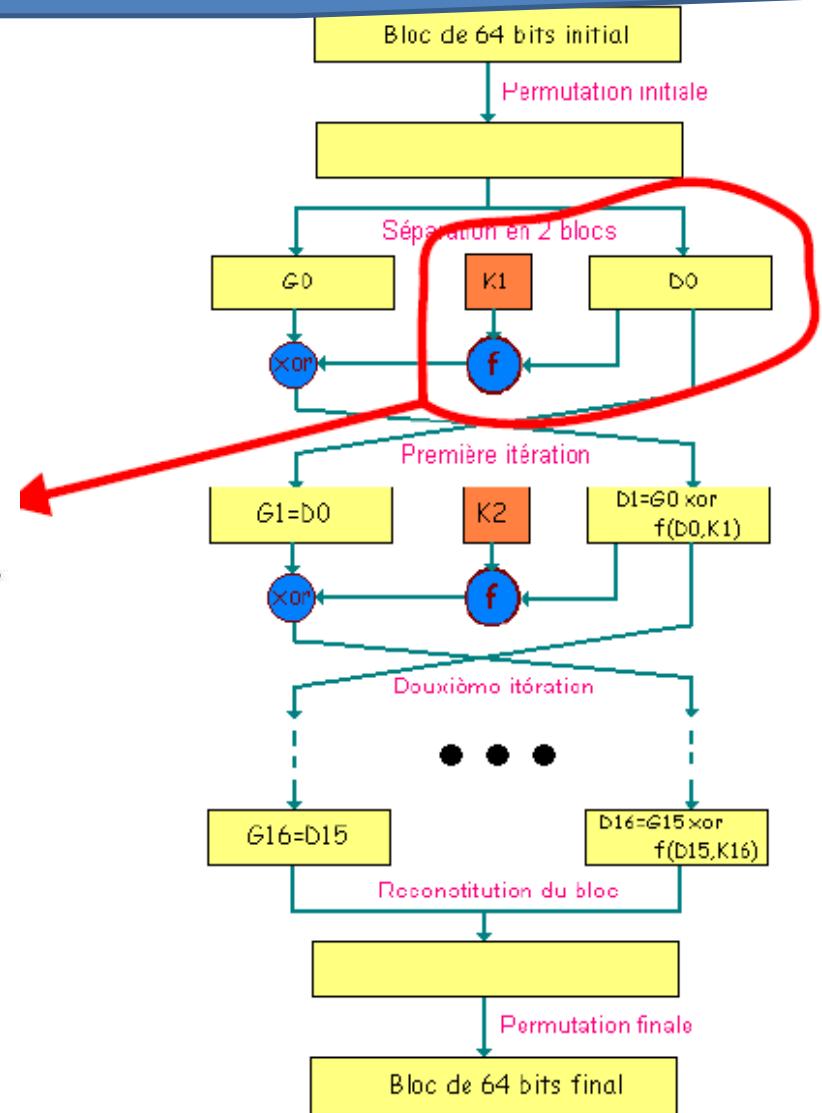
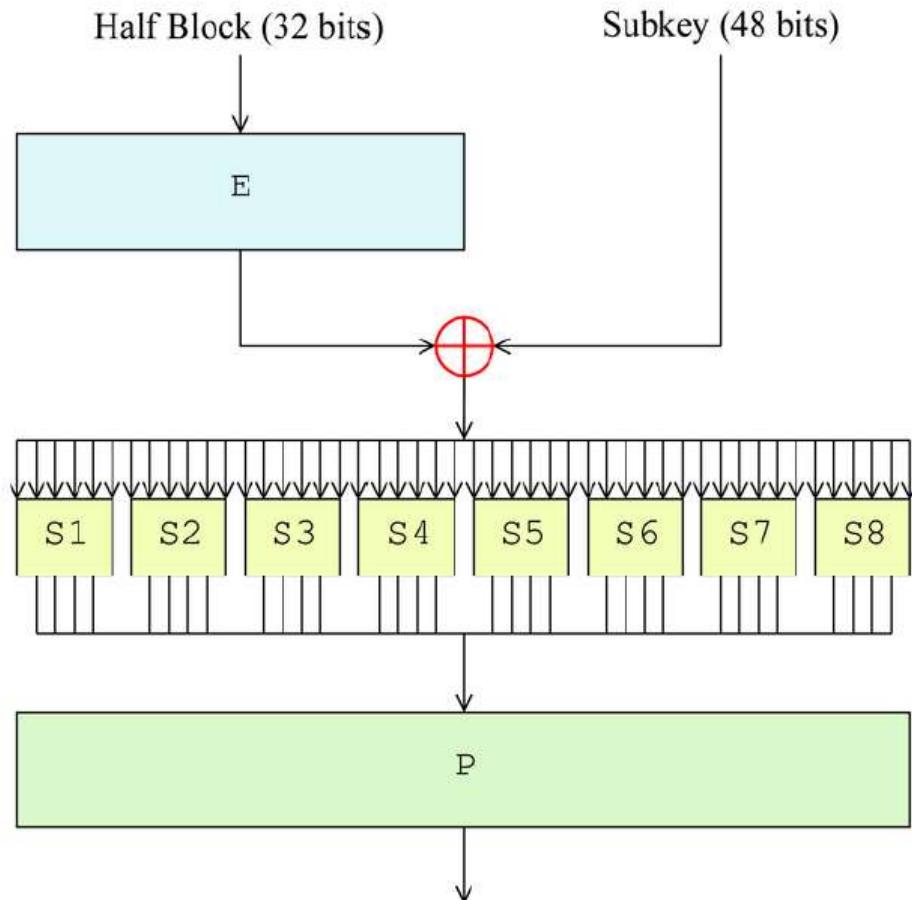
Génération des clés

- La clé secrète est transformée en 16 parties K_i de 48 bits.
- Puis, on permute les deux parties en introduisant une fonction de la clé.
 - $L_1 = R_0$.
 - $R_1 = L_0 + f(K_1, R_0)$.
 - Cette opération se répète 16 fois. A chaque étape i , on a :
 - $L_i = R_{i-1}$.
 - $R_i = L_{i-1} + f(K_i, R_{i-1})$.

Génération des clés

- K_i représente la sous clé numéro i obtenu à partir de la clé secrète.
- Le calcul de f se fait de la manière suivante :
 - les 32 bits de la partie R sont étendue à 48 bits grâce à une table appelée E (Expansion).
 - Ce nouveau R , $E(R)$ pour être plus précis, est additionné à K_i .
 - Le résultat est découpé en huit suites B_i de six bits : Grâce à la table S-Box, les données de ces huit suites donne un résultat de 32bits.

Phase 3 : Rondes (1/3)

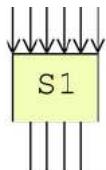


Phase 3 : Rondes (2/3)

- Fonction d'expansion ($32 \Rightarrow 48$ bits)

- Les 32 bits du bloc D_0 sont étendus à 48 bits grâce à une table (matrice) appelé table d'expansion (notée E), dans laquelle les 48 bits sont mélangés et 16 d'entre eux sont dupliqués (1, 4, 5, 8, 9, 12, 13, 16, 17, 20, 21, 24, 25, 28, 29, 32).

E	32	1	2	3	4	5
	4	5	6	7	8	9
	8	9	10	11	12	13
	12	13	14	15	16	17
	16	17	18	19	20	21
	20	21	22	23	24	25
	24	25	26	27	28	29
	28	29	30	31	32	1



- Fonction de substitution et de réduction ($48 \Rightarrow 32$ bits)

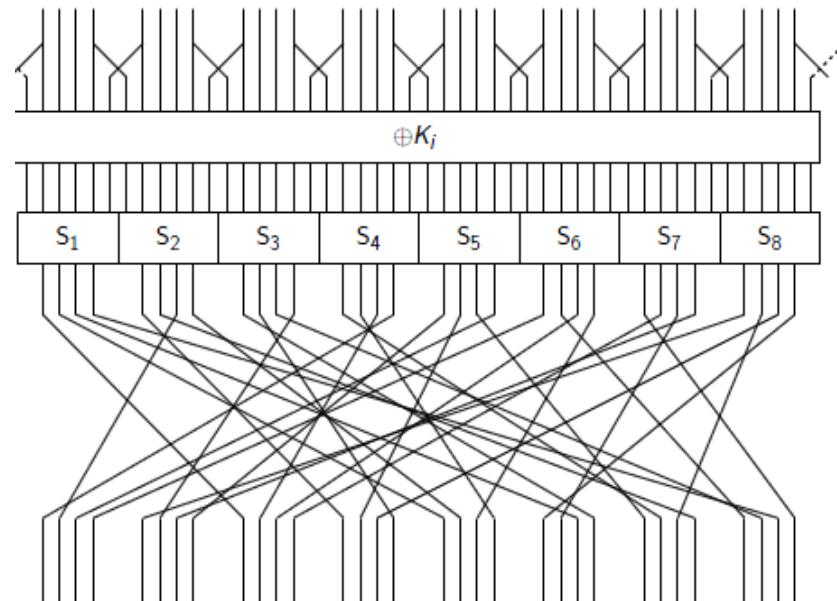
- Soit D_{01} égal à 101110. Les premiers et derniers bits donnent 10, c'est-à-dire 2 en binaire. Les bits 2,3,4 et 5 donnent 0111, soit 7 en binaire. Le résultat de la fonction de sélection est donc la valeur située à la ligne n°2, dans la colonne n°7. Il s'agit de la valeur 11, soit en binaire 1011.

S1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Phase 3 : Rondes (3/3)

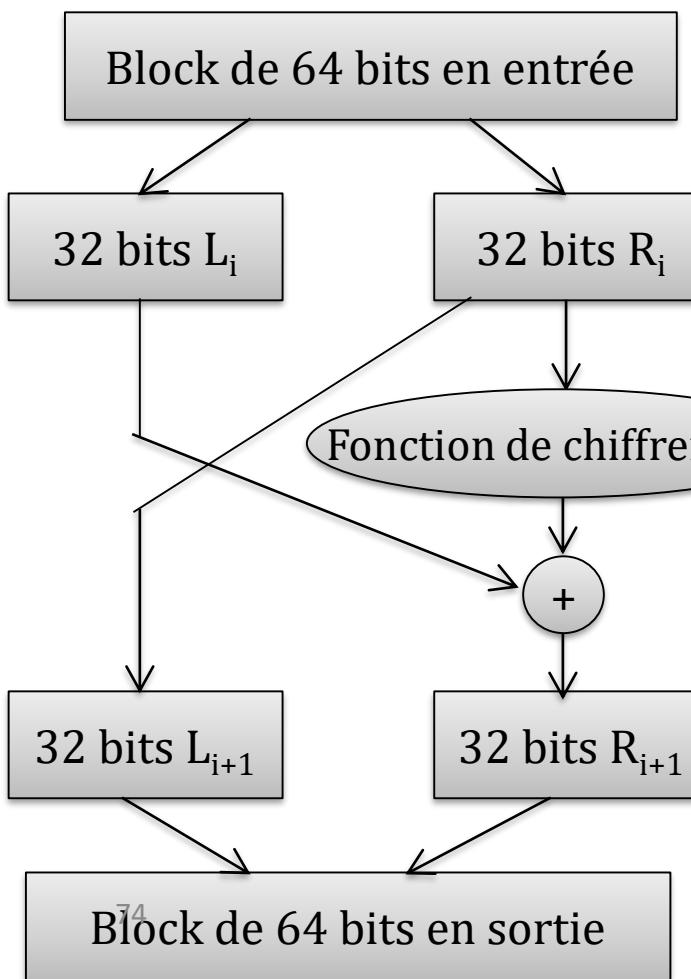
- Permutation
 - Le bloc de 32 bits obtenu est enfin soumis à une permutation P dont voici la table

P	16	7	20	21	29	12	28	17
	1	15	23	26	5	18	31	10
	2	8	24	14	32	27	3	9
	19	13	30	6	22	11	4	25

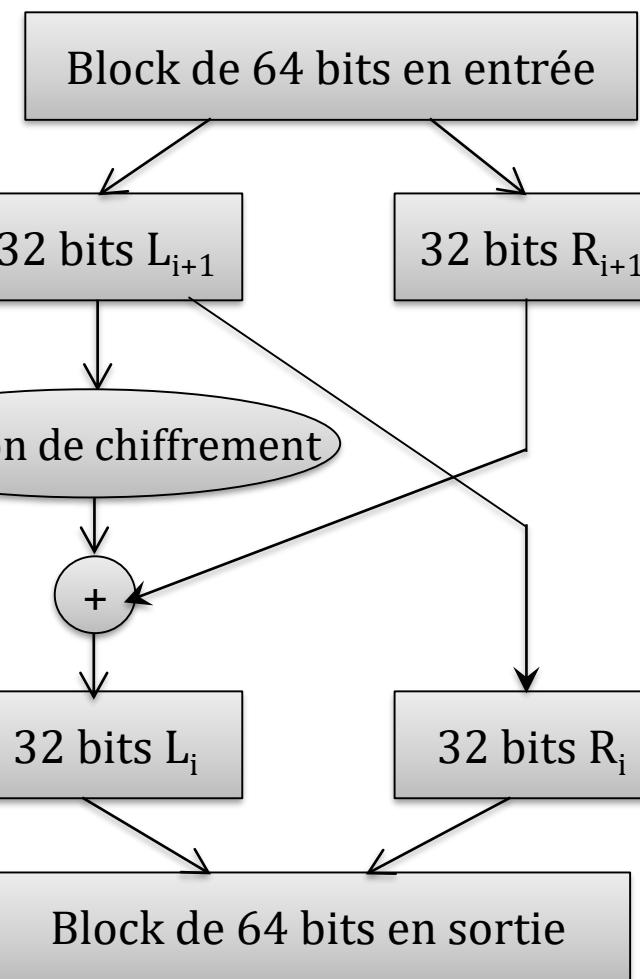


Déchiffrement en DES

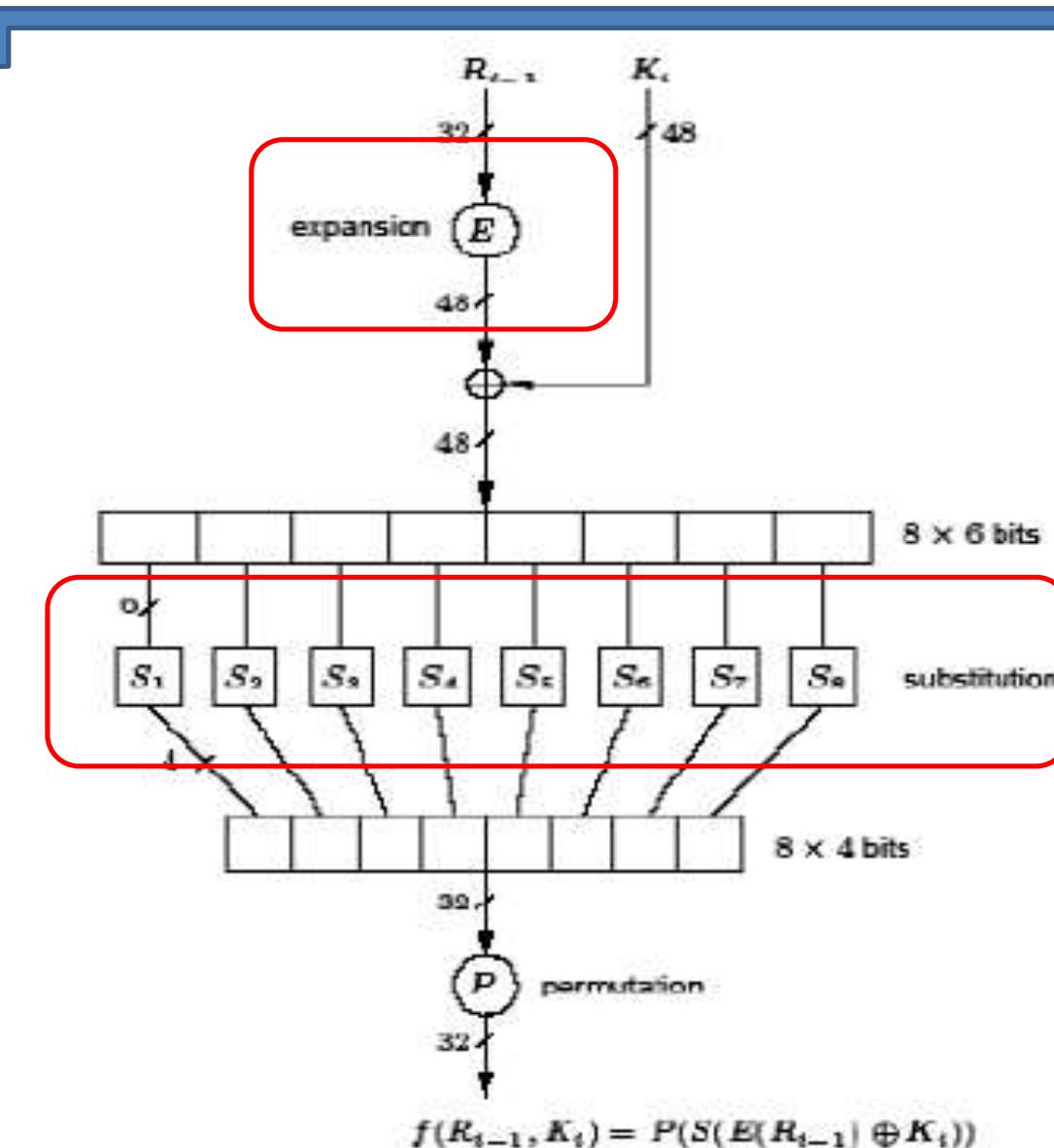
Chiffrement



Déchiffrement

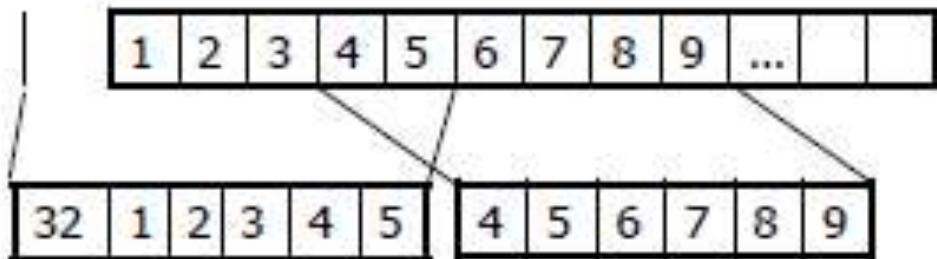


La fonction de cryptage



La fonction d'expansion

- Etendre les blocks d'entrée R_i de 32 bits à un block R'_i de 48 bits.
 - Diviser les 32 bits en des segments de 4 bits
 - Élargir chaque segment de 4 bits avec les bits de ses voisins pour atteindre 6 bits.
 - XOR des 48 bits en sortie avec la clé.

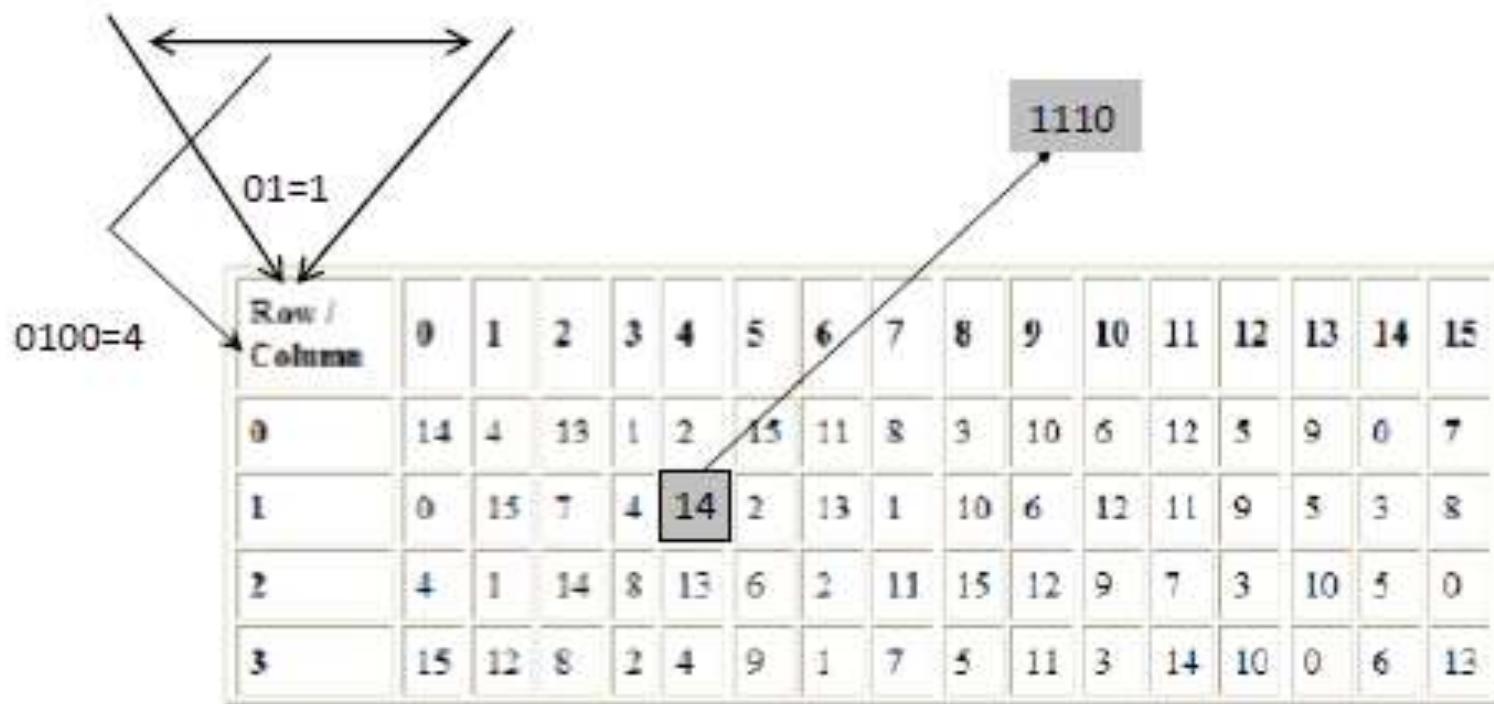
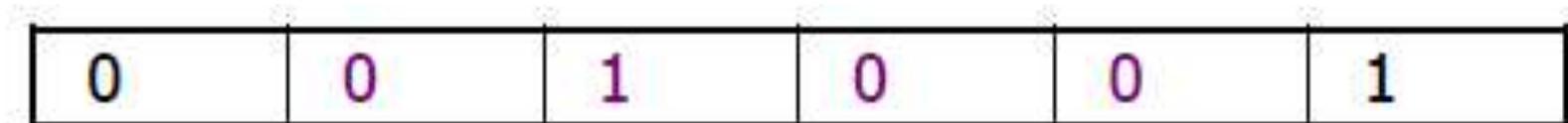


E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

La fonction de substitution

- Il y a 8 S-Box, une pour chacun B_i .
- Chaque S-Box à 16 colonnes et 4 lignes.
- $B_i = b_1 b_2 b_3 b_4 b_5 b_6$. On calcule
 - $r = b_1 b_6$
 - $c = b_2 b_3 b_4 b_5$.
- On regarde le nombre qui figure à la ligne r et à la colonne c . Il est codé sur 4 bits et correspond à la sortie $S_i(B_i)$.
- Ensuite on effectue une permutation représentée par une table appelée P et le résultat de cette permutation est retourné par la fonction f .
- Pour le déchiffrement, il suffit de faire l'opération inverse.

