Sécurité Informatique
Chapitre 2: cryptographie

ITAHRIOUAN Zakaria

z.itahriouan@umi.ac.ma

Introduction

Les trames circulant sur un réseau public ou privé sont autant des lignes de lectures disponibles à un utilisateur de passage ayant quelques connaissances en informatique.

Le but principal de la cryptographie est de garantir la confidentialité, l'intégrité des données et l'authentification de l'émetteur des messages.

1- Principes de cryptographie

Place du chiffrement

Le mécanisme de chiffrement existe à différents niveaux :

- liaison : mise en place de boîtes noires sur les supports de transmission
- réseau : des équipements spécialisés sont placés sur chacun des sites, au niveau des routeurs

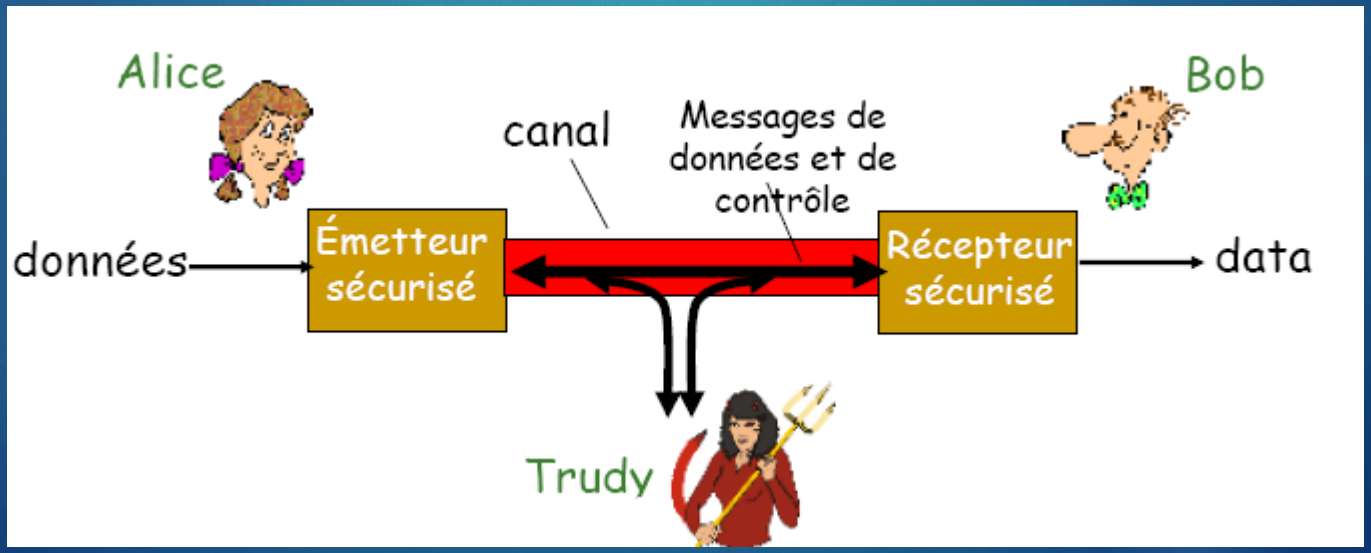
- de bout en bout : seules les données constituant l'information transmise sont chiffrées. Il est mis en œuvre dans les applications du modèle TCP/IP

L'ensemble repose, dans tous les cas, sur un algorithme donné, une clé ou un couple de clés associées et un mécanisme de distribution des clés

1- Principes de cryptographie

Amis et ennemis : Alice, Bob, Trudy

- Bien connus dans le monde de la sécurité !
- Bob et Alice veulent communiquer "de manière sûre"
- Trudy (*intruder*) peut intercepter, effacer et ajouter des messages



1- Principes de cryptographie

Qui pourraient être Bob et Alice?

- Des êtres humains !
- Des serveurs ou browsers Web pour des transactions électroniques (ex : achats en ligne)
- Client/serveur de banque en ligne
- Serveurs DNS
- Routeurs échangeant des mises à jour de tables de routage
- …

1- Principes de cryptographie

Il y a des méchants !

Q : Que peut faire un "méchant" ?

R : Beaucoup de choses!

- *Écoute* : interception de messages
- *Insertion* active de messages dans la connexion
- *Imitation* : falsification (spoof) de l'adresse source dans le paquet (ou tout autre champ du paquet)
- *Détournement* : "prise de contrôle" de la connexion en cours en écartant l'émetteur ou le récepteur et en prenant sa place
- *Déni de service* : empêcher le service d'être utilisé par les autres (ex : en surchargeant les ressources)

1- Principes de cryptographie

Vocabulaire de cryptographie

- *Chiffrer : transcrire, à l'aide d'un algorithme paramétrable un message clair en une suite incompréhensible de symboles*
- *texte en clair : le message à chiffrer*
- *texte chiffré : le résultat du chiffrement*
- *Déchiffrer : retrouver le texte en clair à partir du texte chiffré à l'aide d'un algorithme paramétrable*
- *clé : le paramètre des algorithmes de chiffrement et de déchiffrement*

1- Principes de cryptographie

Vocabulaire de cryptographie

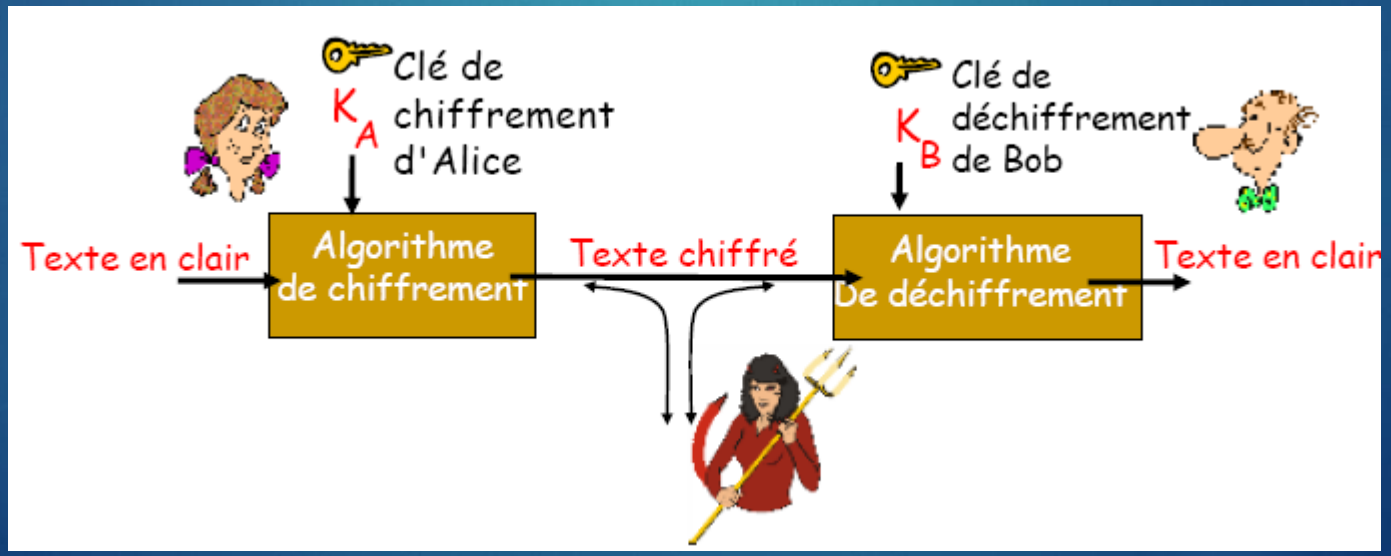
- *Décrypter : retrouver le texte en clair à partir du texte chiffré sans la clé*
- *Cryptographie : science du chiffrement*
- *Cryptanalyse : science du décryptage*
- *Cryptologie : cryptographie et cryptanalyse*
- *Cryptosystème : ensemble des méthodes de chiffrement et de déchiffrement utilisables en sécurité*

1- Principes de cryptographie

Vocabulaire de cryptographie

Cryptographie à clé symétrique : clés identiques pour l'émetteur et le récepteur

Cryptographie à clé publique : clé de chiffrement publique, clé de déchiffrement secrète (privée)



1- Principes de cryptographie

Propriétés des crypto systèmes

- Les algorithmes de chiffrement et de déchiffrement doivent permettre d'atteindre des vitesses de chiffrement élevés et utiliser peu d'espace mémoire
- Le chiffrement doit être difficile à casser que ce soit avec des messages chiffrés seuls ou un échantillon de messages en clair avec leur message chiffré correspondant
- Redondance : le destinataire doit pouvoir vérifier la légitimité d'un message : on ne doit pas pouvoir créer des textes "ressemblant" à des textes chiffrés différents
- Fraîcheur : le destinataire doit pouvoir s'assurer qu'un message est récent

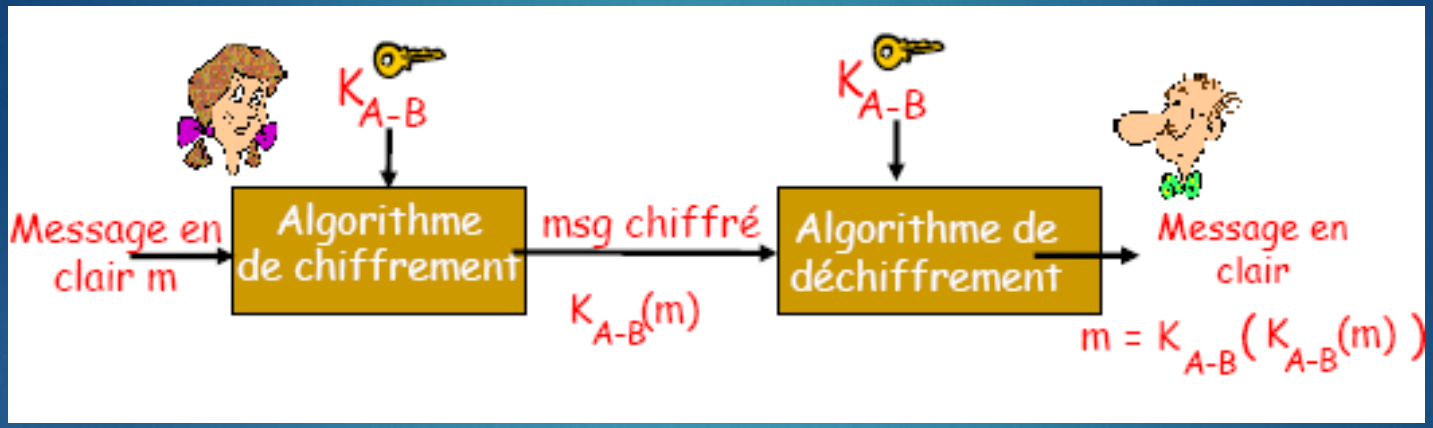
1- Principes de cryptographie

Principe de Kerckhoff

- Principe de Kerckhoff :
  - tous les algorithmes doivent être publics ; seules les clés sont secrètes
- Conséquences :
  - le véritable secret réside dans la clé
  - l'algorithme public doit être fort et la clé doit être longue
- Problèmes :
  - quelle longueur de clés choisir ?
  - quelle fréquence de renouvellement des clés choisir ?
  - comment s'échanger les clés ?