La fonction de substitution



Un sous-bloc de 6 bits est transformé en un sous-bloc de 4 bits.

La fonction de permutation

<i>P</i>			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

- Bit en position 1 est envoyé en position 16

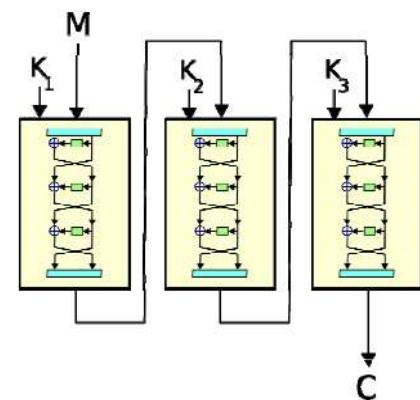
Phase 4 : Permutation finale

- Le résultat en sortie est un texte chiffré de 64 bits

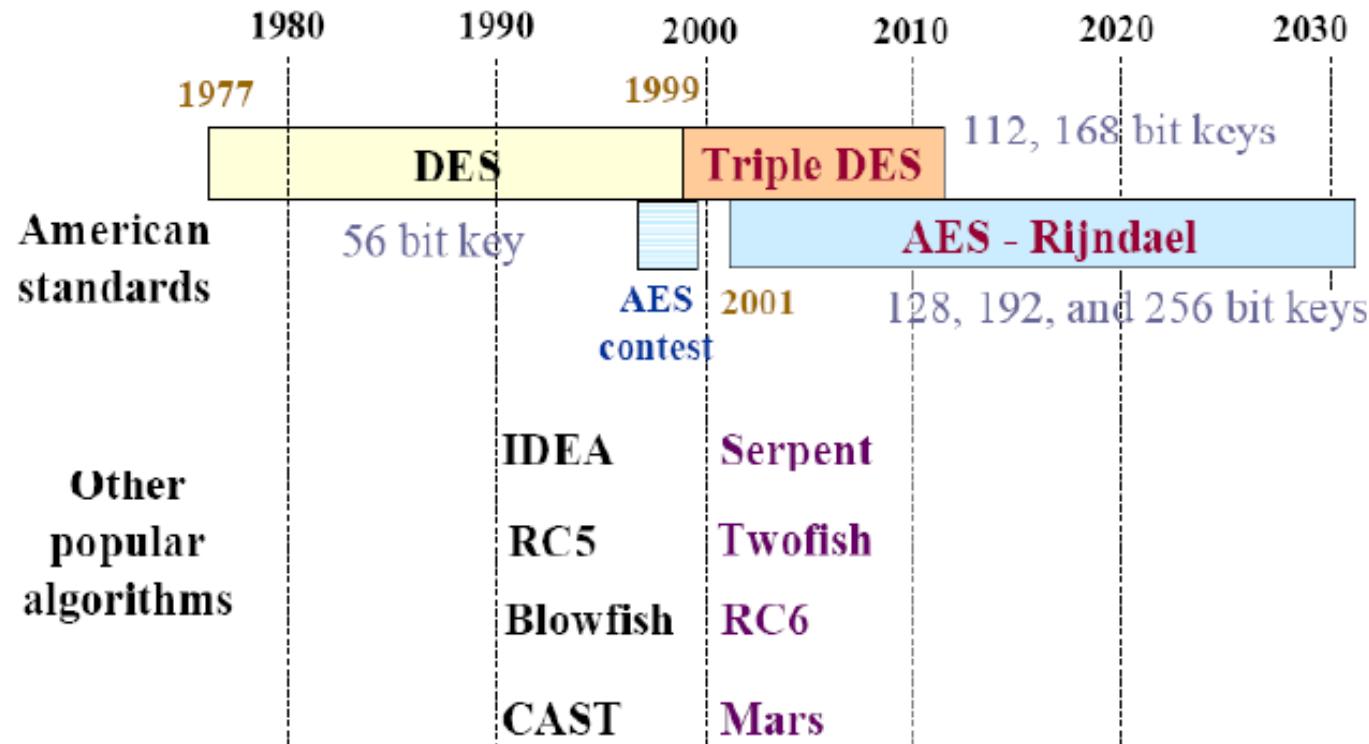
P1-1	40	0	40	16	56	24	64	32
	39	7	47	15	55	23	63	31
	38	6	46	14	54	22	62	30
	37	5	45	13	53	21	61	29
	36	4	44	12	52	20	60	28
	35	3	43	11	51	19	59	27
	34	2	42	10	50	18	58	26
	33	1	41	9	49	17	57	25

Passage vers le Triple-DES

- Les progrès en cryptanalyse et en électronique a fait que la longueur 56 des clés est devenu un problème pour DES . La clé est retrouvée en quelques heures à partir d'un texte clair connu en faisant de « *la force brute* ».
- 3-DES (triple DES) a été lancé comme un nouveau standard en 1999.
 - Utilise 2 ou 3 clés.
 - Niveau de sécurité satisfaisant.
 - Permet de continuer l'utilisation des boîtes S-Box et P-Box matériel et logiciel, en attendant la migration vers AES.
- On distingue habituellement plusieurs types de chiffrement triple DES
 - DES-EEE3 : 3 chiffrements DES avec 3 clés différentes
 - DES-EDE3 : une clé différente pour chacune des 3 opérations DES (chiffrement, déchiffrement, chiffrement)
 - DES-EEE2 et DES-EDE2 : une clé différente pour la seconde opération (déchiffrement)



Algorithmes de chiffrement symétrique



AES : historique d'une compétition

- En 1997 le NIST (National Institute of Standards and Technology) lance un nouvel appel à projet pour élaborer l'AES (Advanced Encryption Standard), un algorithme de chiffrement destiné à remplacer le DES

June 1998

15 Candidates

from USA, Canada, Belgium, France, Germany, Norway, UK, Isreal, Korea, Japan, Australia, Costa Rica

Round 1

Security
Software efficiency

North America (8)

Canada:

CAST-256
Deal

USA:

Mars
RC6
Twofish
Safer+
HPC

Costa Rica:

Frog

Europe (4)

Germany:

Magenta

Belgium:

Rijndael

France:

DFC

Israel, UK, Norway:

Serpent

Asia (2)

Korea:

Crypton

Japan:

E2

Australia (1)

Australia:

LOKI97

August 1999

5 final candidates

Mars, RC6, Rijndael, Serpent, Twofish

Round 2

Security
Hardware efficiency

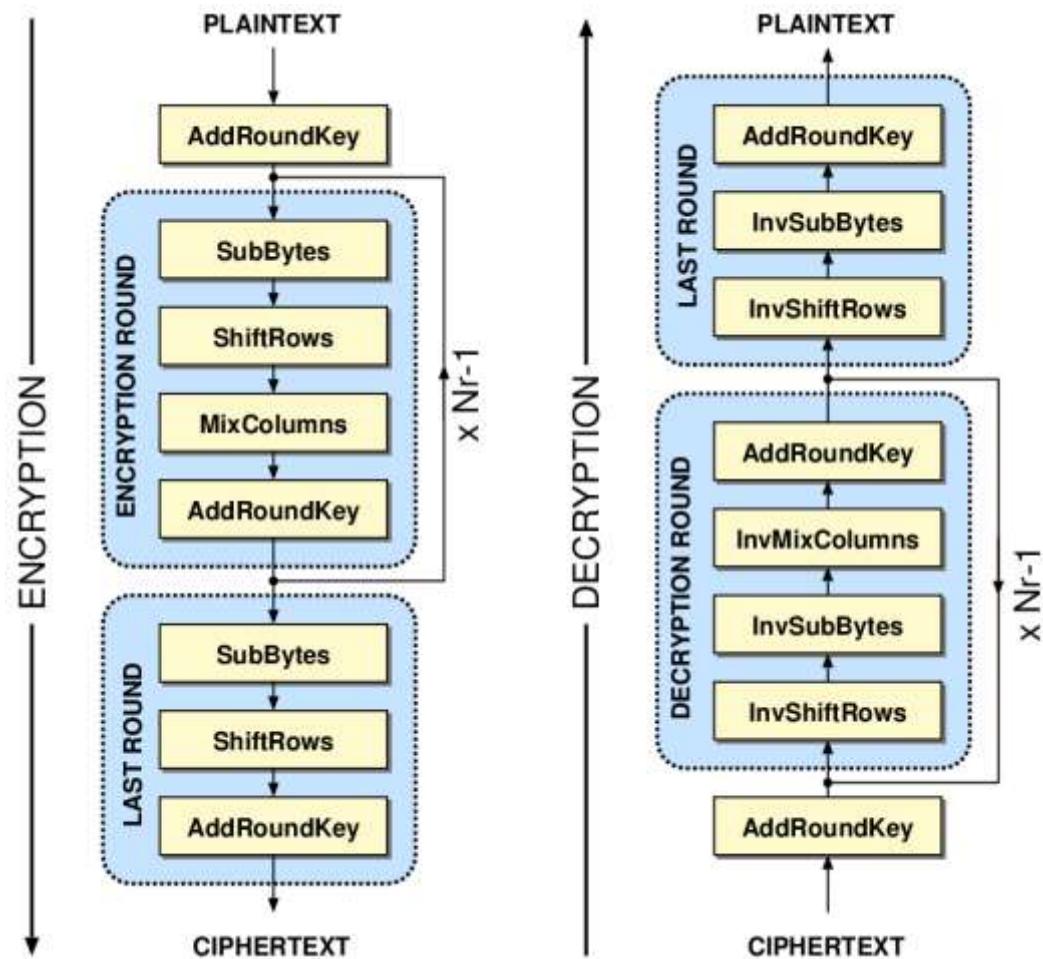
October 2000

1 winner: Rijndael

Belgium

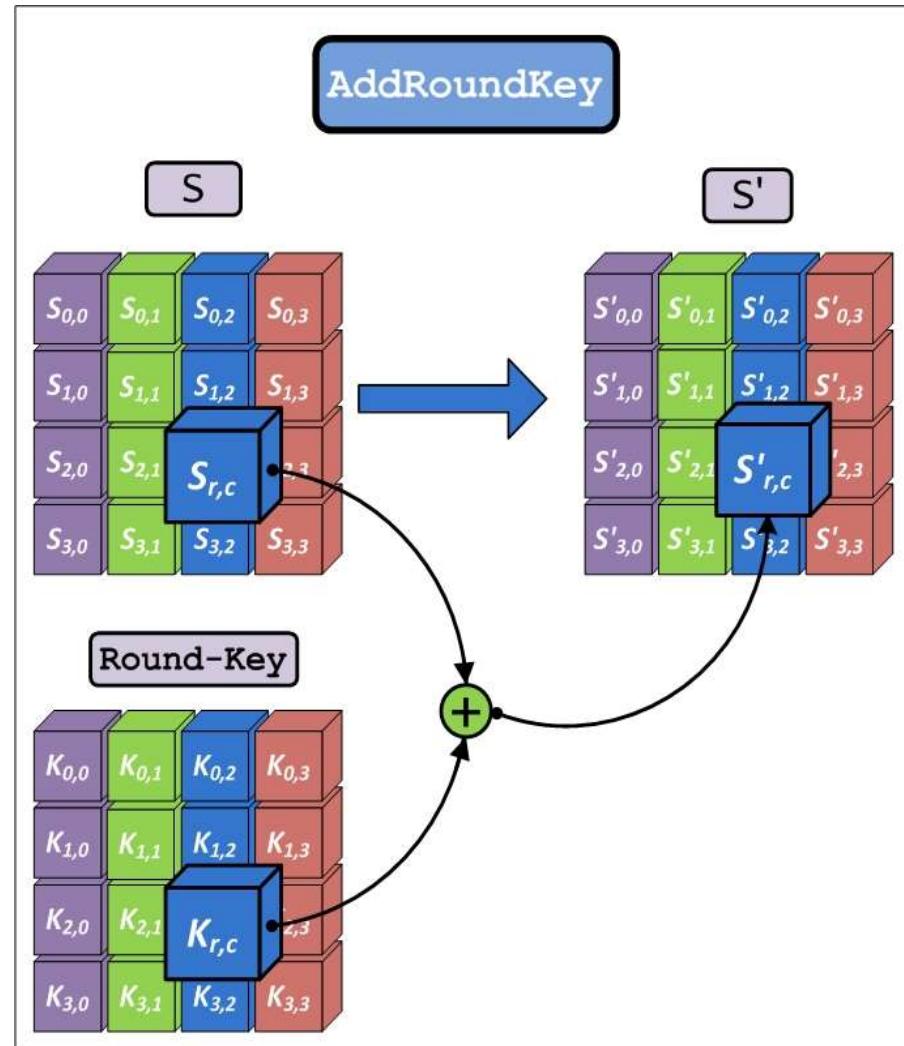
Étapes de l'AES 128

- Il s'agit d'une succession de fonctions à appliquer en 10, 12 ou 14 itérations.
- Les données sont codés en 128 bits
- La clé est codée en 128, 192 ou 256 bits
- Chaque bloc est exprimé en byte. Chaque byte représente un polynôme dans le corps galois GF(28).
- Ce corps admet un polynôme irréductible Le polynôme irréductible sera remplacé $m(x)=x^8+x^4+x^3+x+1$



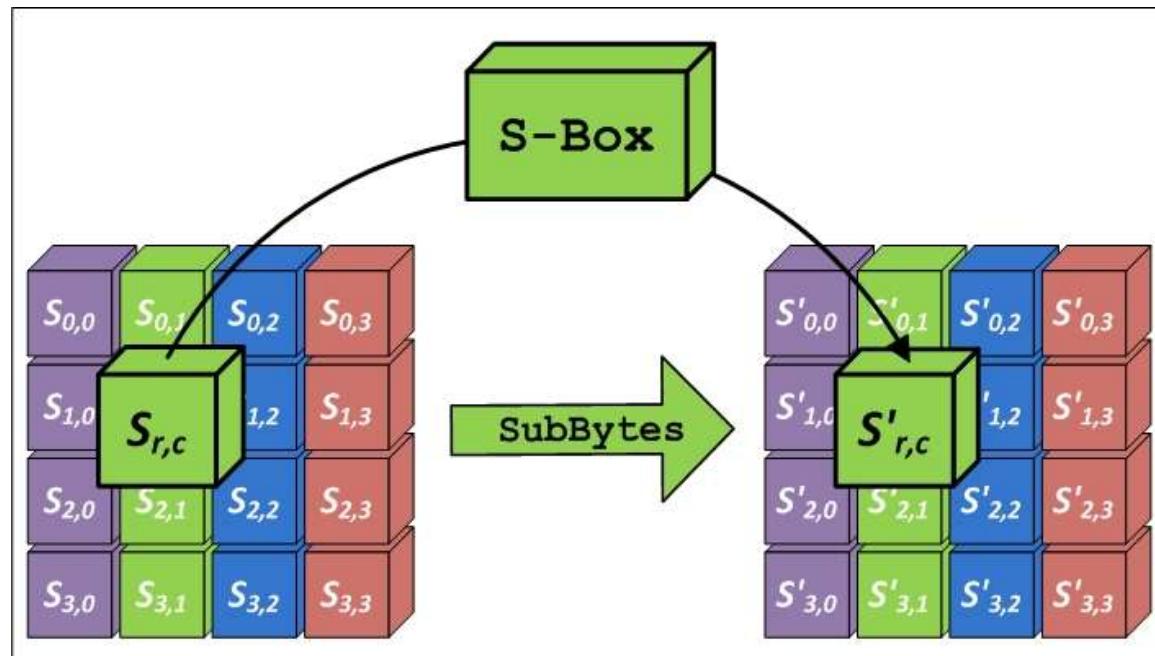
AddRoundKey

Il s'agit d'une addition binaire des bytes de la donnée avec des bytes de la clé (soit alors par l'application de XOR)



SubBytes

- Il s'agit d'une étapes de substitutions de bytes des données selon une matrice S-Box prédéfinie.



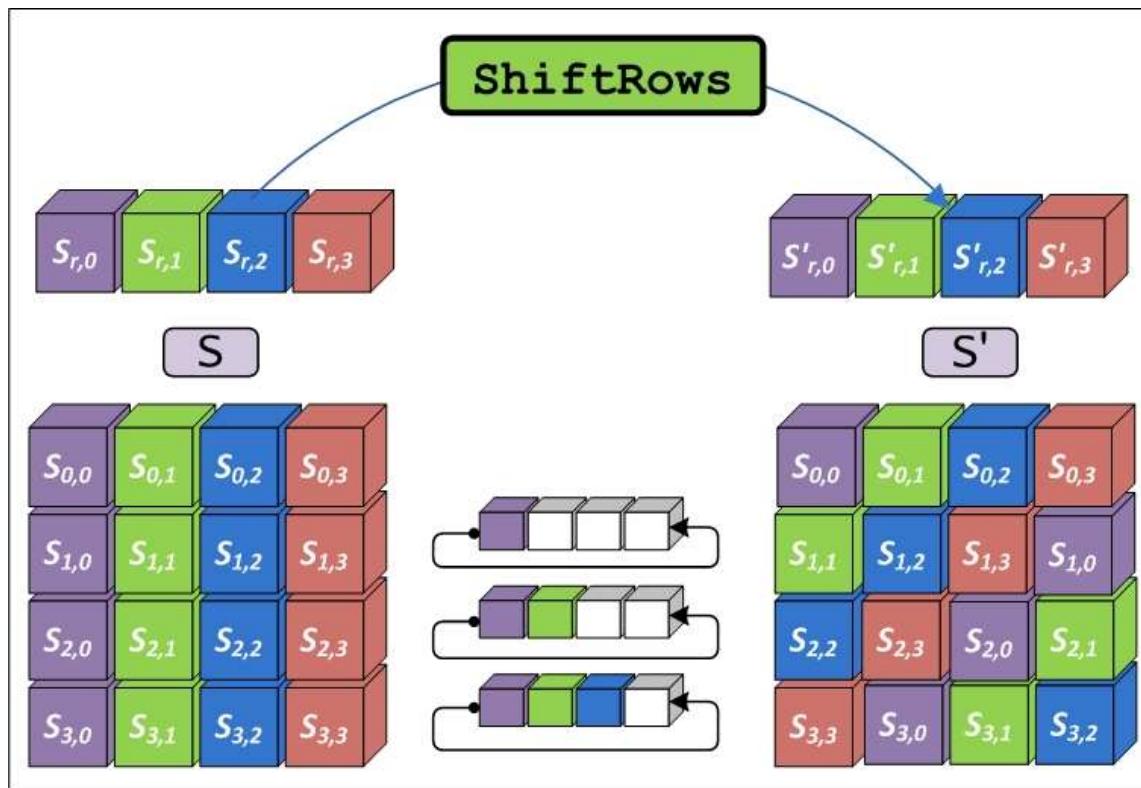
SubBytes

- Par exemple si $s_{11}=\{53\}$, il sera substitué par la case dans la ligne 5 et colonne 3, donc par {ed}.

		y																
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	
		0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
		1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
		2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
		3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
		4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
		5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
		6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
		7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
		8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
		9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
		a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
		b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
		c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
		d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
		e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
		f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

ShiftRows

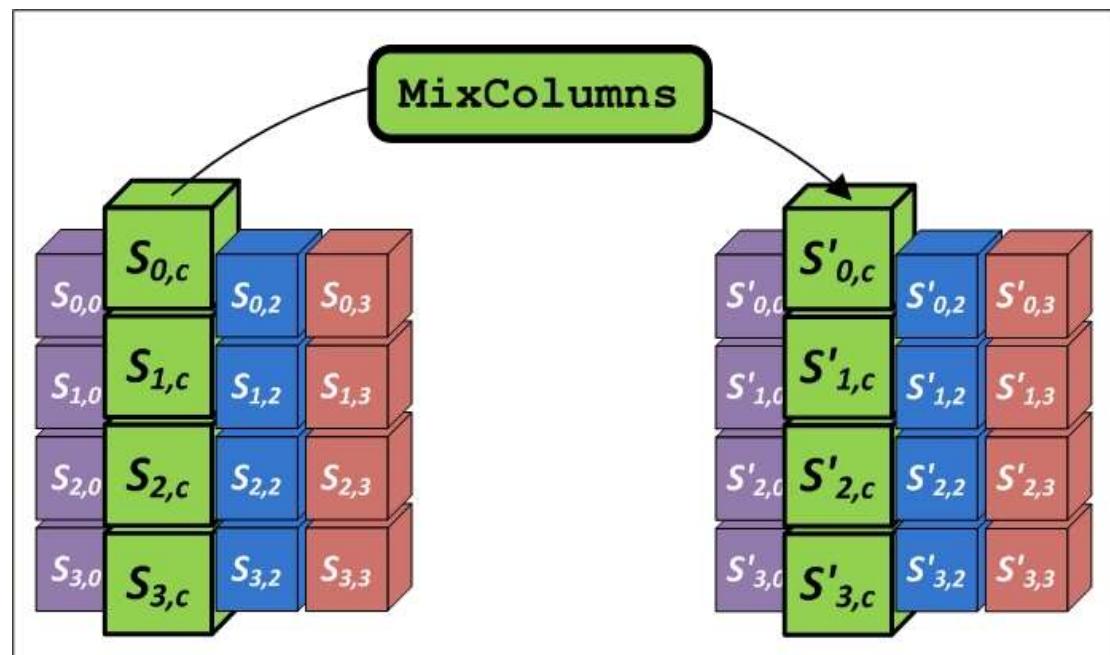
- Décalage circulaire des lignes l_i de la donnée par $i-1$ bytes



MixColumns

- Chaque byte représente un polynôme. On applique à ces polynômes des multiplications par 03 ($P(x)=x+1$) , 02 ($P(x)=x$) ou 01 ($P(x)=1$) aux colonnes.

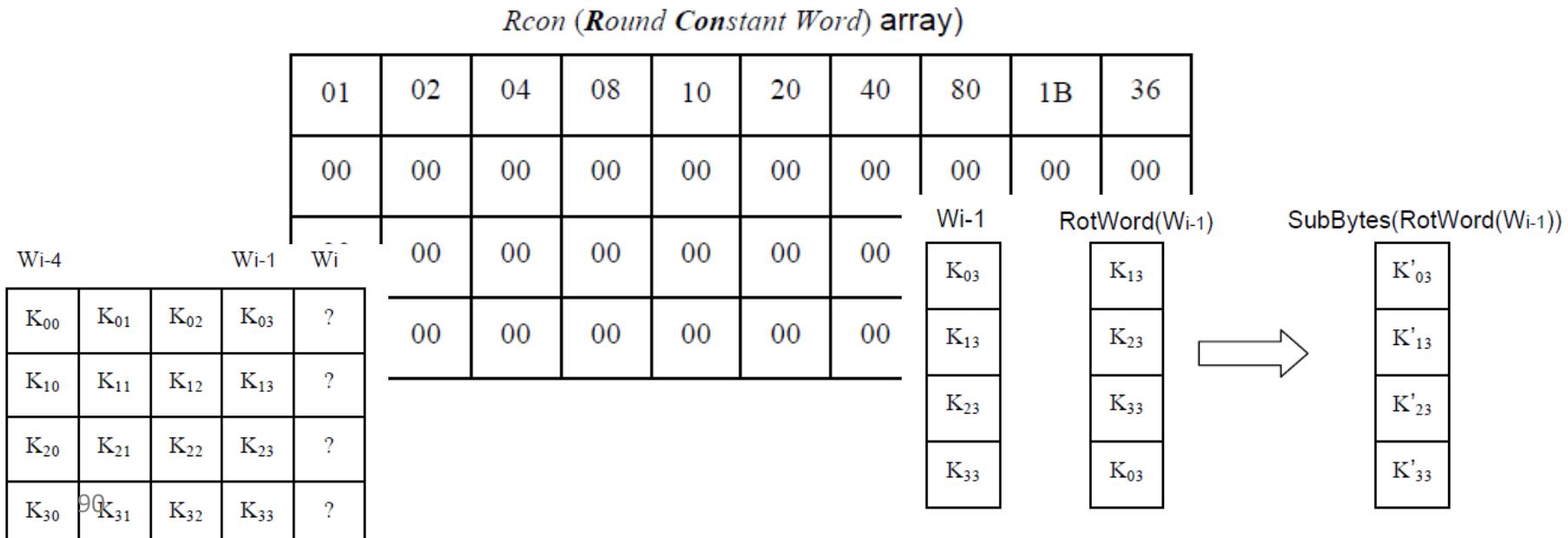
$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$



- Le polynôme irréductible sera remplacé $m(x)=x^4+x^3+x+1$

Génération de la clé

- Les quatre vecteurs composant la matrice *Key* sont appelés *Words*, ce sont des mots de 32 bits.
- Nous allons commencer tout d'abord par calculer le cinquième vecteur (W_i). Puis, nous allons prendre le vecteur W_{i-1} et nous lui appliquons une opération appelée *RotWord* qui consiste en un simple décalage des quatre octets du vecteur vers le haut. Au résultat, nous appliquons l'opération *SubBytes* que nous avons déjà définie.



Outils de cryptographie et d'authentification

Généralités

Chiffrement symétrique

Chiffrement asymétrique

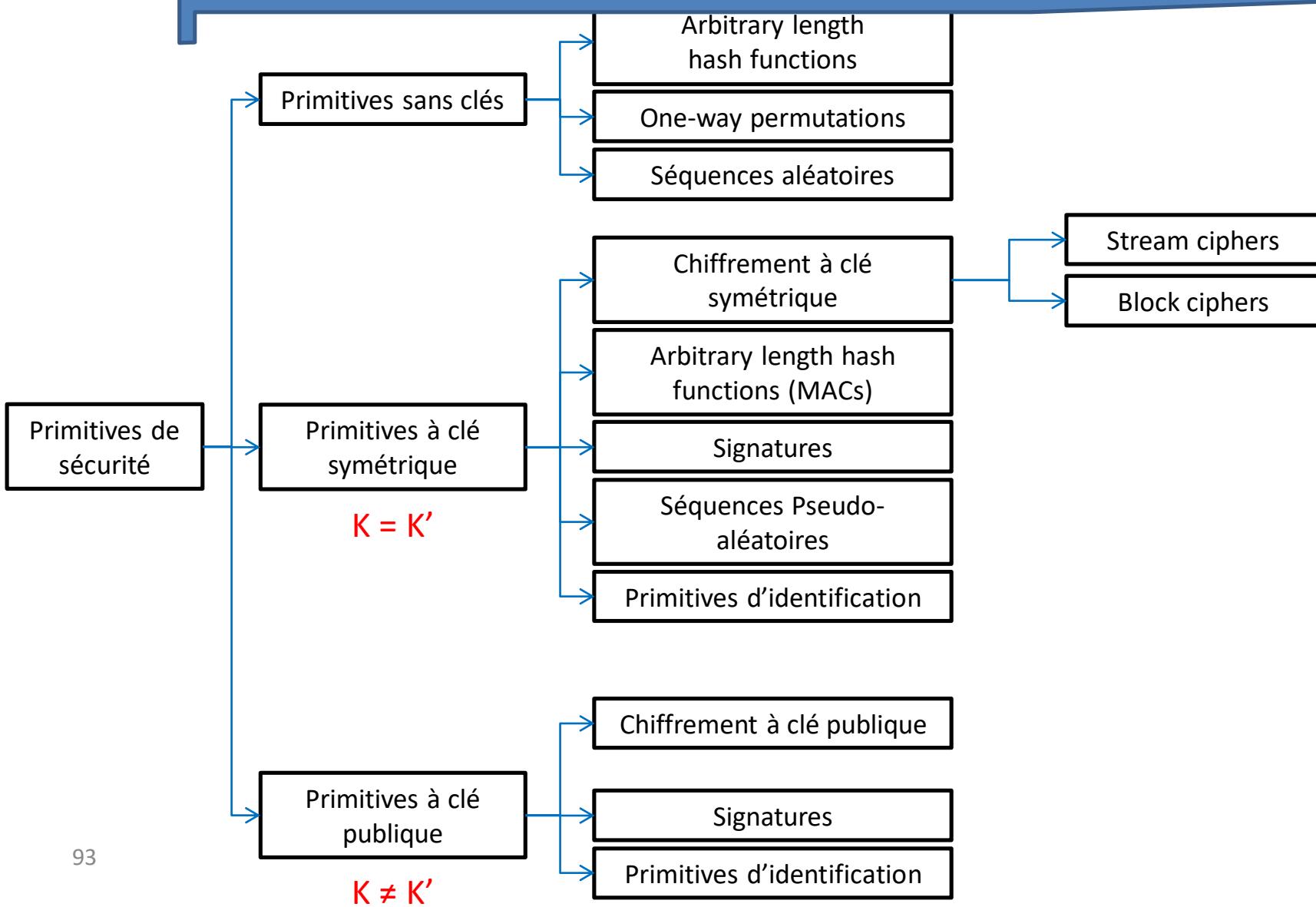
Intégrité d'un message

PKI

Chiffrement asymétrique

- Appelé aussi: cryptographie à clé publique / à paire de clés / asymétrique
- Représente une révolution dans l'histoire de la cryptographie
- Utilisation de deux clés:
 - Clé publique: Connue par tout le monde, et peut être utilisée pour crypter des messages ou pour vérifier la signature.
 - Clé privée: Connue par le récepteur uniquement, utilisée pour déchiffrer les messages, ou pour créer la signature.
- Si on crypte avec l'une de ces clés le déchiffrement se fait uniquement avec l'autre.
- Impossible de trouver la clé privée à partir de la clé publique.

Primitives Cryptographiques



Chiffrement asymétrique

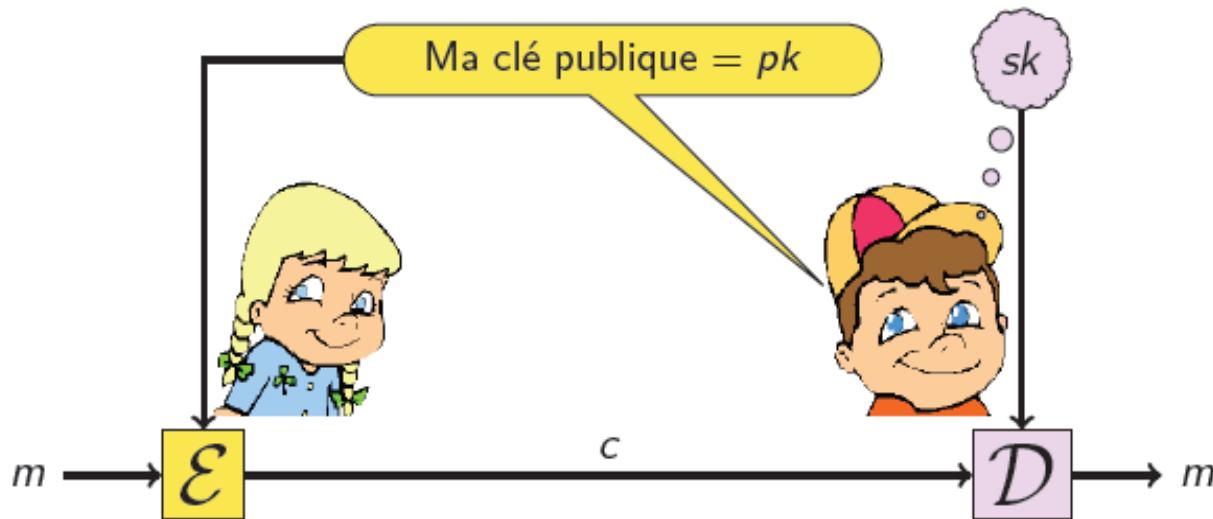
- Clés à grande taille (ex: RSA: 1024-2048-...).
- Utilisé généralement pour :
 - Cryptage / décryptage: assurer la confidentialité.
 - Signature numérique: assurer l'authentification et la non répudiation.
 - Distribution de clés: se mettre d'accord sur une clé de session.

Chiffrement asymétrique

- Confidentialité: Alice utilise la clé publique de Bob pour coder un message que seul Bob (en possession de la clé privée) peut décoder
- Authentification: Bob utilise sa propre clé privée pour coder un message que Alice peut décoder avec la clé publique
- Algorithmes de génération de paires de clés
- Algorithme de chiffrement
- Algorithme de déchiffrement
- Exemples:
 - RSA (Rivest Shamir Adleman).
 - ElGamal
 - ECC

Chiffrement asymétrique

- Certains algorithmes de chiffrement/déchiffrement (exemple: RSA) reposent sur un couple {clé privée sk , clé publique pk }.
 - La clé de chiffrement pk est publique
 - La clé de déchiffrement sk est privée

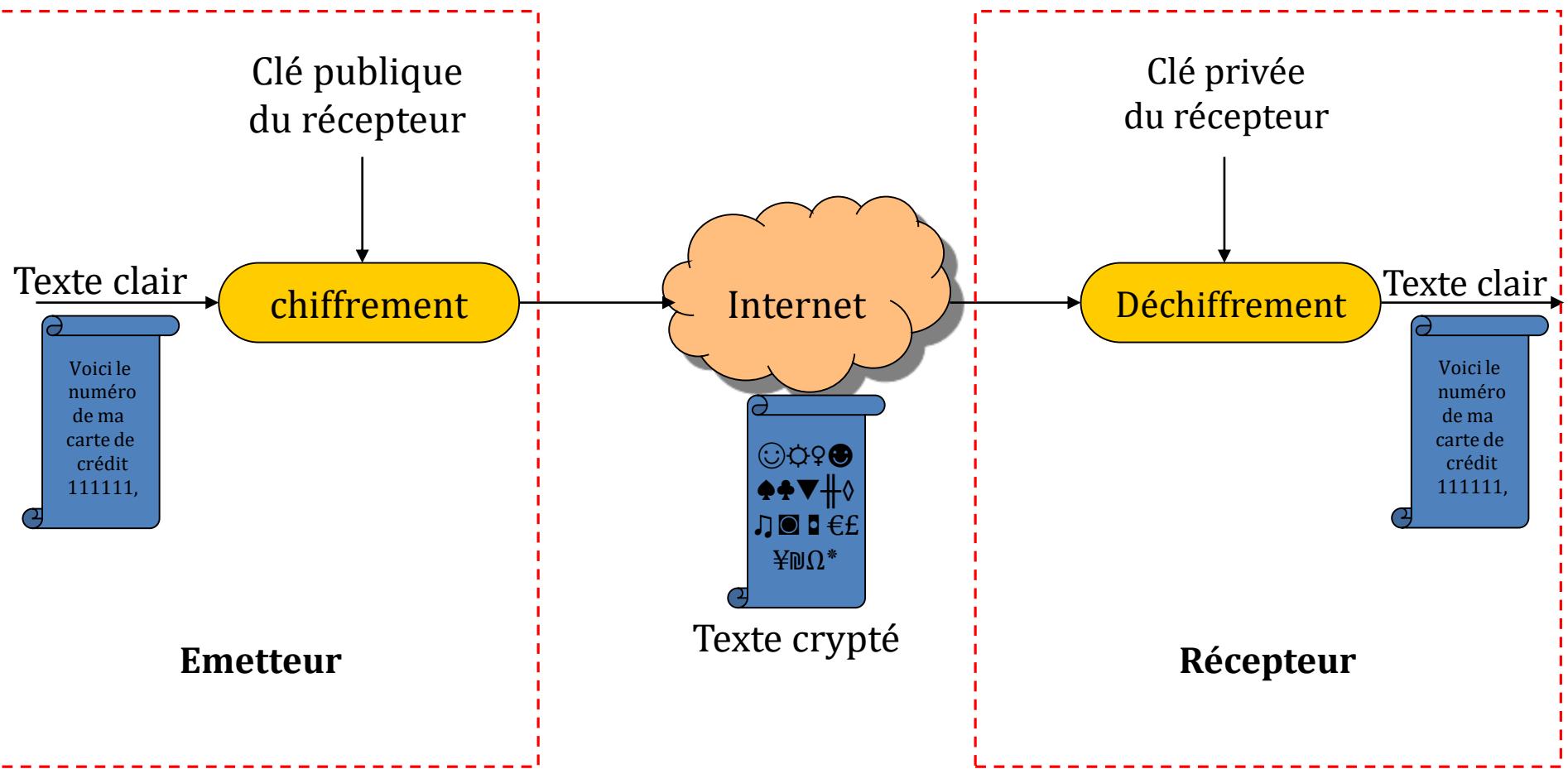


Définition

- Un chiffrement à clé publique se compose de trois algorithmes
- **Algorithme de génération de clé**
 $\mathcal{KG}(\ell)=\{pk, sk\}$ à partir d'un paramètre de sécurité; il produit un couple (clé publique, clé privée)
- **Algorithme de chiffrement**
 $E(m, pk)=c$ utilise la clé publique pour chiffrer m
- **Algorithme de déchiffrement**
 $\mathcal{D}(c, sk)=m$ utilise la clé privée pour remonter à m

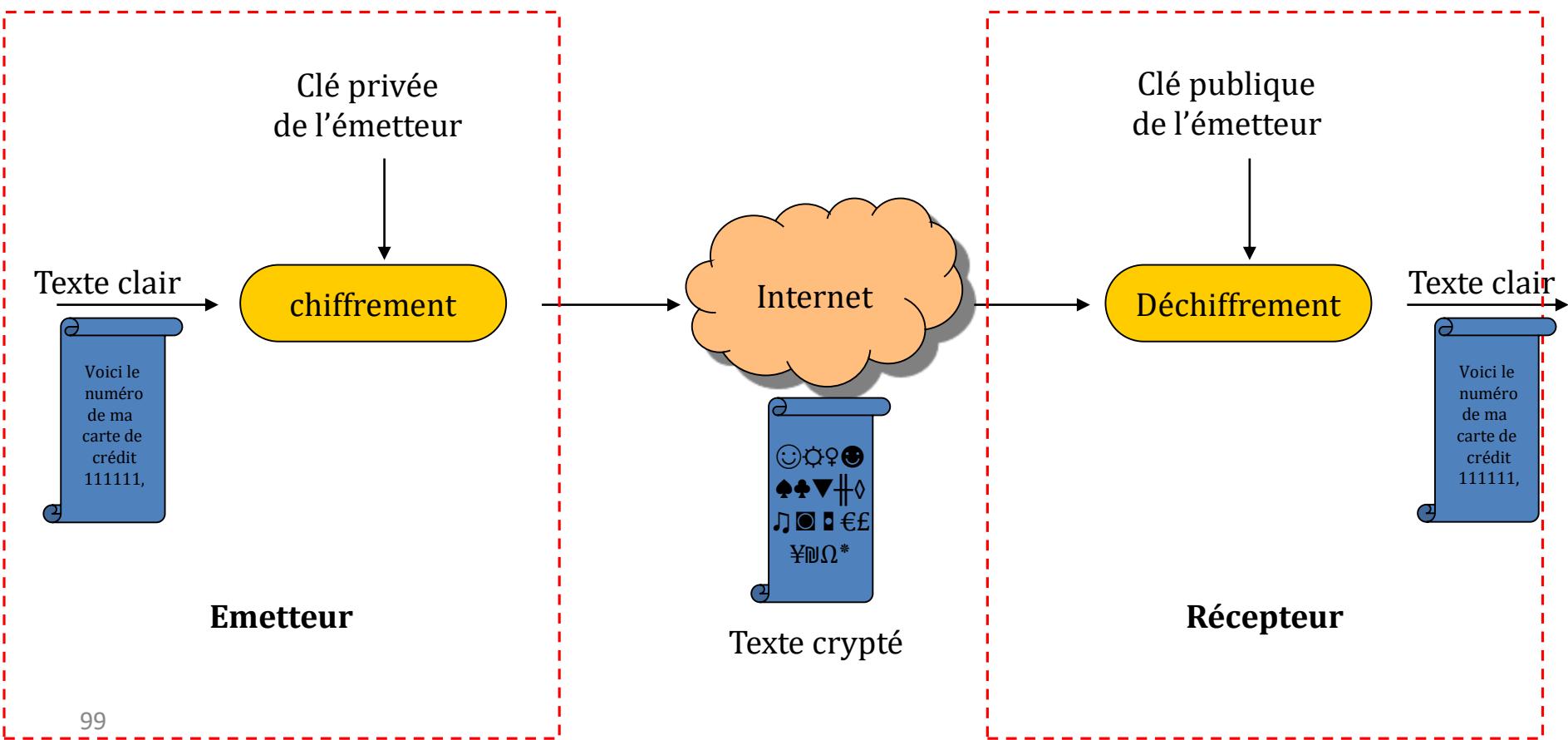
Scénario d'utilisation : confidentialité