2- confidentialité

Cryptographie à clé symétrique



Cryptographie à clé symétrique : Bob et Alice partagent une même clé (connue) : K

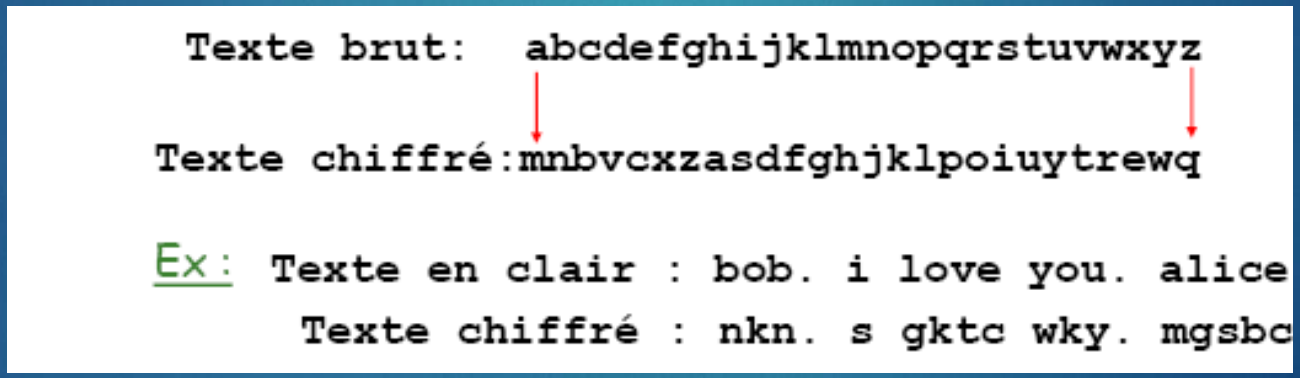
- Ex : la clé consiste à connaître le mode de substitution dans un chiffrement par substitution monoalphabétique
- Comment Bob et Alice se mettent-ils d'accord sur la valeur de la clé ?

2- confidentialité

Cryptographie à clé symétrique

Chiffrement par substitution

- Substitution monoalphabétique : substituer une lettre par une autre



2- confidentialité

Cryptographie à clé symétrique

Chiffrement par substitution

- Substitution homophonique: à un caractère de texte en claire on fait correspondre plusieurs caractères de texte chiffré
- Par exemple A peut correspondre à 5, 13, 25 ou 56 ; B peut correspondre à 7, 19, 31, ou 42 ; etc

2- confidentialité

Cryptographie à clé symétrique

Chiffrement par substitution

- substitution poly alphabétique: consiste à substituer une lettre du message en clair, par une autre choisie en fonction d'un état du cryptosystème. Ce changement de lettre tout au long du processus, s'obtient à l'aide d'une clé.
- *Exemple: texte en claire: « ABCBACCBAACBB »*

  Clé: « DBBCBAACD »

Chiffrement: lire le caractère correspondant du texte en clair en utilisant la grille et le mot-clé associé

K \ C	ABCD
A	CBDA
B	DCAB
C	CABD
D	BDAC

Grille poly alphabétique

Texte chiffré: « BCAADDDAB BACA »

2- confidentialité

Cryptographie à clé symétrique : DES

DES: Data Encryption Standard

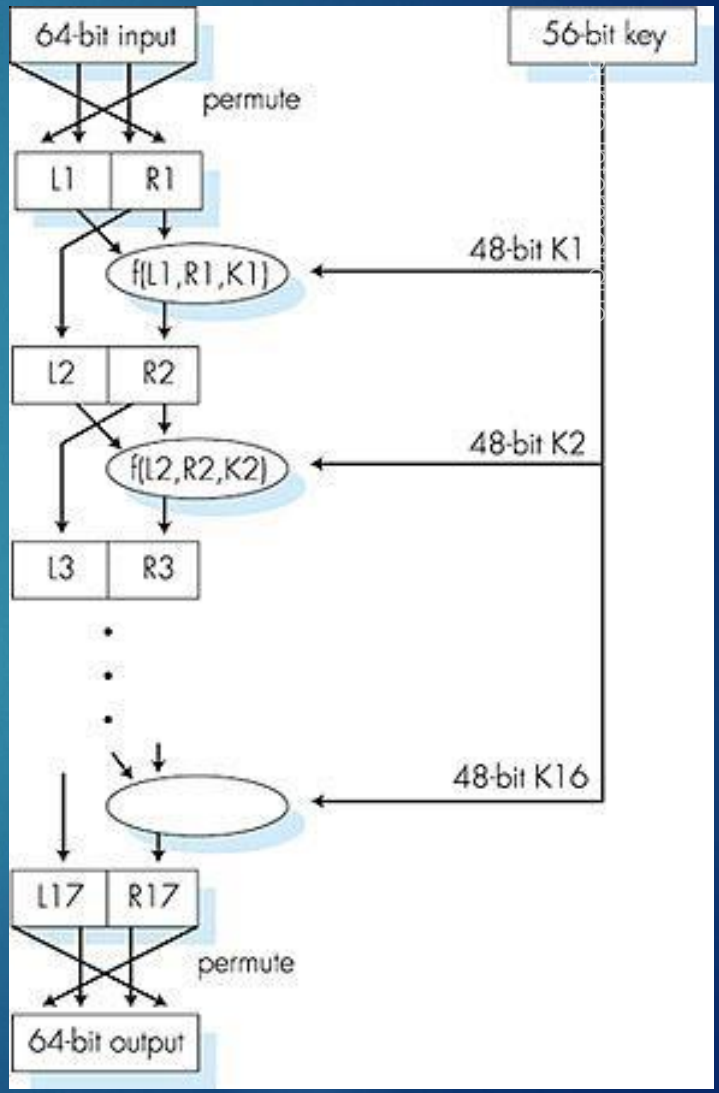
- Standard de chiffrement américain [NIST (National Institute of Standards and Technology) 1993]
- Clé symétrique sur 56 bits, input plaintext 64 bits
- À quel point le DES est-il sûr ?
  - Challenge du DES : phrase ("Strong cryptography makes the world a safer place") chiffrée avec une clé de 56 bits déchiffrée en 4 mois (de manière brutale)
- Rendre le DES plus sûr :
  - Utiliser 3 clés successivement (3-DES) sur chaque donnée

2- confidentialité

Cryptographie à clé symétrique : DES

Algorithme du DES

- Permutation initiale
- 16 boucles identiques de la fonction application, utilisant chacune 48 bits différents de la clé
- Permutation finale



2- confidentialité

Cryptographie à clé symétrique : AES

AES : Advanced Encryption Standard

- Standard NIST à clé symétrique (Nov. 2001) remplaçant le DES
- Traite les données par blocs de 128 bits
- Clés de 128, 192, ou 256 bits
- Le déchiffrement brutal (essai de toutes les clés) prenant 1 sec avec le DES prend 149 trillions d'années pour AES

2- confidentialité

Principaux algorithmes symétriques

- **DES** (*Data Encryption Standard*)
  - Variante : Triple DES
- RC2, RC4 et RC5
- **IDEA** (*International Data Encryption Algorithm*)
- Blowfish
- **AES** (*Advanced Encryption Standard*)

2- confidentialité

Cryptographie à clé publique

Crypto à clé symétrique

- Nécessite le partage d'une clé entre l'émetteur et le récepteur
- Q: comment se mettre d'accord sur la clé au départ (surtout s'ils ne se sont jamais rencontrés) ?

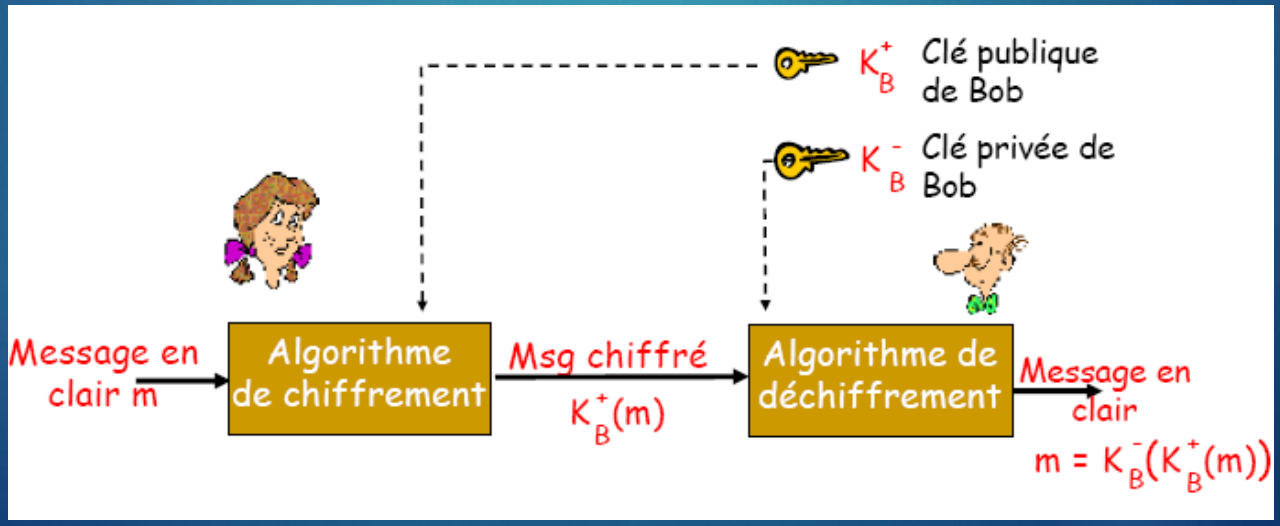
Crypto à clé publique

- Approche radicalement différente
- L'émetteur et le récepteur ne partagent *pas* de clé secrète
- clé de chiffrement *publique* connue de *tous*
- la clé de déchiffrement *privée* n'est connue que du récepteur

2- confidentialité

Cryptographie à clé publique

- Alice possède un message confidentiel qu'il souhaite transmettre à Bob.
- Bob construit deux clés :
  - une clé de chiffrement publique qu'elle transmet à Alice.
  - une clé de déchiffrement privée qu'elle conserve soigneusement.
- Alice utilise la clé publique pour chiffrer le message, et le transmets à Bob.
- Bob utilise la clé privée pour déchiffrer le message reçu.



2- confidentialité

Principaux algorithmes asymétriques

- **RSA** (pour Ron **R**ivest, Adi **S**hamir, Len **A**delman) (http://www.rsa.com)
- **Diffie-Hellman** (http://www.ietf.org/rfc/rfc2631.txt)
- ElGamal

2- confidentialité

Algorithmes de chiffrement par clé publique

Besoins :

1. Besoin de $K_B^+( )$ et de $K_B^-( )$ telles que

$$K_B^-(K_B^+(m)) = m$$

2. À partir de la clé publique K_B^+, il devrait être impossible de calculer la clé privée K_B^-

2- confidentialité

Diffie - Hellmann

- Algorithme a clé publique inventé en 1976
- Objectif :
  - Permet l' échange d'une clé secrète sur un domaine non sécurise, sans disposer au préalable de secret
- Utilisation :
  - entre autres, dans SSL/TLS (Netscape)
- Repose sur :
  - connaissant $g^a \bmod p$ et $g^b \bmod p$,

    -> il est très difficile d'en déduire a et b.