Scénario: confidentialité

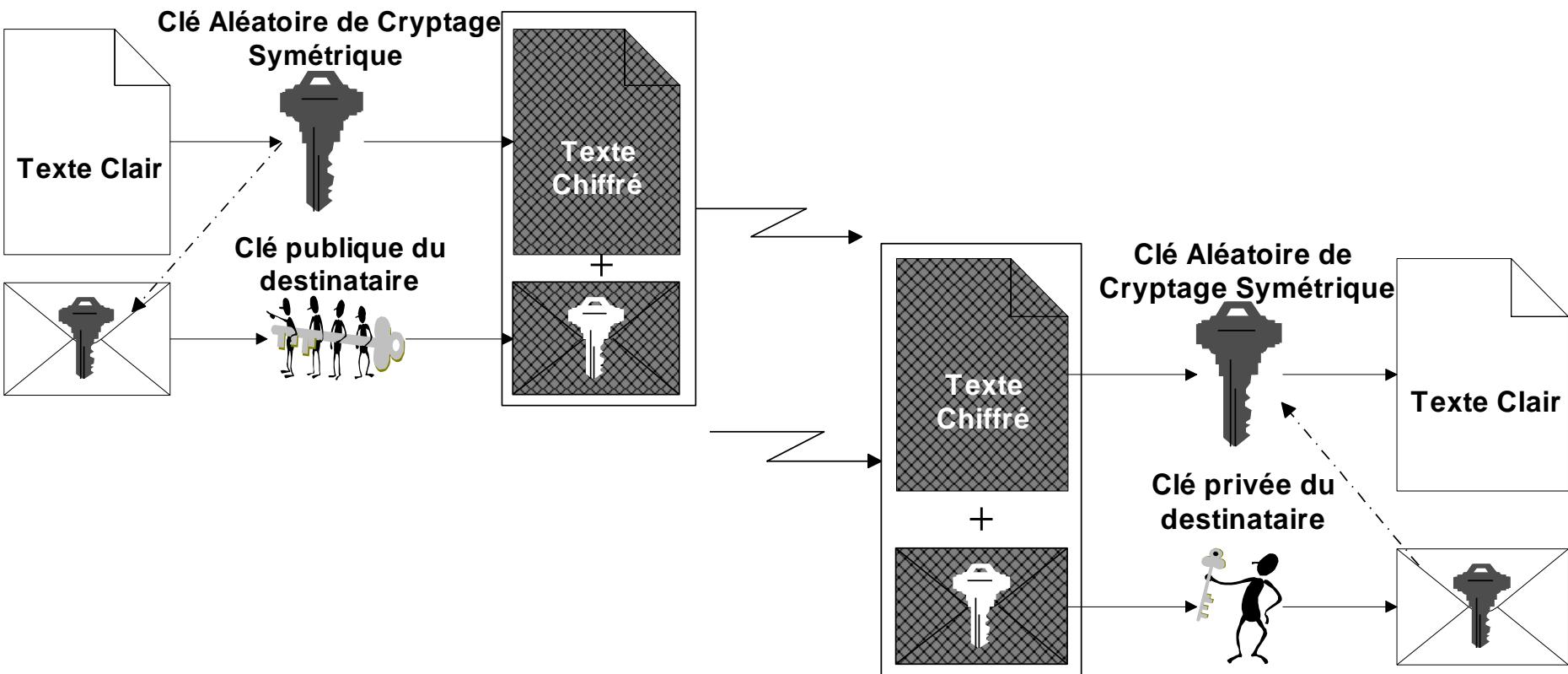


Scénario d'utilisation : non répudiation

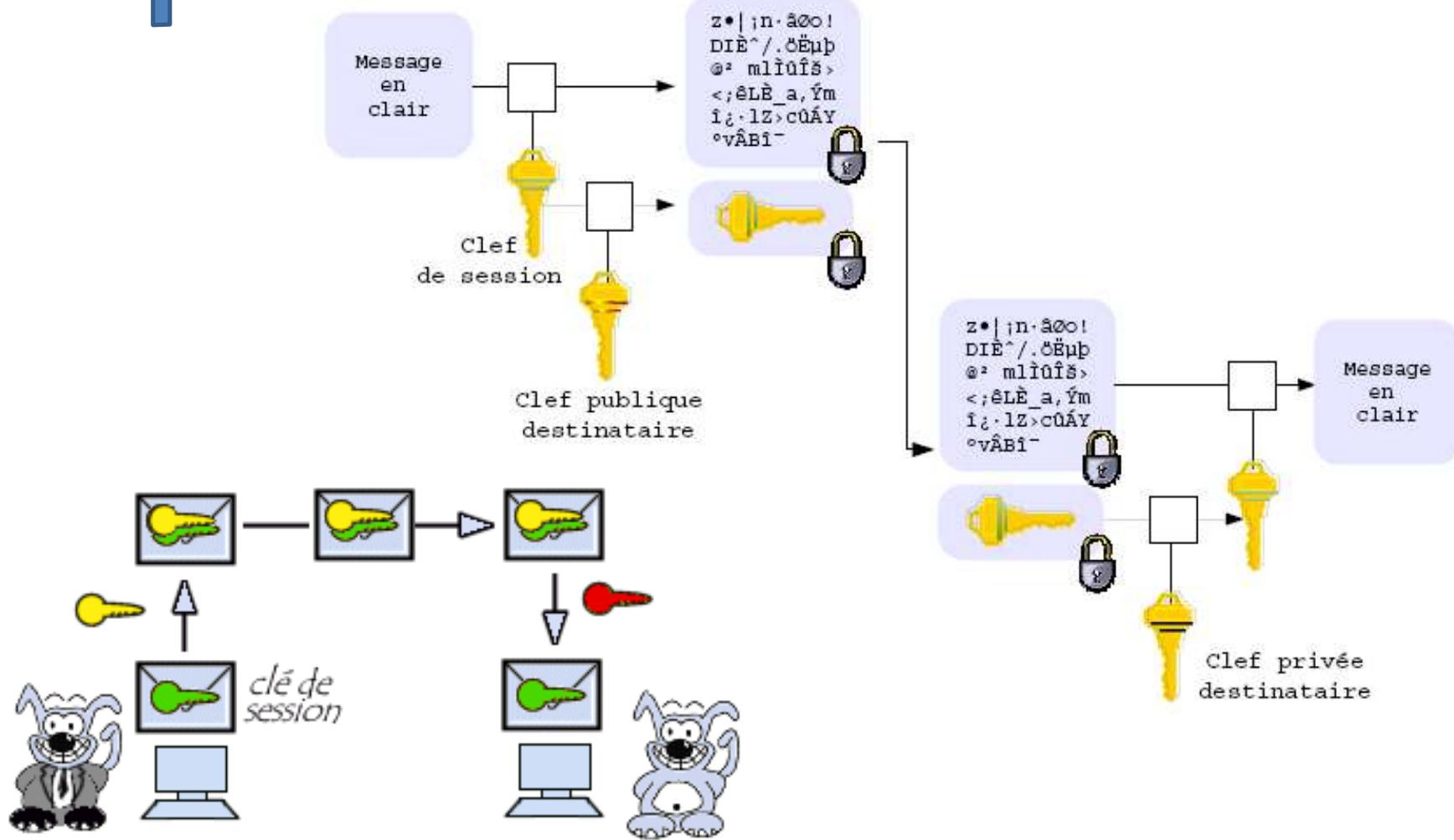
Scénario: authenticité de l'émetteur et non répudiation d'envoi



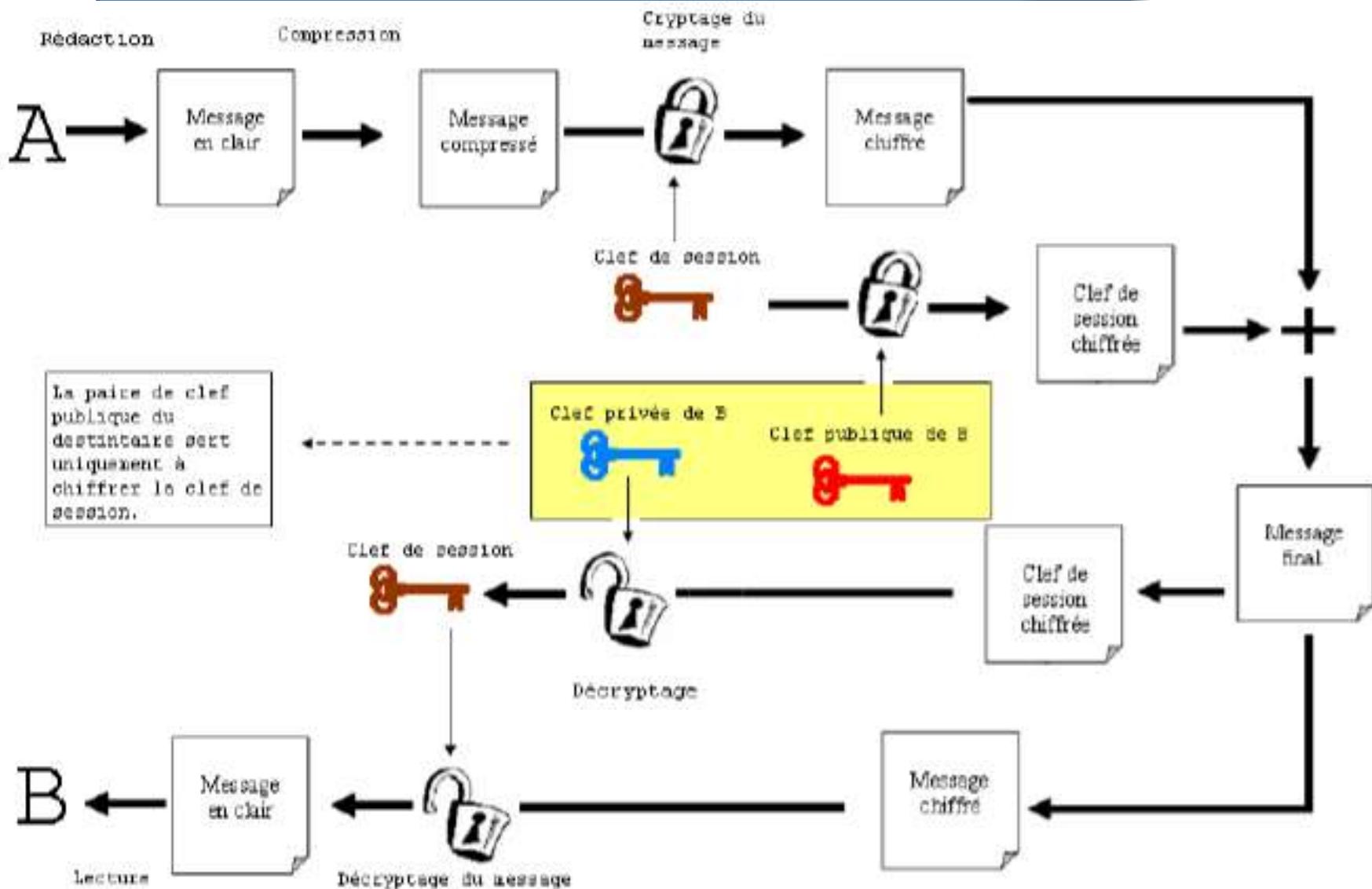
Scénario d'utilisation : chiffrement mixte



Scénario d'utilisation : distribution des clés de sessions (clés symétriques)



Scénario d'utilisation : distribution des clés de sessions (clés symétriques)



RSA: introduction

- RSA: Rivest, Shamir et Adleman du MIT en 1977
- Système le + connu et le + largement répandu
- 100 à 1000 fois plus lent que DES
- basé sur l'élévation à une puissance dans un champ fini sur des nombres entiers modulo un nombre premier
- Le nombre d'exponentiation prend environ $O((\log n)^3)$ opérations → facile
- emploie de grands nombres entiers (par exemple 1024 bits)
- sécurité due au coût de factoriser de grands nombres
- Le nombre de factorisation prend environ $O(e \log n \log \log n)$ opérations → difficile
- Usage : confidentialité, authentification

RSA: principe

- Choisir p et q , deux nombres premiers distincts et de tailles à peu près égales
- Noter le « module de chiffrement » $n=pq$
La longueur de n devrait être entre 1024 et 2048 bits
- Calculer l'indicatrice d'Euler (Ordre du groupe multiplicatif) :
$$\varphi(n)=(p-1)(q-1)$$
- Choisir e , un entier premier avec $\varphi(n)$, appelé « exposant de chiffrement » ($\text{pgcd}(e, \varphi(n))=1$).
- Comme e est premier avec $\varphi(n)$, il est, d'après le théorème de Bachet-Bézout, inversible mod $\varphi(n)$, c'est-à-dire qu'il existe un entier d tel que $ed \equiv 1 \pmod{\varphi(n)}$. d est l'exposant de déchiffrement, $d=e^{-1} \pmod{\varphi(n)}$
- Le couple (e, n) est appelé clé publique, $K_{\text{pu}}=\{e, n\}$
- Le couple (d, n) est appelé clé privée, $K_{\text{pr}}=\{d, n\}$

Le chiffrement RSA

- Chiffrement : $\text{enc}(X) = (X^e) \bmod n$
- Déchiffrement: $\text{dec}(X) = (X^d) \bmod n$
- Théorème d'Euler

$$\forall x \in \mathbb{Z}_n^*, x^{\varphi(n)} = 1 \bmod n$$

- Théorème de Bezout

$$ed + u\varphi(n) = 1$$

- Conséquence du théorème d'Euler

$$D(E(m)) = (m^e)^d = m^{ed} = m^{1-u\varphi(n)} = m \bmod n$$

RSA – Exemple

- Bob choisi $p=61$ et $q=53$
 - $n=pq=3233$
 - $\varphi(n)=60 \times 52 = 3120$
- Bob choisi e tel que $\text{pgcd}(\varphi(n), e) = 1$. On suppose que $e=17$
- $d=e^{-1} \bmod \varphi(n) = 2753 \bmod 3120$ (clé privée)
- Bob publie n et e (clé publique)
- Chiffrement: $\text{enc}(X) = (X^e) \bmod n = (X^{17}) \bmod 3233$
- Déchiffrement: $\text{dec}(X) = (X^d) \bmod n = (X^{2753}) \bmod 3233$
- $\text{enc}(123) = (123^{17}) \bmod 3233$
 $= 337587917446653715596592958817679803 \bmod 3233$
 $= 855$

RSA – Exemple

$\text{dec}(855) = (855^{2753}) \bmod 3233 = 5043288895841606873442289912739446663145387836003550$
931555496756450105562861208255997874424542811005438349865428933638493024645144
150785172091796654782635307099638035387326500896686074771829745822950342950407
903581845940956377938586598936883808360284013250976862076697739667533250542826
093475735137988063256482639334453092594385562429233017519771900169249169128091
505960191787601713497254392792156967017899021343071464689712796102771813783945
869677289869342365240311693217089269617643726521315665833158712459759803042503
144006837883246101784830717585474547252069688925995892544366701432205469543174
00228550092386369424448559733306305160738530286321930291350374547194675777671
357954965202919790505781532871558392070303159585937493663283548602090830635507
044556588963193180119341220178269233441013301164806963340240750469525886698765
866900622402410208846650753026395387052663193358473481094876156227126037327597
360375237388364148088948438096157757045380081079469800667348777958837582899851
327930703533551275090439948178979054899338121732945853544741326805698108726334
828546381688504882434658897839333466254454006619645218766694795528023088412465
948239275105770491133290256843065052292561427303898320890070515110552506189941
712317779515797942971179547529630183784386291397787766129820738907279676720235
011399271581964273076407418989190486860748124549315795374377124416014387650691
458681964022760277668695309039513149683190973245054523459447725658788769269335
3918692354818518542420923064996406822184490119135710885424428521120776

RSA – Exemple

37122383110545543126530739407592789082260604317113339575226603445164525976316184
27745904320191345289329932161307440532227470572894812143586831978415597276496357
09090121513130415756920979851832104115596935784883366531595132734467524394087576
97778908490126915322842080949630792972471304422194243906590308142893930291584830
87368745078977086921845296741146321155667865528338164806795455941891006950919658
99085456798072392370846302553545686919235546299571573587906227458619572172111078
82865756385970941907763205097832395713464119025004702084856040821750949107716553
11765297473803176765820587673140288910328834318508844721164427193903740413155649
86995913736516210845113740224335185995766577539693628125425390068552624545614192
58809437402128886669744109721845342218171980899119537075455420339119645393664617
92968165342652234639936742330970183533904623677693670380534264482173582384219251
59043814852473889686424437031866541996153779139696490030395876065491524494504360
01359392771339521012519285720925978875116019596296156902711643189463734265002363
10045557180036935860552649100009072451837866895644171649072783562810097085452413
54696608448116133878065485451517616730860510806578293652410872326366722805400387
94108643482267500907782651210137281958316531396983090887317417474535988684298559
80718519221597004650810606844559536480892249440542766329674592308898484868435865
47985051154284401646235269693179937784430217857019197098751629654665130278009966
58005217820813931723237901323249468260920081998103768484716787498919369499791482
4716345060937125654122501953795166897601855087599313367797793527822273233375295
802631226653589482055665152894663690320832876804323906115493509545909340

RSA – Exemple

6676402258670848337605369986794102620470905715674470565311242862907354888492989
98356099963609214112849774586146960402870296707014781794902482829074841600836804
58666855076046192252094349804715745268818131850859150194852763596503458153641656
54931601306133040743445796510838030406224027889804282518909471629226689801668448
09636451980905109057965130757037924595807447975237126676101147387874214414915481
35917439279949695641565386688389171544630561180536972834347021920634899953191764
01611039249043917980339897549176539592360851180765318470647331801578207412764787
59273908749295571685366518591266637383123594589126787095838000224515094244575648
74484086877530845395521730636693891702394037184780362774643171470855830491959895
14677629439214310024561306111429937000557751339717282549110056008940898419671319
70911816554290876109008324997831338240786961578492341986299168008677495934077593
06602207814943807854996798945399364063685722697422361858411425048372451244655802
70859179795591086523099756519838277952945756996574245578688383544423685722368139
90212613637440821314784832035636156113462870198514239018429097416386202320510397
12184983355286308685184282634615027441873586395040422815123995059959836537922272
85847422071677836679451343638070865797742198535953931662799887897216959634553463
36497949221130176613162074772661131070123214037138822702217232330854726795330150
79980622538354589480248200431447261915961905260340690619309392907241028494870016
71729695177034679099794409750637649296356755580071162182772760318292179035029048
60909762662853966270243925368902563371014716832740450458306022867631421581599007
916426277000546123229192192997169907690169025946468104141214204472402661

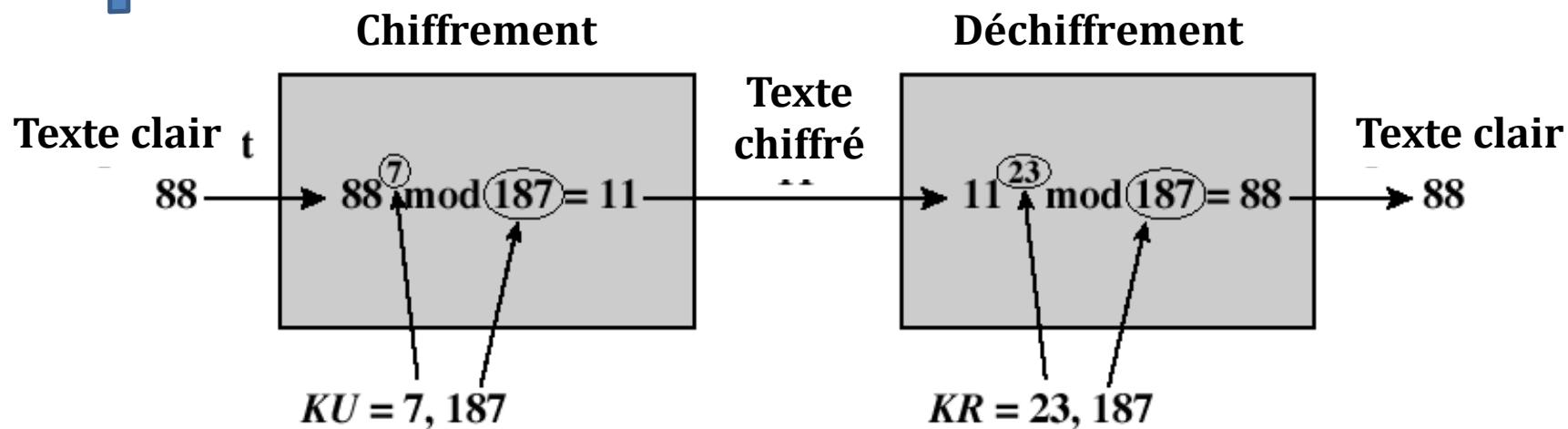
RSA – Exemple

658275680524166861473393322659591270064563044741608529167218700704514464979322666
873214634674904118588676083684030619069578699009652139067520501974407677651043885
151941619318479919134924388152822038464729269446084915299958818598855195149066307
311777238132267516945882593638786107243025659809149010327838482140113655678493410
243151248286452917031410040012016364829985325166349056053794585089424403855252455
477792240104614890752745163425139921637383568141490479320374263373019878254056996
191635201938969825447863130977374915447842763453259399874170013816319811664537720
894400285485000269685982644562183794116702151847721909339232185087775790959332676
311413129619398495926138987901669710881027663862316769405729593253807864344410051
213802508179762272379721035219677326844194648616402961059899027710532570457016332
613431076417700043237152474626393990118997278453629493036369149008810605312316300
090101508393318801166821516389310466665951378274989237455605110040164777168227162
672707837012242465512648784549235041852167426383189733332434674449039780017846897
264054621480241241258338435017048853206014756878623180940900126324196909225202267
988011340807301221626440413388739260052309607238615855496515800103474611979213076
722454380367188325370860671331132581992279755227718486484753261243028041779430909
389923709380536520464625514726788496152777327411926570911661358008414542148768731
039444105479639308530896880365608504772144592172500126500717068969428154627563704
588389042191773981906487319080148287390581594622278672774186101110276324797290412
221199411738820452633570175909067862815928151998221457652796853892517218720090070
3891385628400073322585075904853480465645434983707325

RSA – Exemple

935891427854318266587294608072389652291599021738887957736477387265746104008225511
241827200961681888284938946788104688473126554172620978905678458109651797530087306
315464903021121335281808476122990409576427857316364124880930949770739567588422963
171158464569842024551090298823985179536841258914463527918973076838340736961314097
452298563866827269104335751767712889452788136862396506665408989439495161912002160
777898876864736481837825324846699168307281220310791935646668401591485826999933744
276772522754038533221968522985908515481104022965791633825738551331482345959163328
144581984361459630602499361753097925561238039014690665163673718859582772525683119
989984646027216462797640770570748164064507697798699551061800464719378082232501489
340785113783325107375382340346626955329260881384389578409980417041041777608463062
862610614059615207066695243018438575031762939543026312673774069364047058960834626
0188591118436753252984588804084971092299919565539701911191918832730860376677533
960772245563211350657219106758751186812786344197572392195263333856538388240057190
102564949233944519659592039923922174002472341471909709645621082995477461932289811
812860555658809385189881181290561427408580916876571191122476328865871275538928438
126611991937924624112632990739867854558756652453056197509891145781147357712836075
540017742686609650933051721027230666357394623341363804591423775996522030941855888
003949675582971125836162189014035954234930424749053693992776114261796407100127643
280428706083531594582305946326827861270203356980346143245697021484375 mod 3233 =

Exemple de chiffrement RSA



$$p = 17, \quad q = 11, \quad n = p \times q = 187$$

$$\varphi(n) = 16 \times 10 = 160,$$

Choisir $e = 7$,

$$d \cdot e = 1 \pmod{\varphi(n)} \rightarrow d = 23$$

Exemple 2 de chiffrement RSA

- Saddam souhaiterait envoyer le message suivant à George : « Kisses from Iraq ». Malheureusement, Vladimir les espionne, et pourrait intercepter ce message. Nos deux compères vont donc chiffrer leurs échanges avec la méthode RSA.
- Étape 1
 - George a choisi $p = 37$ et $q = 43$. Il en déduit $n = 37 \times 43 = 1591$, et $\varphi(n) = 36 \times 42 = 1512$.
 - Il choisit ensuite $e = 19$, qui est premier avec 1512. L'inverse de 19 modulo 1512 est ?