2- confidentialité

Diffie – Hellmann

- Privé :
  - a pour Alice
  - b pour Bob
- Public :
  - p : nombre premier
  - g: appelé générateur
  - clé publique d'Alice
  - clé publique de Bob

- Algorithme -

Alice génère a (privé) $< p-1$ et calcule $g^a \bmod p$ (public)

Bob génère b (privé) < p - 1 et calcule $g^b \bmod p$ (public)

Echange des clés publiques

Alice calcule $(g^b)^a \bmod p$

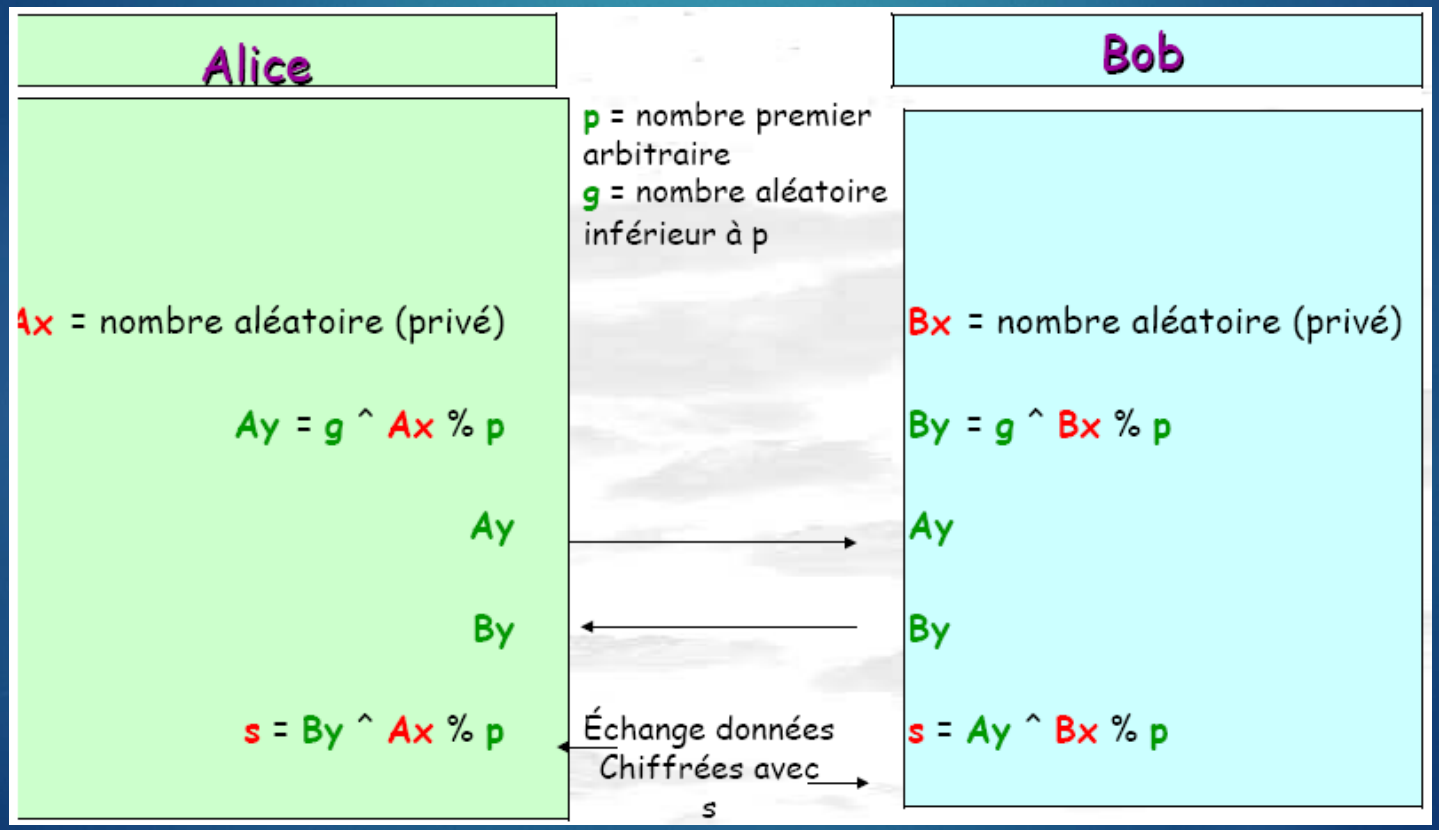
Bob calcule $(g^a)^b \bmod p$

$(g^b)^a \bmod p = (g^a)^b \bmod p = g^{ab} \bmod p = k$ (secret partagé)

- Contrairement à RSA, il ne permet pas de chiffrer des documents. Diffie-Hellman est souvent associé à DSS (Digital *Signature* Standard)

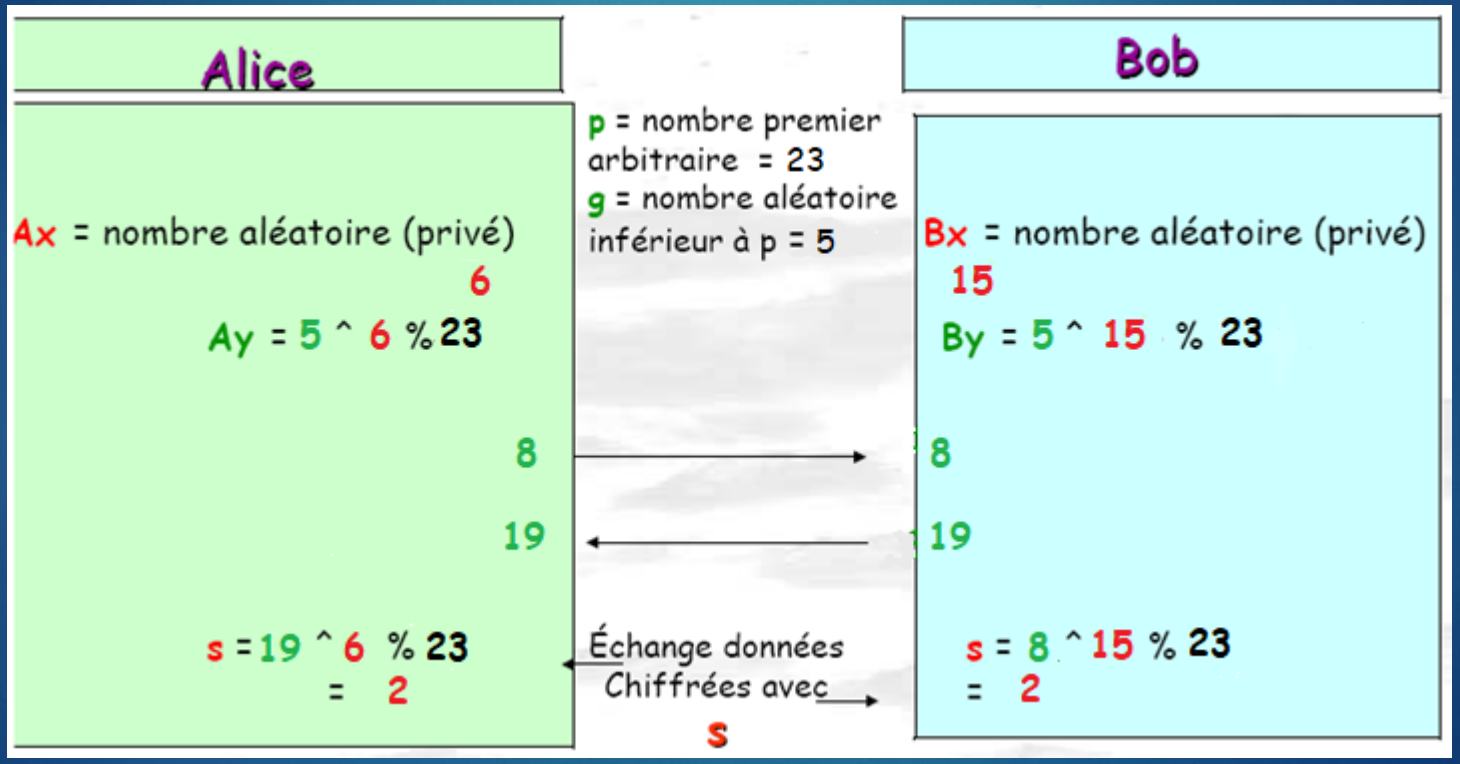
2- confidentialité

Diffie – Hellmann (exemple)



2- confidentialité

Diffie – Hellmann (exemple)



2- confidentialité

Diffie – Hellmann (exercice)

À partir du nombre premier: **p = 419** et du nombre **g = 7**
gênerez trois clés de chiffrement secrètes.

2- confidentialité

RSA :

- Inventé en 1977 par Rivest, Shamir et Adlleman
- Parmi les plus utilisés

- Sécurité :
  - Le RSA a clé de 512 bits a récemment été cassé, mais ce n'est pas a la portée de tout le monde.
  - Incassable actuellement sur du 768 ou 1024 bits.

2- confidentialité

RSA : choix des clés

1. Choisir deux grands nombres premiers *p*, *q*. (ex 1024 bits chacun)
2. calculer $n = pq$, $z = (p-1)(q-1)$
3. choisir *e* (avec $e<n$) n'ayant aucun facteur commun avec z. (*e* et *z* sont premiers entre eux).
4. choisir *d* tel que *ed-1* soit divisible par *z*. (*ed* mod *z* = *1*).
5. La clé p*ublique* est $\underbrace{(n,e)}_{K_B^+}$. La clé p*rivée* est $\underbrace{(n,d)}_{K_B^-}$.

2- confidentialité

RSA : chiffrement, déchiffrement

0. soient (n,e) et (n,d) calculés comme précédemment

1. Pour chiffrer le caractère m, calculer
   $c = m^e \bmod n$ (cad le reste de la division de m^e par n

2. Pour déchiffrer le caractère c, calculer
   $m = c^d \bmod n$ (cad le reste de la division de c^d par n)

$$m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$$

2- confidentialité

Exemple RSA :

Bob choisit $p=5$, $q=7$. alors $n=35$, $z=24$.

$e=5$ (e et z premiers entre eux).

$d=29$ ($ed-1$ divisible par z).

	lettre	message m	m^e	$c = m^e \bmod n$
Chiffrement	l	12	248832	17

	c	c^d	$m = c^d \bmod n$	lettre
Déchiffrement	17	48196857210675091509141182522307169 7	12	l

2- confidentialité

Exercice RSA :

- On prend deux nombres premiers : p = 17, q = 19

1- calculer n.

2- calculer z et choisir e.

3- proposer un algorithme permettant de donner les valeurs possibles de d.

4- vérifiez que les 3 premières valeurs de d sont valables pour réaliser des clés en présentant les différentes clés.

5- cryptez le message « hello » à l'aide d'une clé publique.

6- décryptez le message obtenu dans -5- à l'aide de sa clé privée.