Exemple 2 de chiffrement RSA

- Saddam souhaiterait envoyer le message suivant à George : « Kisses from Iraq ». Malheureusement, Vladimir les espionne, et pourrait intercepter ce message. Nos deux compères vont donc chiffrer leurs échanges avec la méthode RSA.
- Étape 1
 - George a choisi $p = 37$ et $q = 43$. Il en déduit $n = 37 \times 43 = 1591$, et $\varphi(n) = 36 \times 42 = 1512$.
 - Il choisit ensuite $e = 19$, qui est premier avec 1512. L'inverse de 19 modulo 1512 est $d=955$.
 - George peut donc maintenant publier sa clé publique, par exemple sur son site internet : $e = 19$ et $n = 1591$

Exemple 2 de chiffrement RSA

- Étape 2
 - Saddam va utiliser la clé pour chiffrer son message. Comme Saddam veut envoyer le message sous forme d'un fichier informatique, le mieux est d'utiliser le code ASCII.

K	i	s	s	e	s		f	r	o	m		I	r	a	q
75	105	115	115	101	115	32	102	114	111	109	32	43	114	97	113

- Étape 3
 - Il suffit à Saddam de coder chaque nombre comme expliqué.

Exemple 2 de chiffrement RSA

- Étape 2
 - Saddam va utiliser la clé pour chiffrer son message. Comme Saddam veut envoyer le message sous forme d'un fichier informatique, le mieux est d'utiliser le code ASCII.

K	i	s	s	e	s		f	r	o	m		I	r	a	q
75	105	115	115	101	115	32	102	114	111	109	32	43	114	97	113

- Étape 3
 - Il suffit à Saddam de coder chaque nombre comme expliqué. Par exemple $75^{19} \text{ mod } 1591 = 371$

Exemple 2 de chiffrement RSA

- Étape 2
 - Saddam va utiliser la clé pour chiffrer son message. Comme Saddam veut envoyer le message sous forme d'un fichier informatique, le mieux est d'utiliser le code ASCII.

K	i	s	s	e	s		f	r	o	m			I	r	a	q
75	105	115	115	101	115	32	102	114	111	109	32	43	114	97	113	

- Étape 3
 - Il suffit à Saddam de coder chaque nombre comme expliqué. Par exemple $75^{19} \text{ mod } 1591 = 371$.
 - On utilise la méthode « modulo exponentiation »

$$\begin{aligned} m &= 75^{19} \text{ mod } 1591 & &= 75 * (5625 \text{ mod } 1591)^9 \text{ mod } 1591 \\ &= 75 * 75^{18} \text{ mod } 1591 & &= 75 * (852)^9 \text{ mod } 1591 \\ &= 75 * (75^2)^9 \text{ mod } 1591 & &= 75 * 852 * (859^2)^4 \text{ mod } 1591 \\ &= 75 * (5625)^9 \text{ mod } 1591 & &= 75 * 852 \text{ mod } 1591 = 371 \end{aligned}$$

Exemple 2 de chiffrement RSA

- Étape 3

75	105	115	115	101	115	32	102	114	111	109	32	43	114	97	113
371	1338	1410	1410	1174	1410	930	1397	632	703	483	930	1405	632	532	1441



371	1338	1410	1410	1174	1410	930	1397	632	703	483	930	1405	632	532	1441
-----	------	------	------	------	------	-----	------	-----	-----	-----	-----	------	-----	-----	------

- Étape 4

— Saddam envoie cette suite de nombres à George, qui va la déchiffrer avec sa clé d . Il va pouvoir retrouver le message original : $371^{955} \text{ mod } 1591 = 75$

371	1338	1410	1410	1174	1410	930	1397	632	703	483	930	1405	632	532	1441
75	105	115	115	101	115	32	102	114	111	109	32	43	114	97	113
K	i	s	s	e	s		f	r	o	m		I	r	a	q

Faiblesses de l'algorithme RSA

- Pour casser complètement le système, il faut retrouver l'exposant d ou mieux p et q **par factorisation**.
- On peut aussi factoriser n . En effet, n est connu et si on le factorise, on obtient p et q puis $\phi(n)$ et connaissant $\phi(n)$ et d , on obtient e .
- Mais, la factorisation de n n'est pas une chose facile. La factorisation de grands nombres suffit, à elle seule, à dissuader de nombreuses tentatives.
- En pratique, il y a deux difficultés pour implémenter RSA:
 - La première est la génération de grands nombres premiers (p et q)/
 - la seconde est l'élévation de nombre à des puissances très grandes. Un standard de RSA est PKCS 1.

Faiblesses de l'algorithme RSA

- Pour déchiffrer un message, il faut retrouver m à partir de m^e : extraction de racine e -ièmes



Si on connaît la factorisation, on casse RSA

- On retrouve toutes les données secrètes
- RSA se réduit à la factorisation !



Si on connaît les données secrètes, il faut calculer les racines e -ièmes : on ne sait pas faire

Limites du RSA

- p et q doivent être très grands. On estime qu'il faut moins d'une seconde pour casser un code à base de nombre de 32 bits.
- Il faut utiliser des clés à 128 bits minimum
- Il faut chiffrer le message par bloc de plusieurs caractères pour éviter de forcer le message par analyse fréquentielle : déchiffrer un message en se basant sur la fréquence d'apparition des lettres dans une langue. Par exemple le *e* en français est très souvent utilisé.
- Cas d'un modulo commun

Cas du modulo commun

- Supposons qu'Alice et Bob utilisent le même modulo.
 - Si un même message a été chiffré à l'intention d'Alice et Bob, il devient possible à tout le monde de le déchiffrer
- Message m chiffré avec les exposants e_A et e_B supposés premiers entre eux
- Bezout :

$$u e_A + v e_B = 1$$

- Soient $c_A = m^{e_A}$ et $c_B = m^{e_B}$
- N'importe qui peut calculer

$$c_A^u \times c_B^v = m^{u e_A + v e_B} = m$$

Attaques du RSA

- Attaques mathématiques : Factorisation d'un Grand nombre
 - Idée: retrouver p et q en factoriser le modulo N
 - Méthode Fermat
 - Méthode Euler
 - Méthode Pollard's Rho
- Attaque d'implémentation : Obtention de données physiques lors de l'implémentation d'un système cryptographie (analyse fréquentielle)
 - Timing Attack

Méthode de Fermat

- Le nombre n peut s'écrire sous la forme

$$n = \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2$$

- Choisir k , le plus petit entier tel que

$$k^2 > n$$

- Augmenter k un par un jusqu'à ce qu'il existe un entier h

$$(k + g)^2 - n = h^2$$

- De même pour g

$$n = (k+g+h)(k+g-h)$$

Attaques mathématiques

- Durée des attaques en s

Number of bit (N)	Method Rho	Fermat
10	0	31
20	94	219
25	200	13375
25	198	3375
25	188	765
32	282	899515
40	8000	
40	8456	

Avantages et inconvénients

- Avantages
 - Pas besoin d'établir un canal sûr pour la transmission de la clé.
 - Plusieurs fonctions de sécurité: confidentialité, authentification, et non-répudiation
- Inconvénient
 - Généralement dix fois plus lent que le chiffrement symétrique.
 - Problème d'implémentation sur les équipements disposants de faible puissance de calcul (ex: cartes bancaire, stations mobiles, etc.)
 - Clés longues
 - Complexité algorithmique de la méthode (ex: réalisation des opérations modulo n)

→ Solution: Utilisation du chiffrement asymétrique pour l'échange des clés secrètes de session d'un algorithme symétrique à clés privées.

Outils de cryptographie et d'authentification

Généralités

Chiffrement symétrique

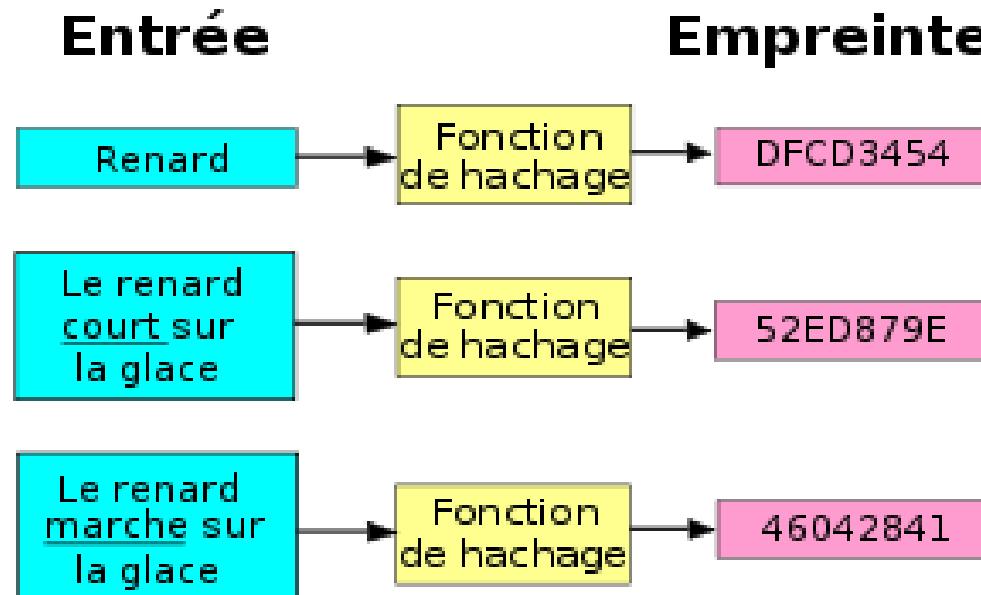
Chiffrement asymétrique

Intégrité d'un message

PKI

Fonctions de hashage

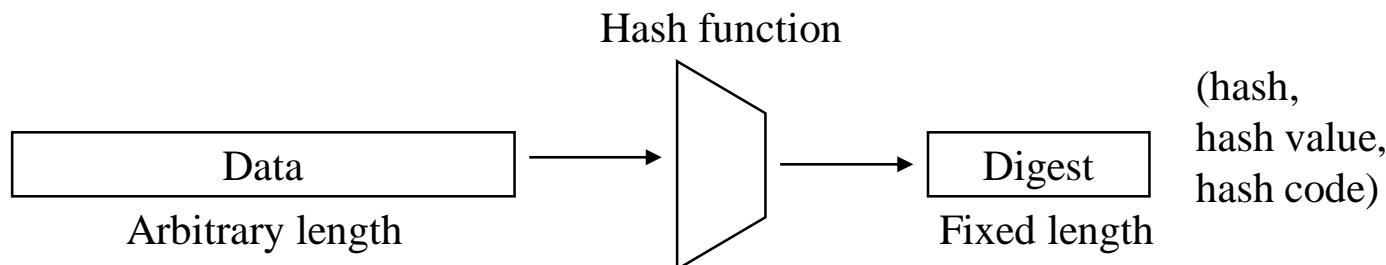
- **Une fonction de hachage** est une fonction qui, à partir d'une donnée fournie en entrée, calcule une *empreinte* servant à identifier rapidement, bien qu'incomplètement, la donnée initiale.



- Cette empreinte peut être aussi appelée selon le contexte *somme de contrôle*, *hash*, *résumé de message*, **condensé**, **condensat** ou encore *empreinte cryptographique*

Fonctions de hashage

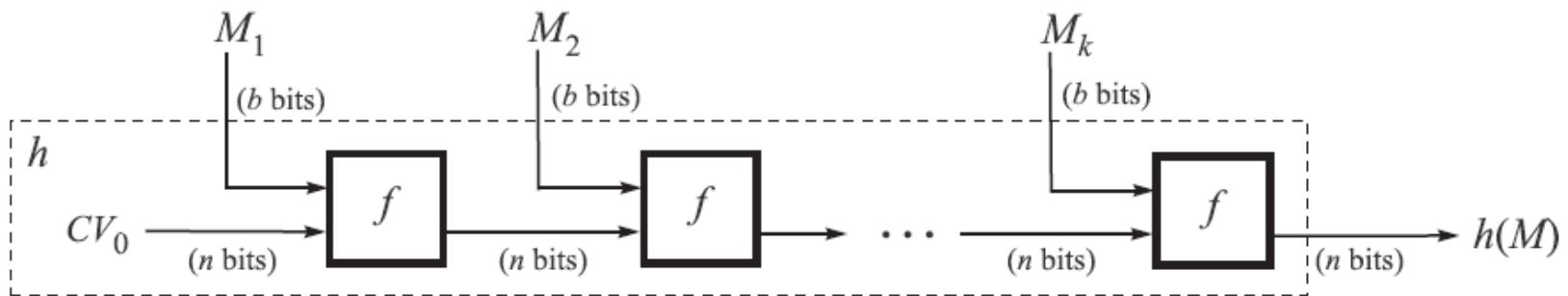
- Propriétés à respecter pour une fonction de hashage cryptographique:
 - **Collision resistance**: Il est très difficile de trouver deux valeurs x et x' telles que $h(x) = h(x')$.
 - **Weak collision resistance**: Etant donné une valeur x , il est très difficile de trouver une valeur x' qui vérifie $h(x') = h(x)$. Cette propriété est parfois appelée *second pre-image resistance*.
 - **One-way property**: Pour n'importe quelle valeur hash y , il est très difficile de trouver une valeur x telle que $h(x) = y$.



- Très difficile = *techniquement impossible en pratique* par toutes techniques algorithmiques et matérielles, en un temps raisonnable

Principe de base

$$M = M_1 \parallel M_2 \parallel \dots \parallel M_k$$



CV = chaining variable

f = algorithme de compression

k = Nombre de blocs

n = longueur du code hash

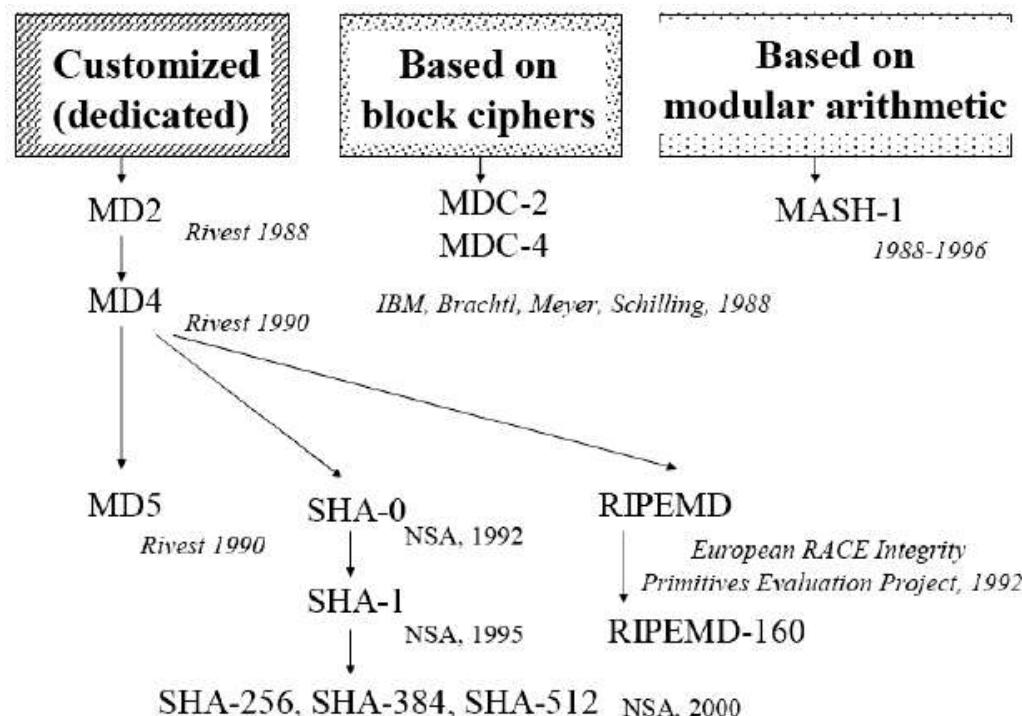
b = longueur du bloc

Fonctions de hashage

- Définition
 - Une fonction de hachage H est telle que
 - si $H(x) \neq H(y)$ alors $x \neq y$
 - si $H(x) = H(y)$ alors probablement $x = y$
- But: s'assurer que le message reçu correspond au message émis
 - "Empreinte digitale" électronique d'un document
 - Fonction à sens unique (avec l'empreinte on ne pas remonter au document)
 - Une petite modification du document initial entraîne une modification importante de la signature électronique

Fonctions de hashage

- Algorithmes
 - MD5 (Message Digest) : hachés sur 128 bits à partir de blocs de 512 bits
 - SHA1 (Secure Hash Algorithm 1) : hachés sur 160 bits à partir de blocs de 512 bits
 - SHA2 (Secure Hash Algorithm 2) : hachés sur 256, 384 ou 512 bits



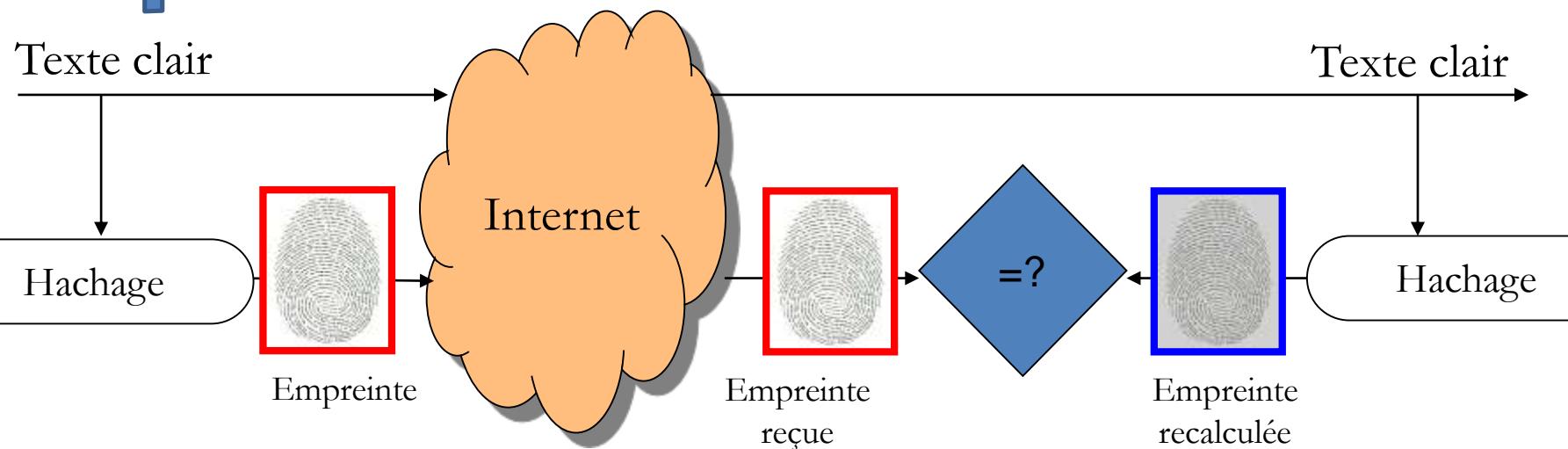
Caractéristiques du hashage

- Entrée: message M avec contenu et taille arbitraire.
- Sortie: message de taille fixe $C=H(M)$.
 - C est de taille fixe (16 ou 20 octets) : C est appelé **condensât**, ou **empreinte**, ou **fingerprint**, ou **digest**
- La fonction de hachage permet d'extraire une empreinte qui caractérise les données.
 - Une empreinte a toujours une taille fixe indépendamment de la taille des données.
- Irréversible:
 - Étant donné h , il est difficile de trouver x tel que: $h = H(x)$
 - Complexité de l'ordre de 2^n , n est le nombre de bits du *digest*.
- Calcul facile et rapide (plus rapide que le chiffrement symétrique).

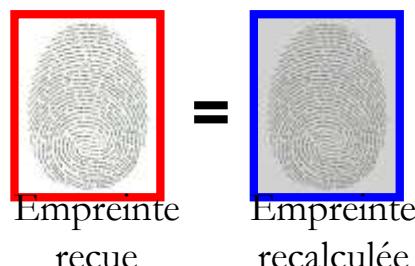
Caractéristiques du hashage

- Irréversible
 - Soit « y » le résultat de hachage, il est pratiquement infaisable de trouver « x » tel que $h(x)=y$.
- Résistance forte à la collision:
 - Soit « x » et « $y=h(x)$ », il est pratiquement infaisable de trouver « $x' \neq x$ » tel que $h(x')=h(x)$.
 - Il est pratiquement infaisable de trouver deux valeurs distinctes « x' » et « x » tel que $h(x')=h(x)$.