2- confidentialité

RSA : autre propriété importante

$$\underbrace{K_B^-(K_B^+(m))}_{\text{Utiliser la clé publique, PUIS la clé privée}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{Utiliser la clé privée, PUIS la clé publique}}$$

Le résultat est le même !

2- confidentialité

Chiffrement hybride

Le chiffrement hybride d'un message M se déroule selon les étapes suivantes:

- l'émetteur choisit une clé symétrique K aléatoire. Il utilise ensuite cette clé K pour chiffrer (symétriquement) le message M.
- il chiffre (asymétriquement) la clé K avec la clé publique du destinataire.
- Il envoie à son destinataire les chiffrés de M et de K.
- Le destinataire déchiffre d'abord la clé K, puis l'utilise pour retrouver M.

Le chiffrement hybride cumule les avantages de la cryptographie asymétrique avec ceux de la cryptographie symétrique:

- la clé K étant courte le chiffrement asymétrique de cette clé reste efficace.
- Le message M, potentiellement très long, est traité par un chiffrement symétrique.

3- authentification & Intégrité

La notion de confidentialité ne suffit pas forcément :

- qui m'a réellement envoyée le message ?
- quelqu'un a-t-il pu usurper son identité ?
- ai-je bien reçu le message complet ?
- quelqu'un a-t-il pu remplacer le message initial par un autre ?

La confidentialité ne répond pas à ces questions !

3- authentification & Intégrité

Fonctions de hachage

Dans la pratique, un sceau, signature, c'est petit!!

– On condense le texte avant de le sceller ou de le signer

– La fonction qui effectue ce "résumé" est appelée fonction de hachage

- Choix des algorithmes de hachage

– Une fonction de hachage "H" transforme une entrée d'une dimension variable "m" et donne comme résultat une sortie de données inférieure et fixe "h"

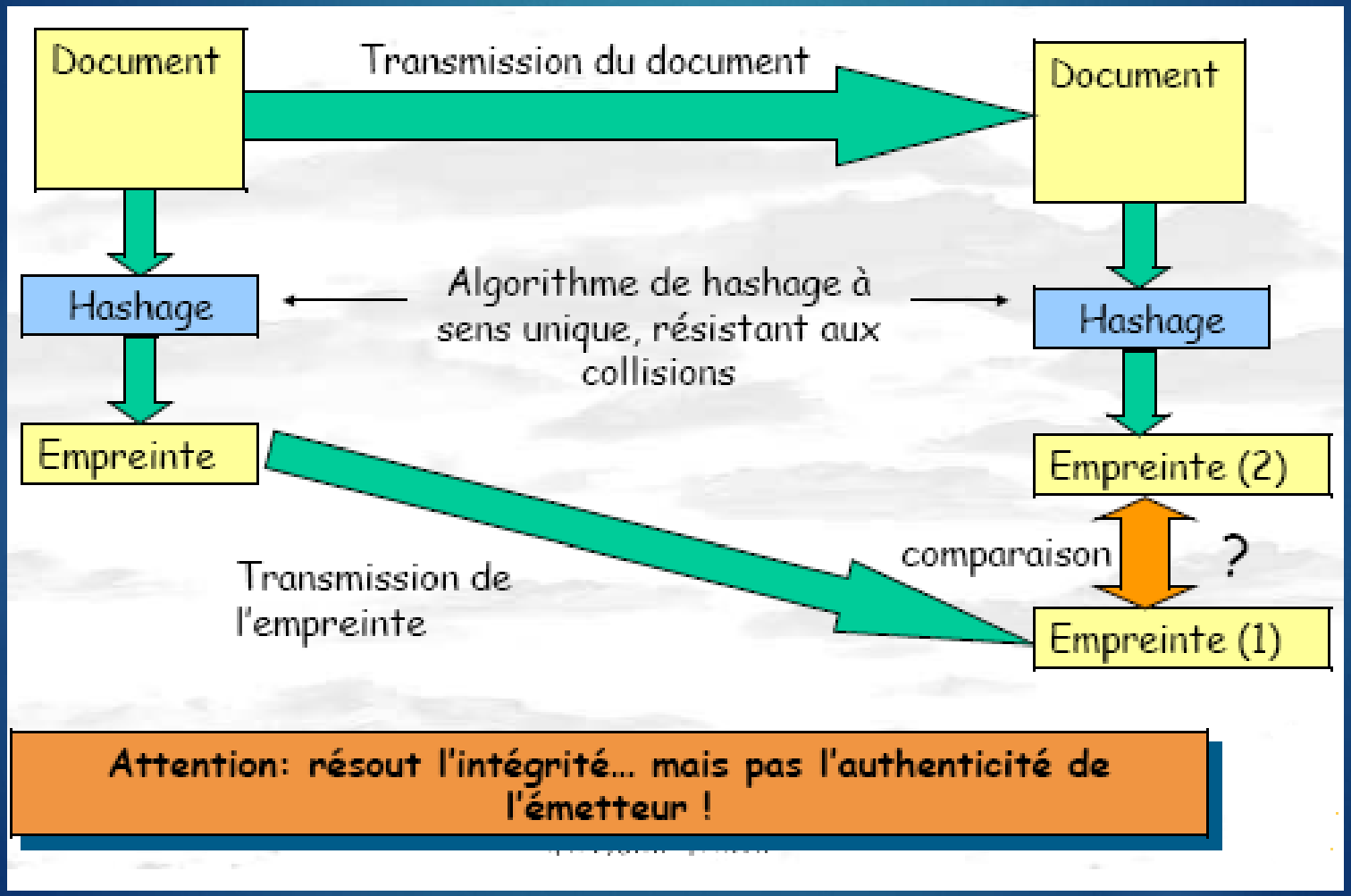
fonction de hachage doit remplir les conditions