Principe du hashage

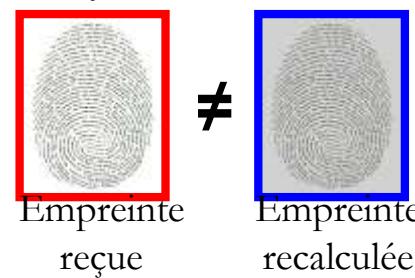


1)



Le texte reçu est intégrer

2)



Le texte reçu est altéré

Exemple de fonctionnement

Message initial

Mr Robert je vous envois ce message pour vous confirmer notre rendez-vous à 13H00

Empreinte du message initial

215e781c0c3f7d1353518bd5f649805b

Message reçu

Mr Robert je vous envois ce message pour vous confirmer notre rendez-vous à 9H00

Empreinte du message reçu

0601e38b93c1cc1c1a4b87dd8771b452

L'empreinte du message du message reçu est différente de celle du message initial

- Quelqu'un à modifier le message
- Il y a une erreur de communication

Exemple d'algorithme de hashage

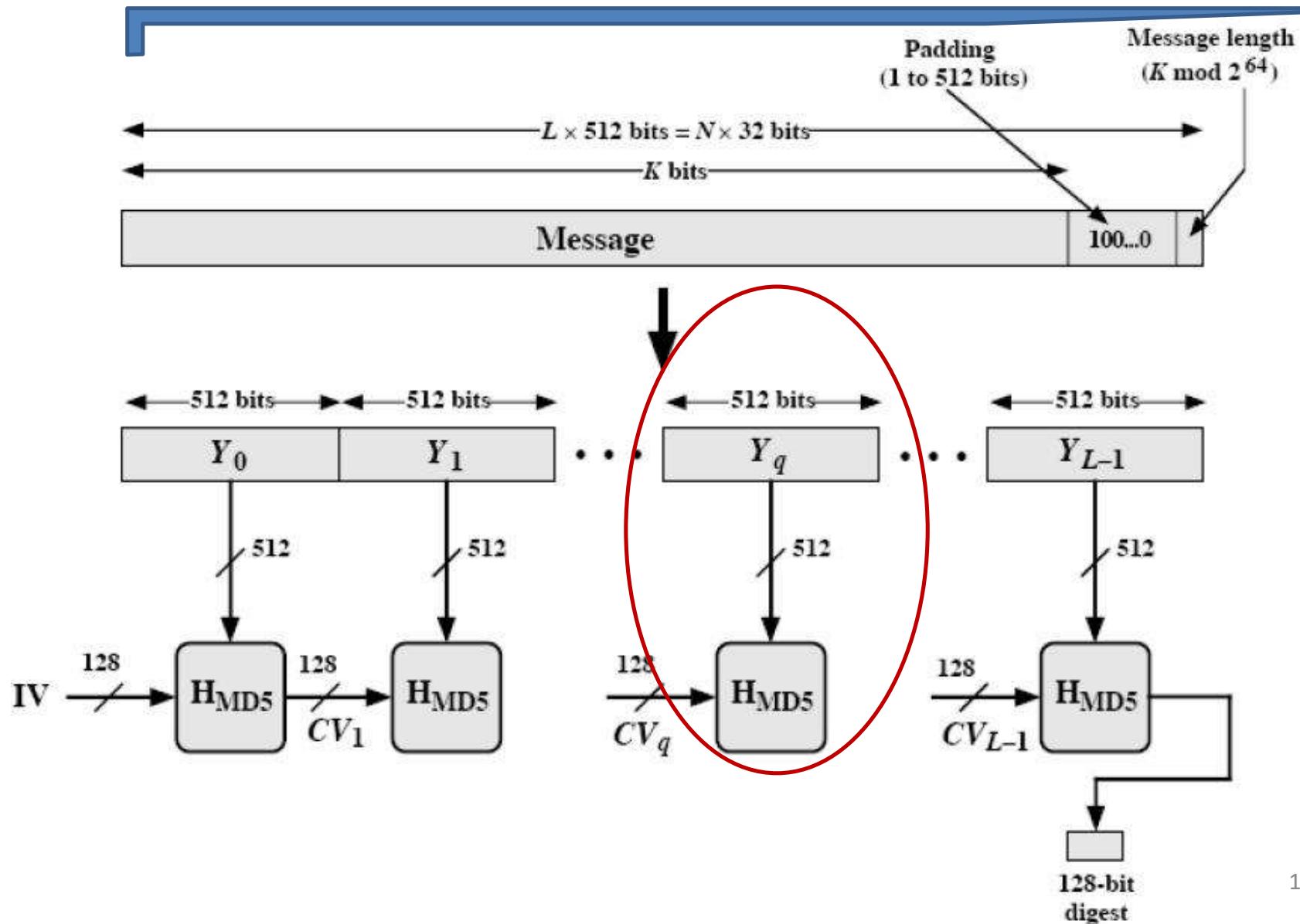
- **MD5** : *Message Digest 5*
 - Développé en 1991
 - Génère une empreinte de taille 128 bits en traitant les données d'entrée par blocs de 512 bits.
- **SHA-1** : *Secure Hash algorithm*
 - Génère une empreinte de taille 160 bits.
 - Plus fort que MD5.

	SHA-1	SHA-256	SHA-384	SHA-512
Message block size	512	512	1024	1024
Number of digest rounds	80	64	80	80

Exemple: MD5

- MD5: Message Digest 5, est une fonction de hachage cryptographique qui permet d'obtenir l'empreinte numérique d'un fichier / *message*
- Inventé par Ronald Rivest (le R de RSA)
- En 1996, une faille qualifiée de **grave** (*possibilité de créer des collisions à la demande*) est découverte et indique que MD5 devrait être mis de côté au profit de fonctions plus robustes comme SHA-1.
- MD5 est encore largement utilisé comme outil de vérification lors des téléchargements pour valider l'intégrité de la version téléchargée grâce à l'empreinte.
- MD5 peut aussi être utilisé pour calculer l'empreinte d'un mot de passe ; c'est le système employé dans GNU/Linux
 - plutôt que de stocker les mots de passe dans un fichier, ce sont leurs empreintes MD5 qui sont enregistrées; quelqu'un qui lirait ce fichier ne pourra pas découvrir les mots de passe.

MD5: Principe



MD5: Principe

- Les quatre fonctions non-linéaires disponibles sont :

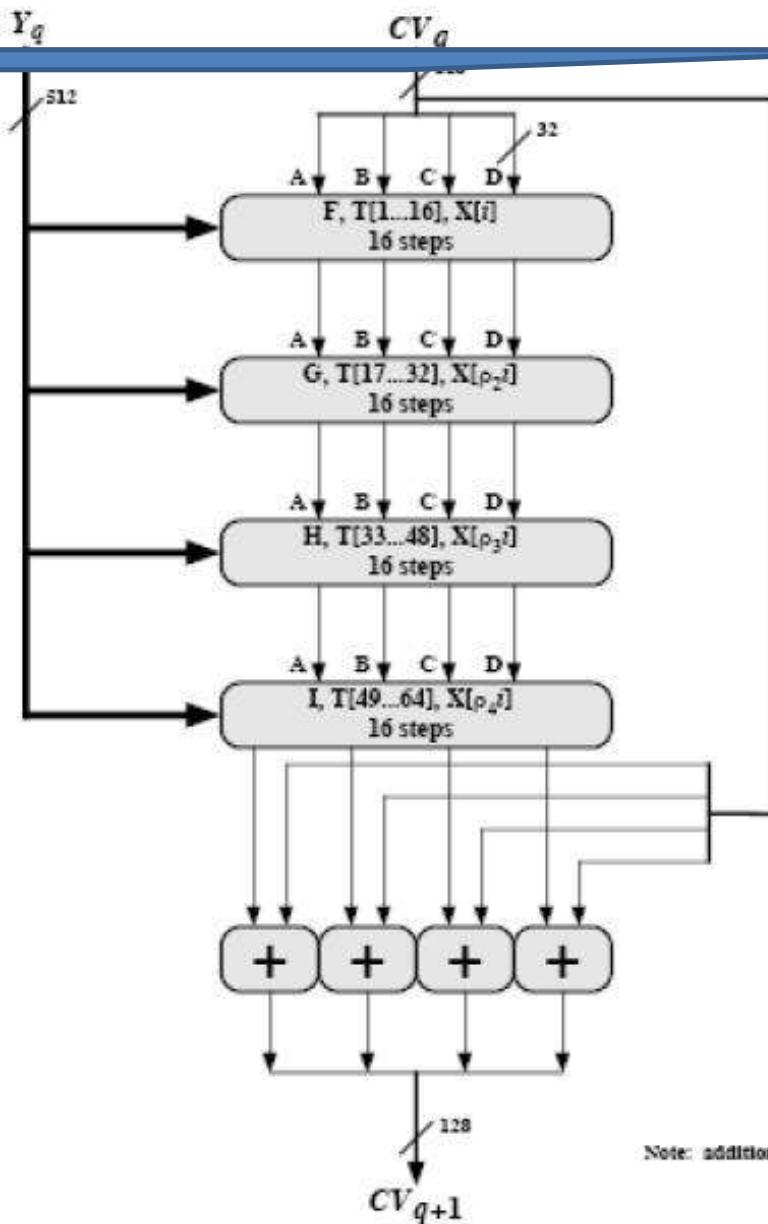
$$F(B, C, D) = (B \wedge C) \vee (\neg B \wedge D)$$

$$G(B, C, D) = (B \wedge D) \vee (C \wedge \neg D)$$

$$H(B, C, D) = B \oplus C \oplus D$$

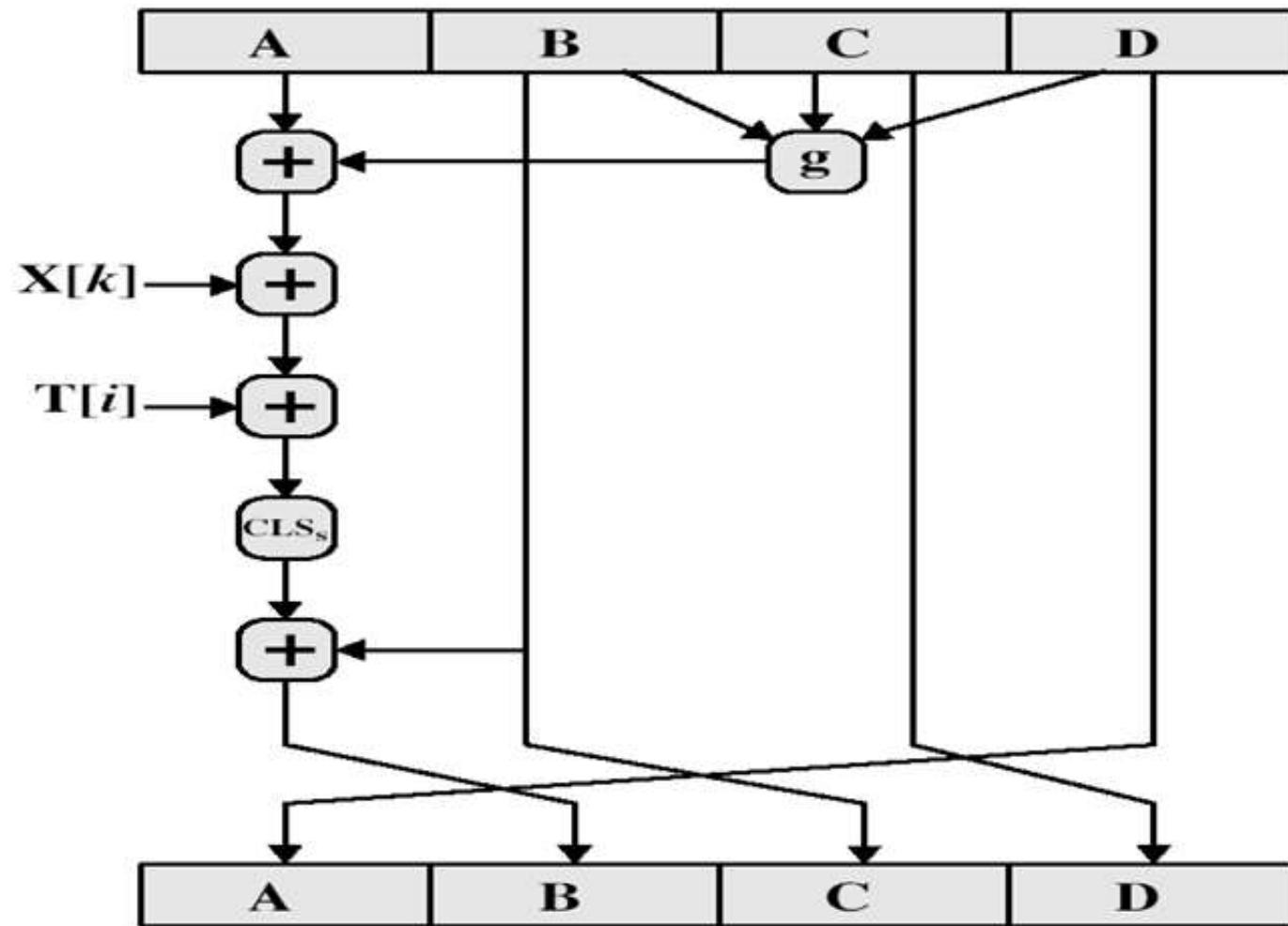
$$I(B, C, D) = C \oplus (B \vee \neg D)$$

- Notation:
 - cls_s est une rotation de s bits vers la gauche, s varie pour chaque opération.
 - [+] symbolise l'addition modulo 2^{32} .
 - $\oplus, \wedge, \vee, \neg$ symbolisent respectivement les opérations booléennes XOR, AND, OR et NOT.



Note: addition (+) is mod 2^{32}

MD5: Principe



L'algorithme MD5

- Étape 1 : Complétion
 - Le message est constitué de b bits $m_1 \dots m_b$
 - On complète le message par un 1, et suffisamment de 0 pour que le message étendu ait une longueur congruente à 448, modulo 512
 - Ajout à ce message de la valeur de b , codée en binaire sur 64 bits
 - On obtient donc un message dont la longueur totale est un multiple de 512 bits.
On va travailler itérativement sur chacun des blocs de 512 bits
- Étape 2 : Initialisation
 - On définit 4 buffers de 32 bits A,B,C et D, initialisés
 - A=01234567 ; B=89abcdef ; C=fedcba98 ; D=76543210
 - On définit aussi 4 fonctions F,G,H et I, qui prennent des arguments codés sur 32 bits, et renvoie une valeur sur 32 bits, les opérations se faisant bit à bit

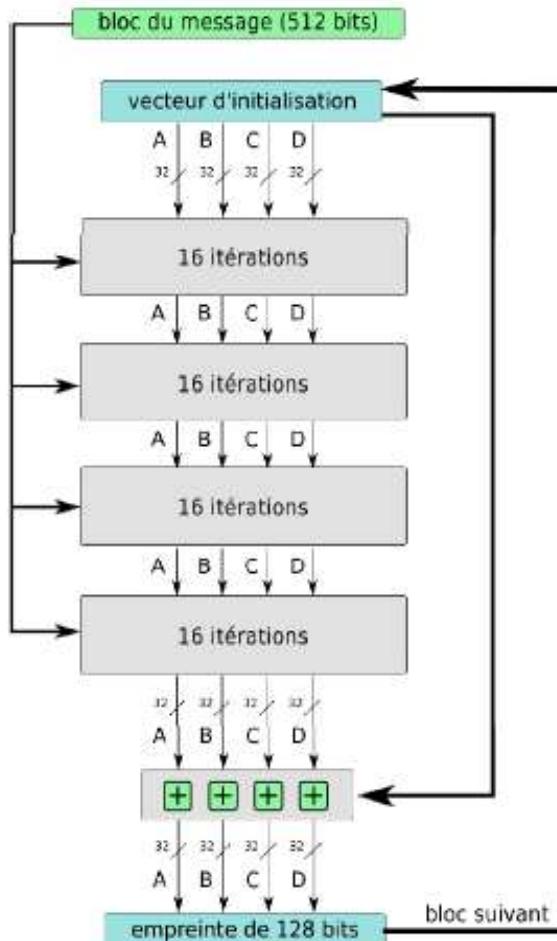
$$\begin{aligned} F(X,Y,Z) &= (X \text{ AND } Y) \text{ OR } (\text{not}(X) \text{ AND } Z) \\ G(X,Y,Z) &= (X \text{ AND } Z) \text{ OR } (Y \text{ AND } \text{not}(Z)) \end{aligned}$$

$$\begin{aligned} H(X,Y,Z) &= X \text{ xor } Y \text{ xor } Z \\ I(X,Y,Z) &= Y \text{ xor } (X \text{ OR } \text{not}(Z)) \end{aligned}$$

L'algorithme MD5

- Étape 3 : Calcul itératif : Pour chaque bloc de 512 bits du texte, on fait les opérations suivantes :
 - on sauvegarde les valeurs des registres dans A,B,C,D
 - on calcule de nouvelles valeurs pour A,B,C,D à partir de leurs anciennes valeurs, à partir des bits du bloc qu'on étudie, et à partir des 4 fonctions F,G,H,I
 - Au final on fait $A=A+A$, $B=B+B$, $C=C+C$, $D=D+D$.
- Étape 4 : Écriture du résumé
 - Le résumé sur 128 bits est obtenu en mettant bout à bout les 4 buffers A, B, C, D de 32 bits

L'algorithme MD5



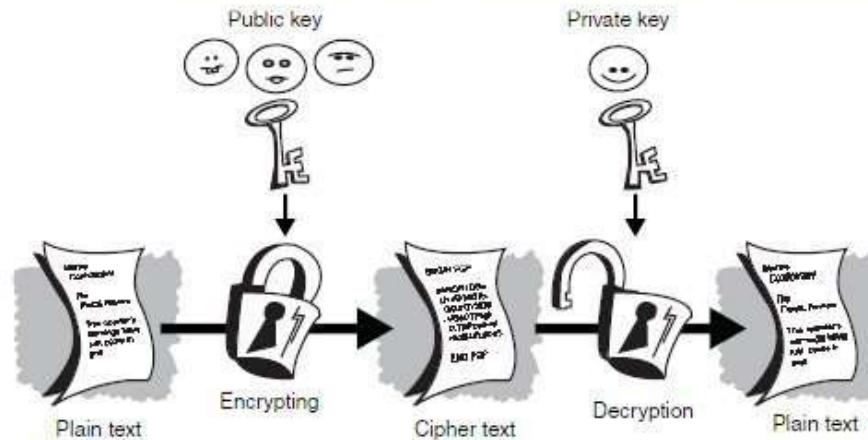
L'algorithme MD5 est composé de 64 étapes
: **4 fois 16 itérations**

Le résumé sur 128 bits est obtenu en
mettant bout à bout les 4 buffers A,B,C,D de
32 bits

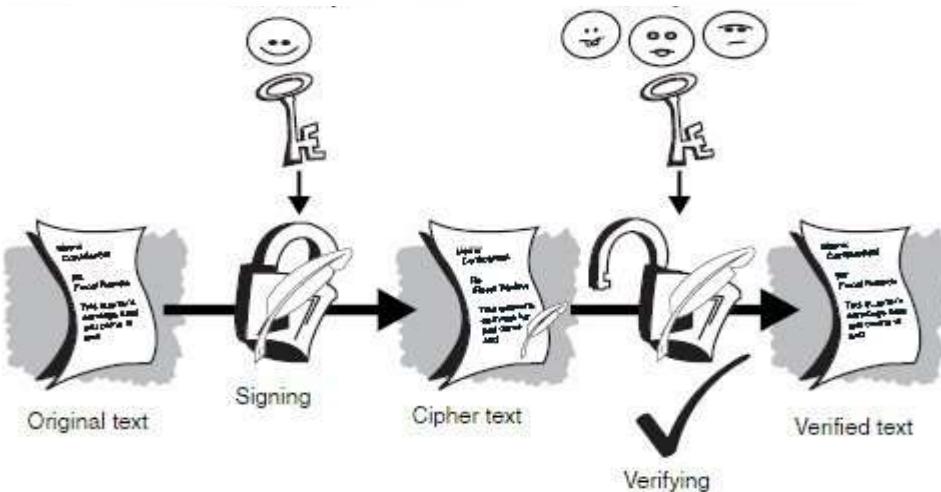
Le message est traité 4 fois à chaque fois le
message est décomposé en blocs de 32 bits
Il faut donc 16 itérations pour traiter le
message

Signatures digitales

- Chiffrement à clé publique

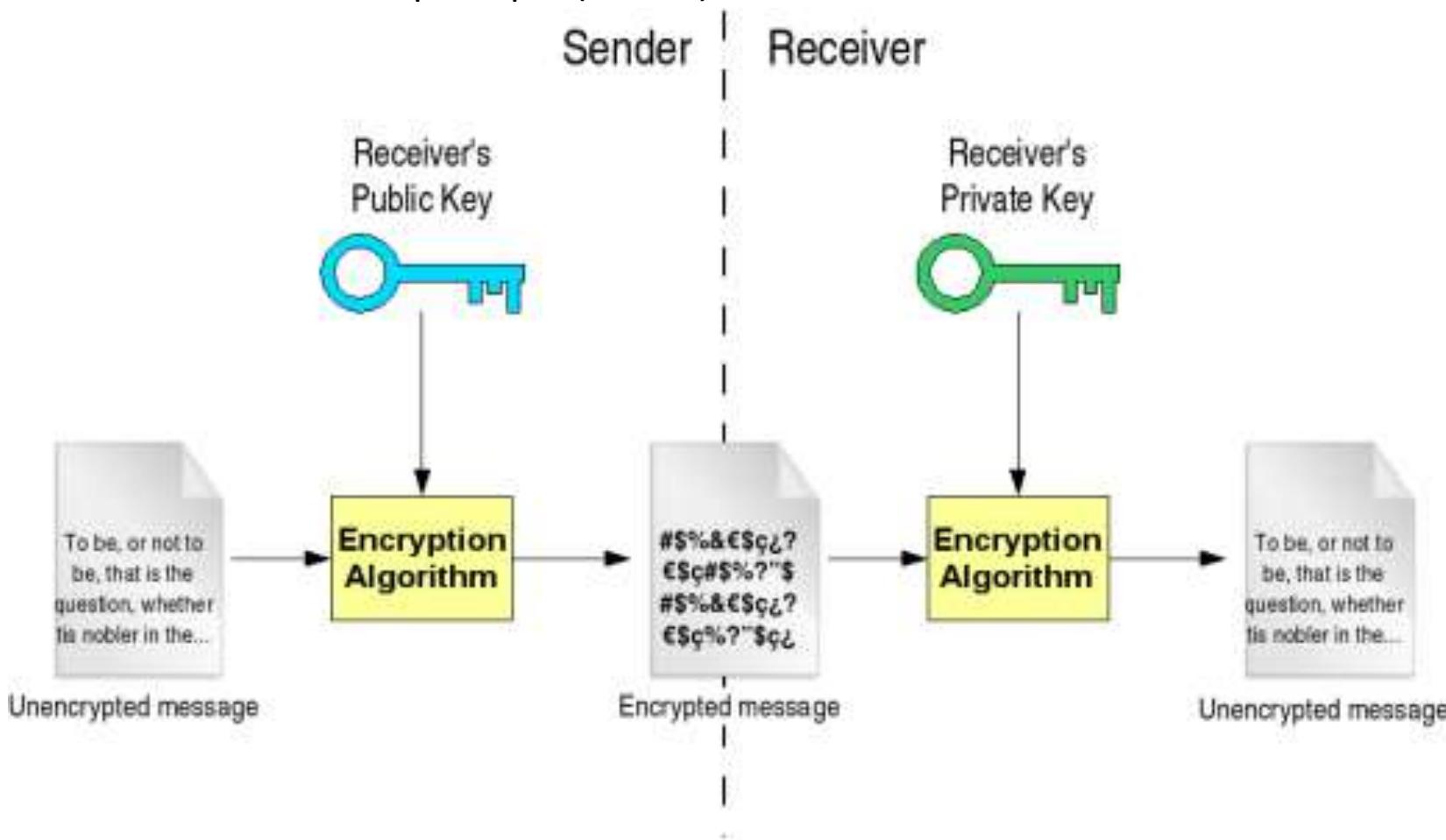


- Signature à clé publique



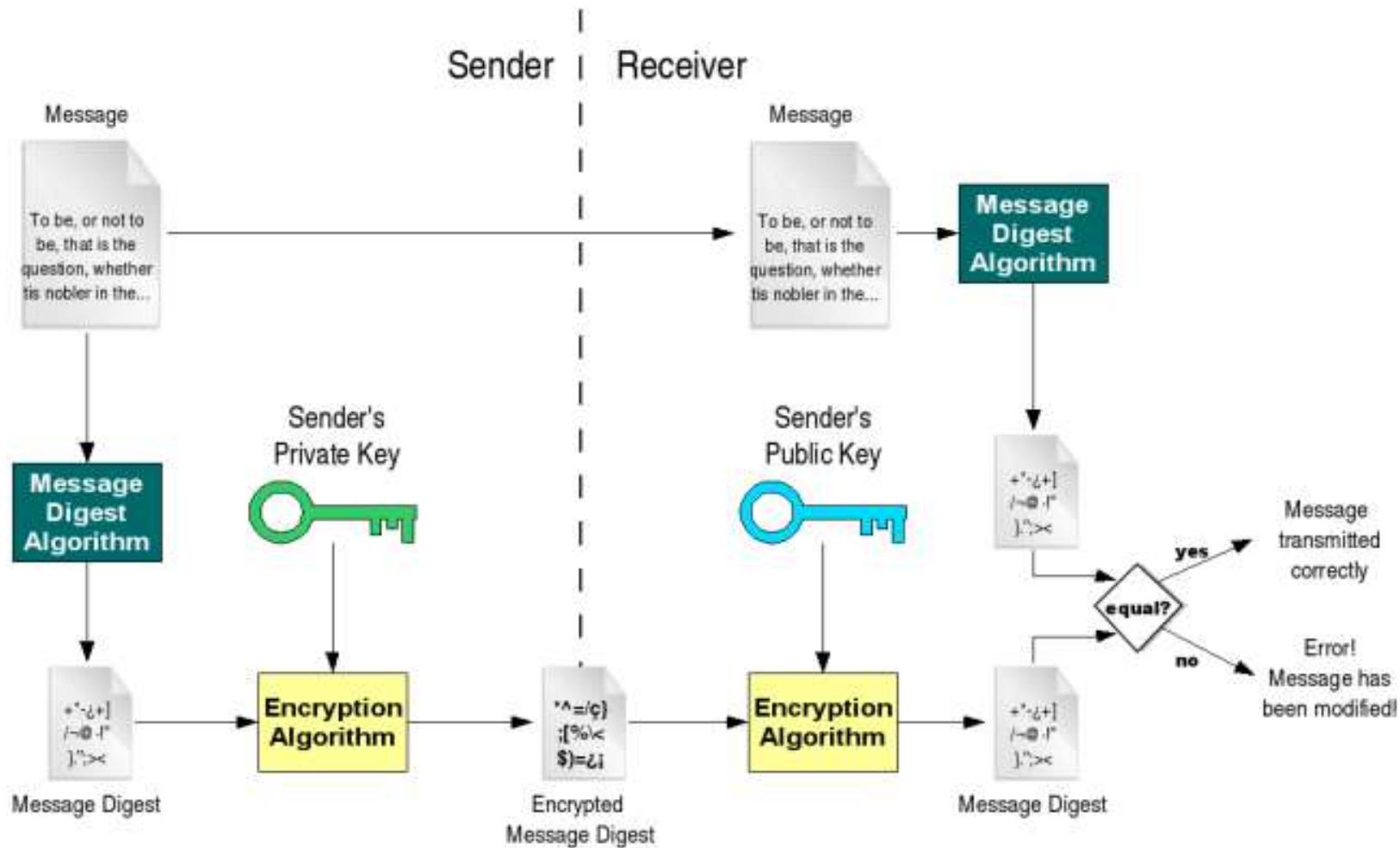
Signatures digitales

- Chiffrement à clé publique (détails)



Signatures digitales

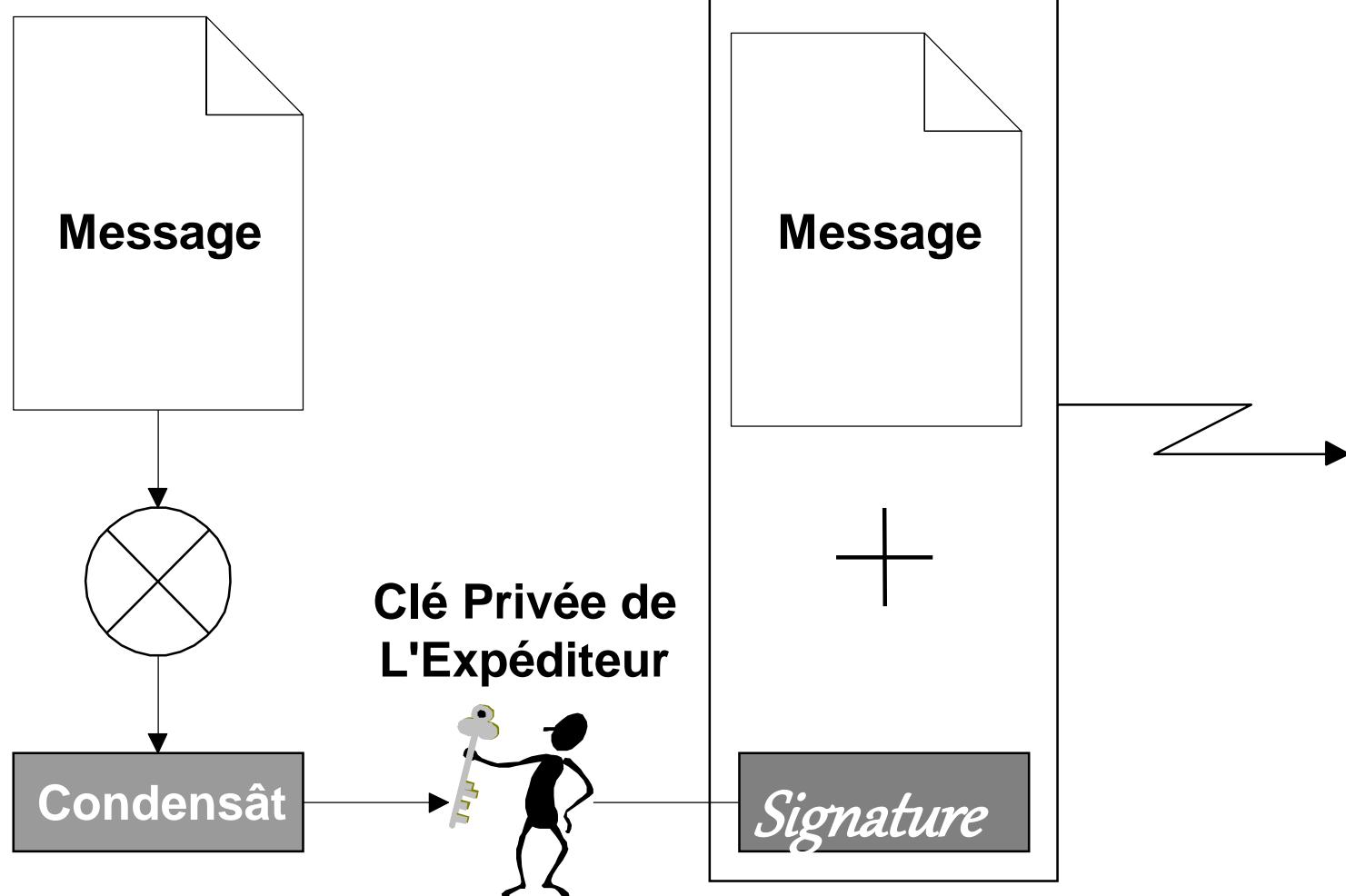
- Signature à clé publique (détails)



Certificats

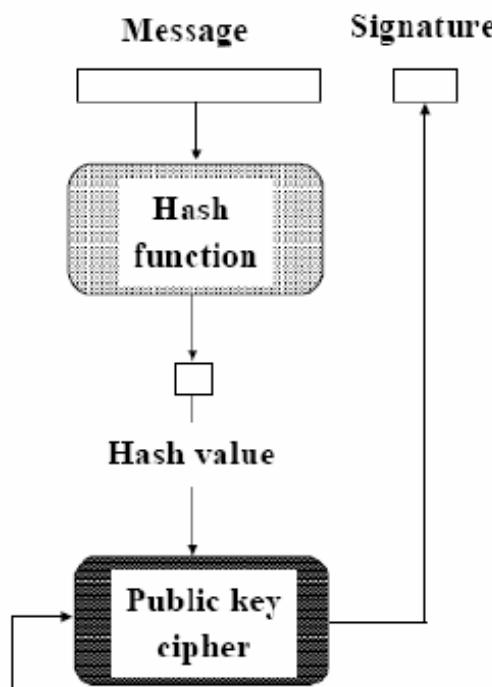
- Un **certificat électronique** est une carte d'identité numérique dont l'objet est d'identifier une entité physique ou non-physique.
- Le certificat numérique ou électronique est un lien entre l'entité physique et l'entité numérique (Virtuel).
- L'autorité de certification fait foi de tiers de confiance et atteste du lien entre l'identité physique et l'entité numérique.
- Le standard le plus utilisé pour la création des certificats numériques est le X.509.

Signature numérique

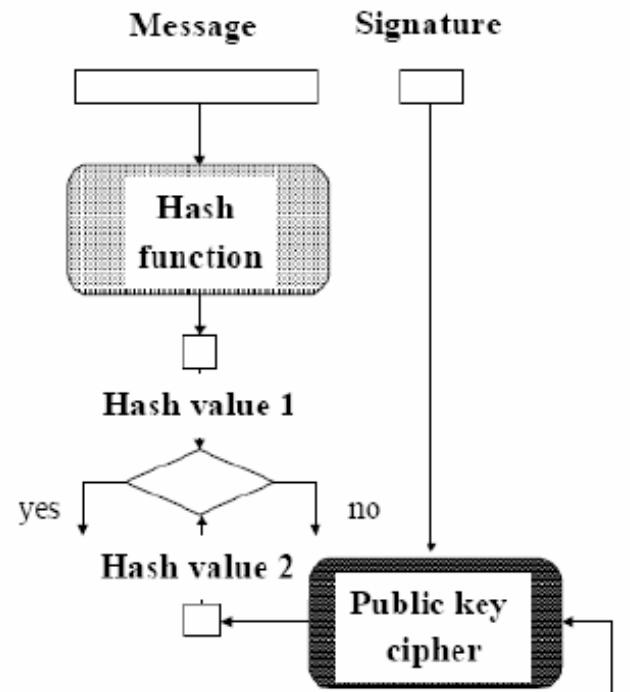


Signature numérique

Alice



Bob



Alice's private key

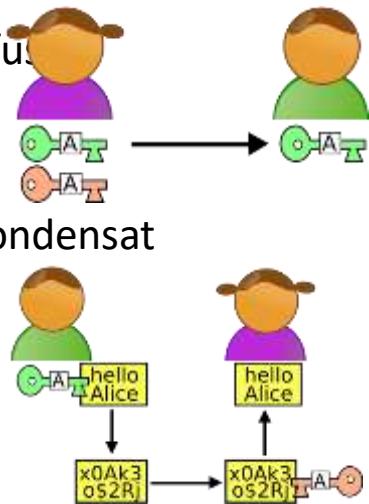
Alice's public key²⁸

Domaine d'utilisation de la signature

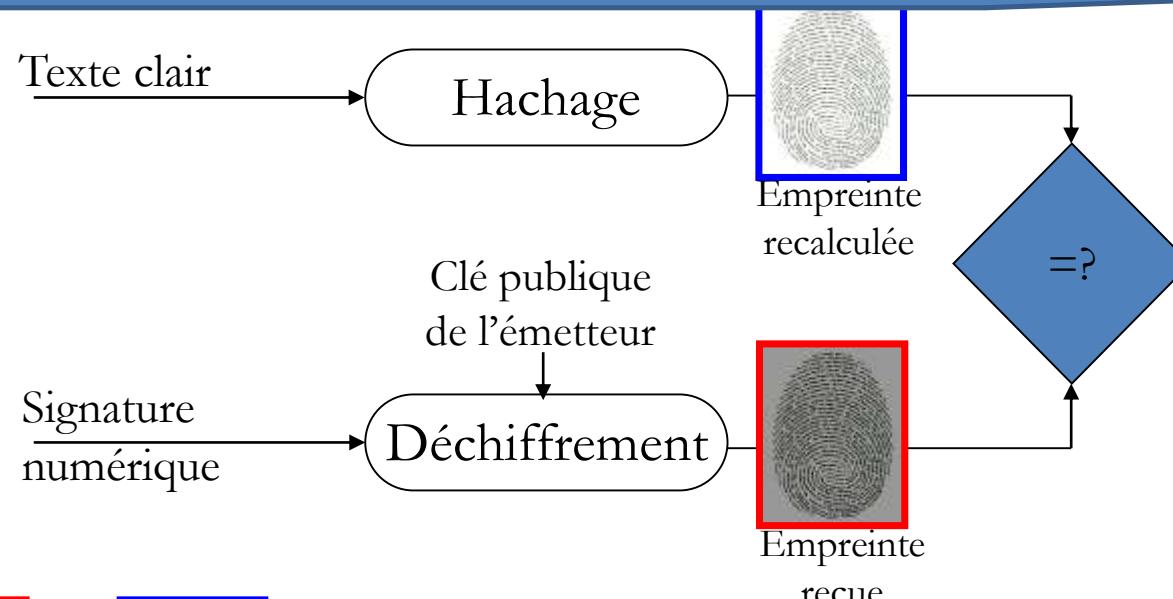
- La signature permet de mettre en œuvre les services:
 - Intégrité du message
 - Authentification
 - Non-répudiation
 - Génération d'une clé de chiffrement symétrique pour le service de Confidentialité

La signature pour l'authentification

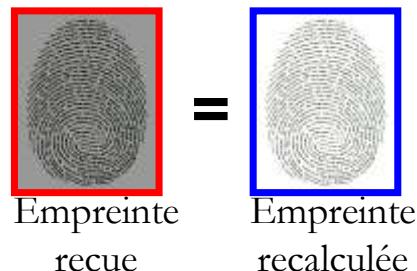
- Objectif : Bob souhaite envoyer des données chiffrées à Alice en lui garantissant qu'il en est l'expéditeur.
 - Bob crée une paire de clés asymétriques : il conserve la clé privée et diffuse librement la clé publique (notamment à Alice)
 - Alice crée une paire de clés asymétriques
 - Bob effectue un condensat de son message « en clair » puis chiffre ce condensat avec sa propre clé privée
 - Bob chiffre son message avec la clé publique d'Alice.
 - Bob envoie le message +condensat chiffrés
 - Alice reçoit le message +condensat chiffrés de Bob
 - Alice déchiffre le message avec sa propre clé privée. À ce stade le message est lisible mais elle ne peut pas être sûre que Bob en soit l'expéditeur.
 - Alice déchiffre le condensat avec *la clé publique de Bob*. Alice utilise la même fonction de hachage sur le texte *en clair* et compare avec le condensat déchiffré de Bob. Si les deux condensats correspondent, alors Alice peut avoir la certitude que Bob est l'expéditeur. Dans le cas contraire, on peut présumer qu'une personne malveillante a tenté d'envoyer un message à Alice en se faisant passer pour Bob !



La signature pour l'authentification

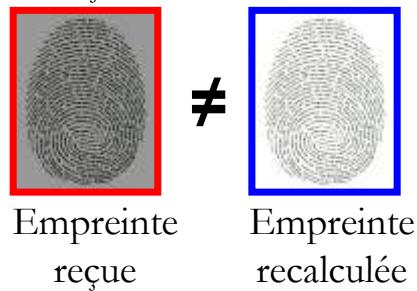


1)



La signature reçue est correcte

2)



La signature reçue est incorrecte

Quelques chiffres

Opération à effectuer	Durée du traitement
DES génération de la clé	6µsec
DES chiffrement	3241 Koctets/s
DES déchiffrement	3333 Koctets/s
MD5 création d'un condensat	36 250 Koctets/s
RSA chiffrement	4,23 Koctet/s
RSA déchiffrement	2,87 Koctet/s
SHA création d'un condensât	36 250 Koctet/s

Les standards des PKCS

- PKCS#1 : RSA Encryption Standard
- PKCS#3 : Diffie-Helman Key Agreement Standard
- PKCS#5 : Password-Based Encryption Standard
- PKCS#6 : Extended-Certificate Syntax Standard
- PKCS#7 : Cryptographic Message Syntax Standard
- PKCS#8 : Private-Key Information Syntax Standard
- PKCS#9 : Selected Attribute Types
- PKCS#10 : Certification Request Syntax Standard
- PKCS#11 : Cryptographic Token Interface Standard
- PKCS#12 : Personal Information Exchange Syntax Standard

Outils de cryptographie et d'authentification

Généralités

Chiffrement symétrique

Chiffrement asymétrique

Intégrité d'un message

PKI (Public Key Infrastructure)

Distribution des clés publiques

1. Alice envoie sa clé publique à Bob



2. Eve intercepte la clé et la remplace avec sa propre clé

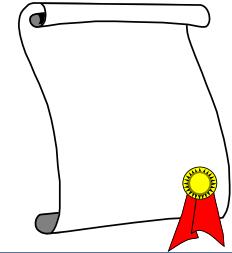


3. Eve peut décrypter tout le trafic et générer des signatures falsifiées

Distribution des clés publiques

Une autorité de certification (CA) résout ce problème:

- Alice envoie sa clé publique au CA
- Alice prouve qu'elle détient la clé privée correspondant à la clé publique envoyée
- Le CA vérifie l'identité d'Alice
- Le CA signe l'ensemble: clé publique et identité d'Alice avec sa clé privée, le document signé est appelé certificat
- Quand Bob reçoit le certificat d'Alice, il est sûr que la clé qui y est certifiée est celle d'Alice



Les certificats

- Un certificat électronique (aussi appelé certificat numérique ou certificat de clé publique) peut-être vu comme une carte d'identité numérique. Il est utilisé principalement pour identifier une entité physique ou morale, mais aussi pour chiffrer des échanges.
- Un certificat est un message signé avec la clé privée d'une entité
- On nomme cette entité une autorité de certification (autorité publique de gouvernance de l'internet)
- Le certificat peut être vérifié avec la clé publique de l'autorité de certification
- Les autorités de certification peuvent être organisées en arbre. Il suffit d'avoir confiance en une seule pour pouvoir vérifier un certificat quelconque de la hiérarchie.

Les certificats



Alice

Nom : Alice
Clé : WRP512KRJH
Mail : mathweb@free.fr
Profession : informaticienne

Alice fournit une fiche d'identité à l'organisme de certification.



Organisme de certification



Résumé

clé privée

Résumé

- * Vérifie les informations d'Alice
- * Ajoute ses propres informations
- * Calcule un résumé, le chiffre avec sa clé privée
- * Signe le certificat avec ce résumé



Bob



Résumé

||???

clé publique

Résumé

- * Télécharge le certificat
- * Calcule son résumé
- * "Ouvre" la signature avec la clé publique de l'organisme
- * Compare les 2 résumés

Standardisation des certificats numériques

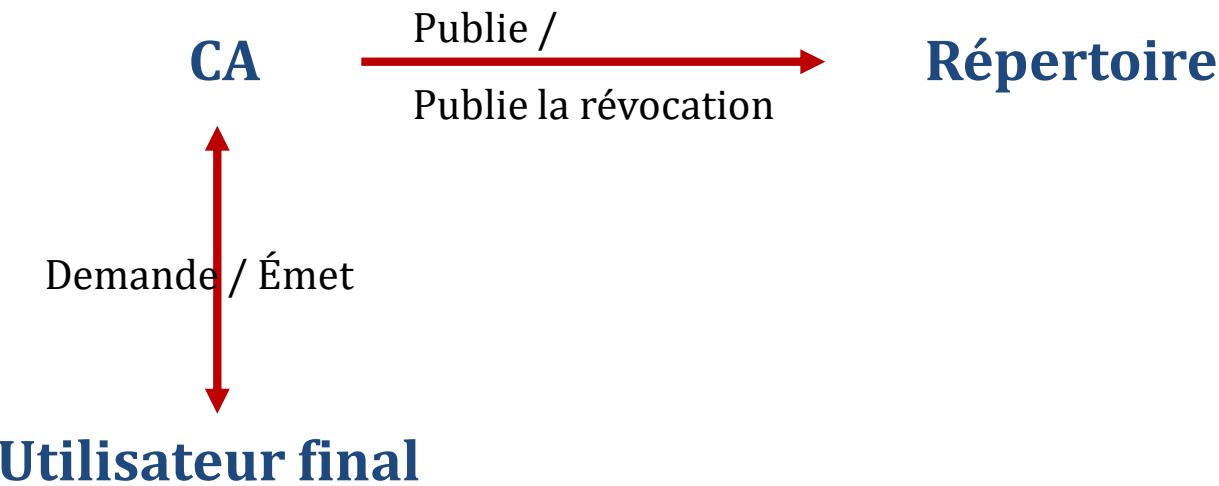
- Les utilisateurs de certificats étant de plus en plus nombreux, le format de ce certificat doit de ce fait être commun à tous les utilisateurs pour cette raison, les certificats numériques sont soumis à un standard.
 - Standard X509
 - Ce document électronique contient une clé publique, un certain nombre de champs propres à la norme X509 et une signature.
 - On parle de PKI (**Public Key Infrastructure**) : infrastructure à clé publique.

Composants d'une PKI

- Principaux
 - Autorité de certification CA : Cette autorité est une autorité de confiance qui a pour but de créer les certificats des utilisateurs.
 - Autorité d'enregistrement RA : Cette autorité a la tâche d'enrôler des nouveaux utilisateurs dans la PKI, elle reçoit les demandes de certificats CSR (Certificate Signing Request) ; elle à la responsabilité de vérifier la teneur de la demande.
 - Annuaire de publication
 - Administrateurs
- Complémentaires
 - Base de données
 - Serveur d'horodatage.
 - Serveur HTTP, SMTP, POP.

Infrastructure à clé publique (PKI)

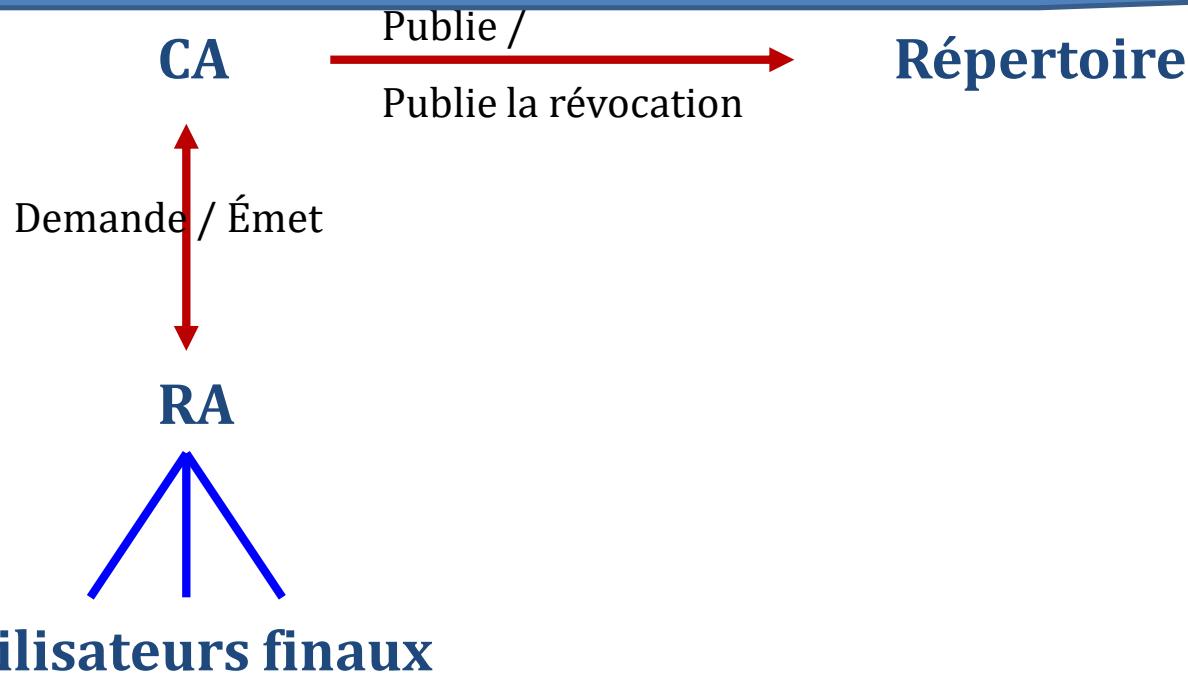
D'après le standard X.509:



Mais:

- Le CA doit vérifier les détails de chaque utilisateur
- Risques pour la sécurité du CA

Infrastructure à clé publique (PKI)

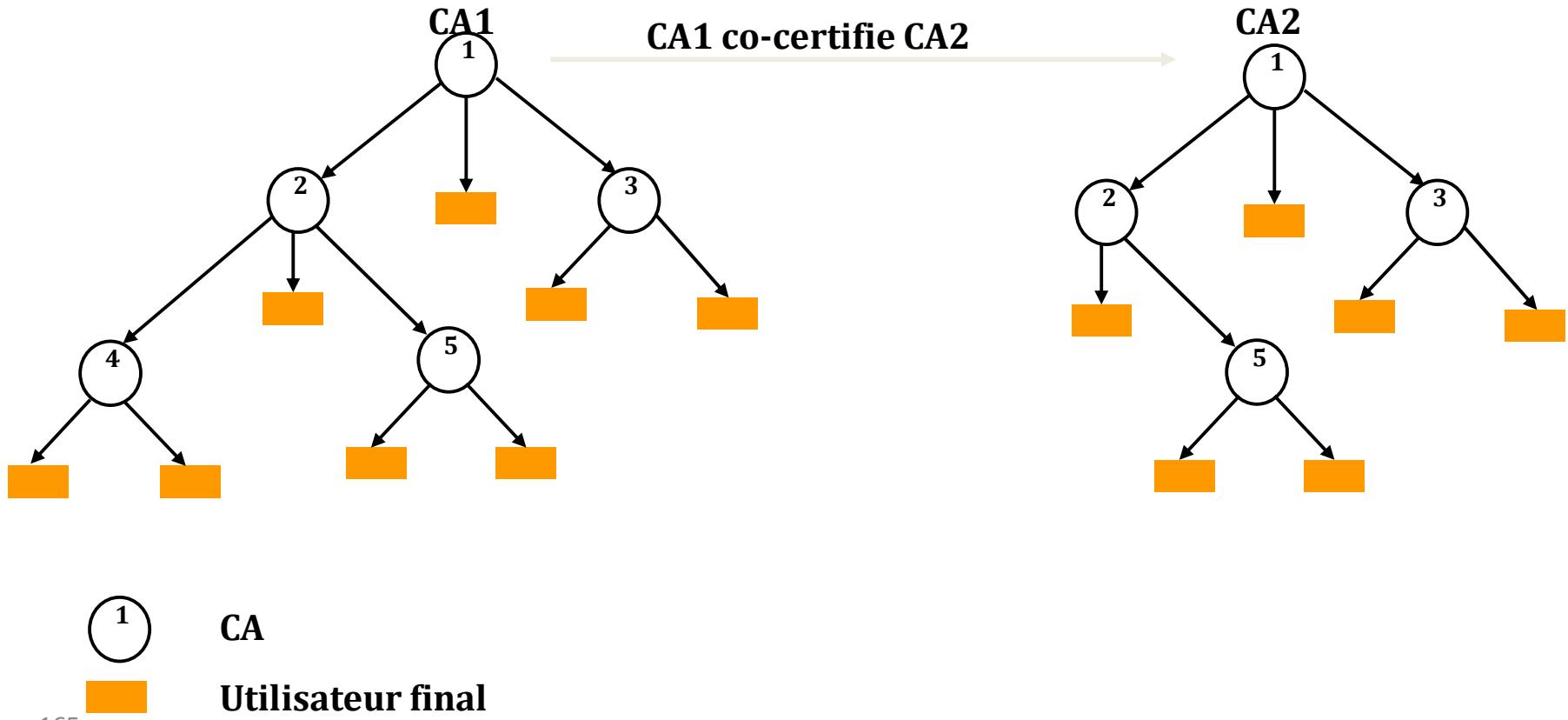


Autorité d'enregistrement (RA)

- Intermédiaire entre détenteur de clé et CA
- Vérifie les requêtes des utilisateurs et les transmet au CA
- Le niveau de vérification dépend de la politique de certification (CPS) mise en œuvre

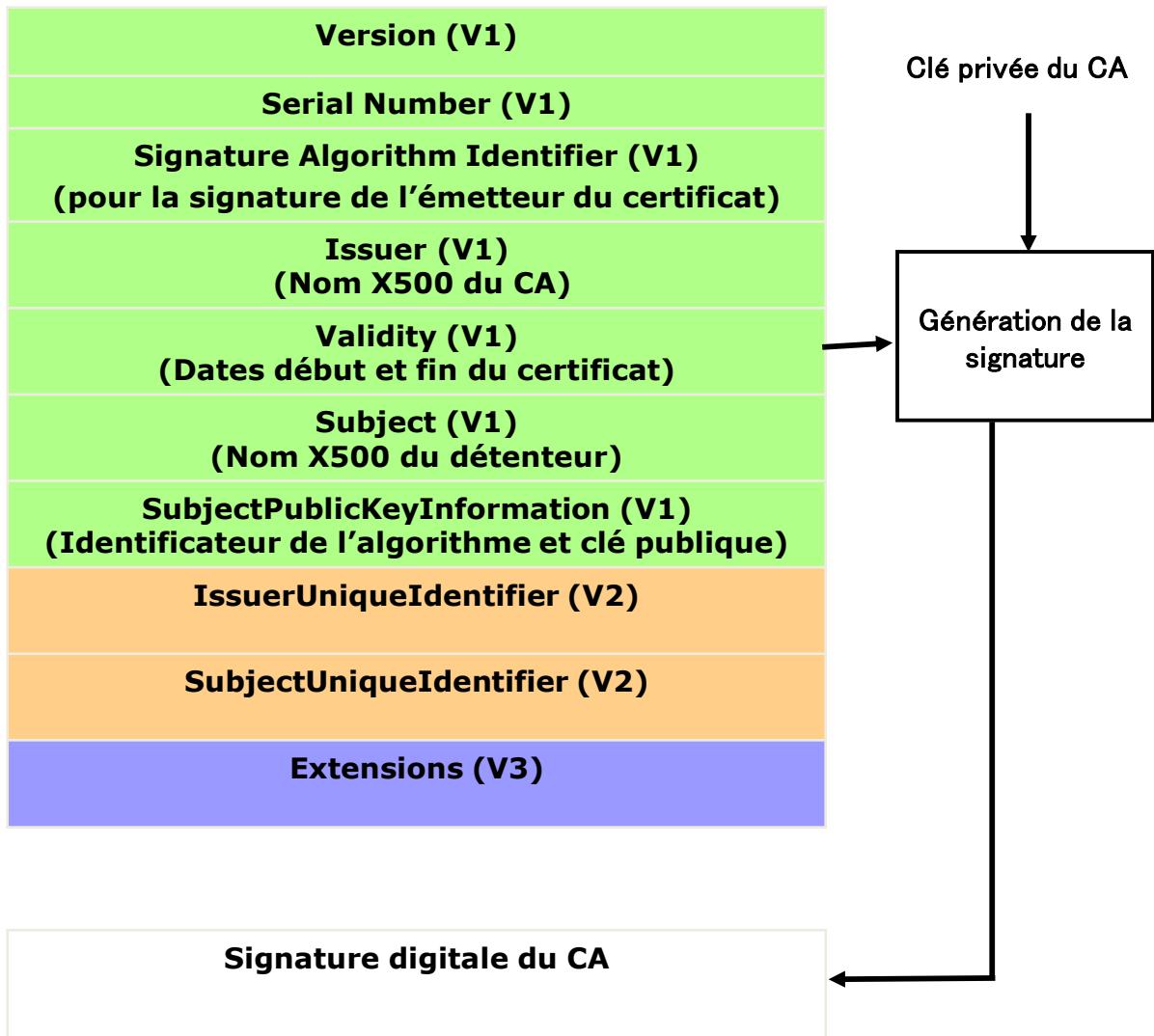
Modèle de confiance dans X.509

- Infrastructure hiérarchique
- Possibilité de certification entre 2 CAs appartenant à des arbres différents, c'est la co-certification



Certificats X.509

- Principal format utilisé pour les certificats
- Norme: ITU-T X.509, ou ISO/IEC 9594-8
- Versions successives:
 - 1988 : v1
 - 1993 : v2 = v1 + 2 nouveaux champs
 - 1996 : v3 = v2 + extensions



Structure d'un certificat X.509

— **IssuerUniqueIdentifier** identifie de façon unique la clé utilisée par le CA pour signer le certificat (cas où le CA a utilisé plusieurs clés depuis sa mise en œuvre)

— **SubjectUniqueIdentifier** différencie entre plusieurs clés publiques, issues par le même CA, appartenant à un même détenteur

```
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 2 (0x2)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=FR, O=CNRS, CN=CNRS
Validity
Not Before: Apr 27 05:46:49 2001 GMT
Not After : Apr 26 05:46:49 2011 GMT
Subject: C=FR, O=CNRS, CN=CNRS
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (2048 bit)
Modulus: 00:d0:e1:1e:21:3d:06:0b:ea:bd:5e:b4:a8:db:0f:
93:87:b4:ed:07:3d:8c:82:00:2d:ca:ff:b5:4a:8e:
e7:56:ad:8f:61:2e:f1:a0:2a:ab:f6:2a:dd:7c:2c:
b1:ef:75:55:0b:ac:09:4e:74:e1:c0:e7:0c:f0:
96:15:45:12:01:c8:9c:eb:c3:12:68:ed:63:10:16:
2e:cb:07:11:d9:81:a5:dc:29:82:9b:11:56:2:01:
1e:8a:5f:a7:e8:a9:58:11:44:56:93:5d:b3:4e:78:
70:2d:df:b6:fd:72:81:45:05:f1:ee:4d:ce:f1:be:
d9:3d:0c:90:20:45:8a:09:00:af:01:4c:da:20:0e:
86:bt:3a:b3:eb:27:8c:09:b9:bf:c0:a1:e4:40:dc:
3a:fd:ka:2a:bf:40:d5:2c:71:80:fb:f8:ba:fb:e4:
en:2a:0d:ab:2f:be:9a:f0:a7:76:cd:98:29:fc:0f:
2f:f0:42:f2:18:97:5b:c9:f6:cc:19:5f:ba:02:be:
12:d2:5c:cb:0:90:94:c0:b7:cb:06:04:ef:ff:30:ed:
32:2d:7a:4a:f7:93:bb:a0:09:a4:b4:ee:33:cb:d0:
83:9b:b5:b5:bd:00:da:8e:90:1e:59:9d:20:d5:4b:
1w:ed:d7:4c:4f:86:fa:1c:3a:2a:a1:e9:ec:05:a1:
9d:bf
Exponent: 45517 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints:
    CA:TRUE
X509v3 Subject Key Identifier:
47:59:AS:HS:07:74:49:03:8F:05:CP:CC:2E:A4:18:D6:10:CB:9E:3C
X509v3 Authority Key Identifier:
Keyid:56:BB:68:B9:D2:5C:7E:99:85:AS:51:C1:01:69:E3:5B:C4:99:CB:87
DirName:/C=FR/O=CNRS/CN=CNRS
serial:00
X509v3 Key Usage:
    Certificate Sign, CRL Sign
Signature Algorithm: md5WithRSAEncryption
06:03:47:83:72:45:90:c2:4e:e1:21:87:ab:17:a9:01:55:06:
ca:40:6d:55:a2:id:5e:ab:e2:14:23:59:e4:09:e2:00:f6:3c:
0d:31:06:0f:4b:a7:26:12:65:cd:ea:06:8a:72:eb:db:6c:db:
ba:5f:ef:79:36:25:9e:00:d7:f3:06:94:fb:83:44:29:26:37:
c7:ee:c9:87:ce:6c:86:80:1b:71:30:02:62:a1:f0:cd:62:c0:
53:0f:ec:7a:93:00:8d:7b:2e:33:a0:41:1d:aa:be:65:99:76:
f1:95:07:74:4b:a6:2f:53:75:df:4b:06:3d:4b:29:c4:f6:de:
8e:13:00:40:10:73:82:ad:15:7b:04:71:50:b5:37:13:f2:c8:
64:bb:a1:10:7e:36:c6:ad:af:4f:70:52:a6:d1:aa:cc:ba:
b0:48:59:12:8f:62:0d:ad:03:dd:4b:2a:e8:89:39:6b:51:2f:
ed:41:68:b7:30:67:db:27:55:6d:66:87:a3:51:09:80:61:71:
51:05:be:13:1d:df:41:10:fc:75:5f:ca:96:9b:18:ff:be:90:
81:b4:13:c0:72:11:08:0d:6a:9a:6a:07:bd:f4:63:2d:b4:60:
36:41:07:fa:3d:6a:a7:b0:90:04:76:83:bd:13:cb:34:e2:17:
99:04:0b:a1
```

Extensions d'un certificat X.509

- Le concept d'origine des certificats X.509 est de relier l'identité d'une entité à une clé publique
- Nouvelles situations: besoin d'avoir d'autres informations que l'identité
- Solution: Introduction de blocks de données pouvant supporter n'importe quel type d'informations pour satisfaire des besoins locaux

Extensions d'un certificat X.509

- Rajout de nouveaux champs sans la modification de la définition ASN.1 d'un certificat
- Permettre le rajout d'extensions selon le besoin des implémentations
- L'identificateur d'une extension est défini selon ITU-T Rec. X.660 | ISO/IEC 9834-1

Identificateur Extension1	Flag critique (1 ou 0)	Valeur Extension1
Identificateur Extension2	Flag critique (1 ou 0)	Valeur Extension2
Identificateur Extension3	Flag critique (1 ou 0)	Valeur Extension3

Extensions d'un certificat X.509

Les extensions sont classées en 4 catégories:

- Les extensions d'information sur la clé et la politique de sécurité
- Les extensions d'informations sur le détenteur et l'émetteur
- Les extensions de contraintes sur le chemin de certification
- Les extensions de révocation

Extensions d'information sur la clé et la politique de sécurité

- **Key Usage:** définit l'utilisation de la clé certifiée
 - digitalSignature
 - nonRepudiation
 - keyEncipherment
 - keyAgreement
 - keyCertSign/cRLSign
- **Extended Key Usage:** autres cas d'utilisation
 - ServerAuthentication
 - clientAuthentication
 - codeSigning
 - emailProtection
 - timeStamping

Extensions d'information sur la clé et la politique de sécurité

- **Private Key usage Period:** définit les dates début et fin de validité de la clé privée
 - Une signature peut être valide pour 10-20 années, mais la clé privée doit être utilisée uniquement pour 1 ou 2 années
- **Certificate Policies** Informations sur la politique du CA sous laquelle le certificat a été émis
 - X.509 délègue à la politique du CA tout ce qui concerne la sémantique de confiance du certificat. Plusieurs politiques servent pour protéger le CA de toute responsabilité

Extensions d'informations sur le sujet et l'émetteur

- Alternative Name (Subject / Issuer) ou General Name
 - Nom rfc822 (adresse mail)
 - Nom DNS (Nom DNS d'une machine)
 - uniformResourceIdentifier (URL)
 - Adresse IP
 - Adresse X.400
 - Nom EDI
 - OID
 - Toute autre forme de nom ...
- Subject directory attributes
 - Transporte une séquence d'attributs concernant le sujet du certificat: un rôle, une appartenance à un groupe, une autorisation, un numéro de téléphone...

Extensions de contraintes sur le chemin de certification

- **Basic Constraints**
 - Précise si le certificat émis est un certificat de CA ou pas (le détenteur du certificat peut agir comme un CA)
 - Si le certificat émis est un certificat de CA, une « Distance de certification » est définie
- **Name Constraints**
 - utilisé dans les certificats de CAs
 - indique un espace de noms où tous les noms des sujets ultérieurs dans le chemin de certification doivent figurer
- **Policy Constraints**
 - Identification explicite de la politique de sécurité

Extensions de révocation



- **CRL Distribution Points**
 - identifie les points de distribution de la CRL
- **Freshest CRL**
 - identifie la CRL qui a les informations les plus récentes