- L' entrée peut être de dimension variable.
- La sortie doit être fixe.
- H(m) doit être relativement facile à calculer.
- H(m) doit être une fonction a sens unique.
- impossible de trouver m a partir de h.
- H(m) doit être "sans collision" c'est-à-dire que deux messages distincts doivent avoir très peu de chances de produire la même signature:

$$M1 \neq M2 \wedge H(M1)=H(M2)$$

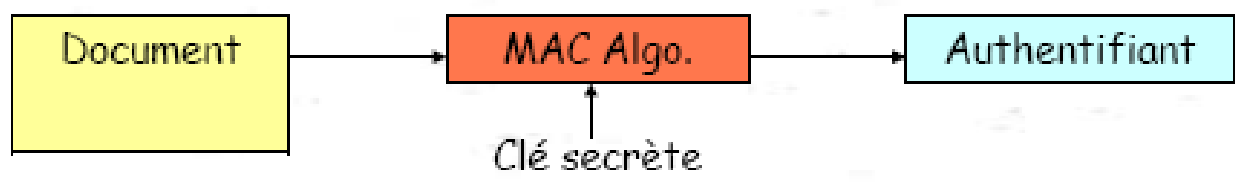
3- authentification & Intégrité

Contrôler l'intégrité



3- authentification & Intégrité

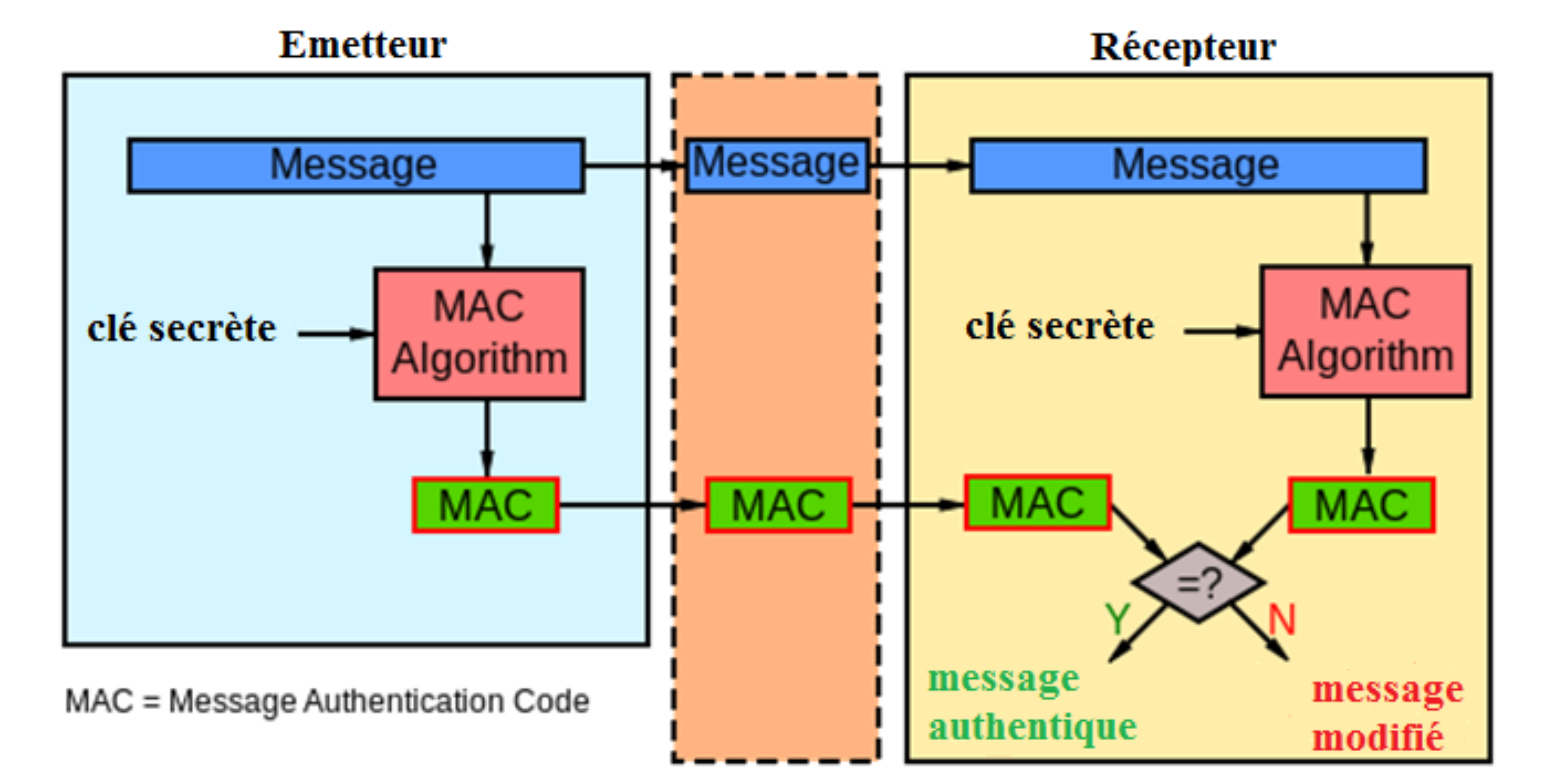
Code d'authentification de message (MAC)



- Objectif : Intégrité et authenticité
- Algorithmes de MAC :
  - Fonction de hachage
    - compression et facilité de calcul.
  - Convenir au préalable d'une clé secrète.
  - Pas (ou très peu) de collisions.
- Exemples : DES-CBC, HMAC …

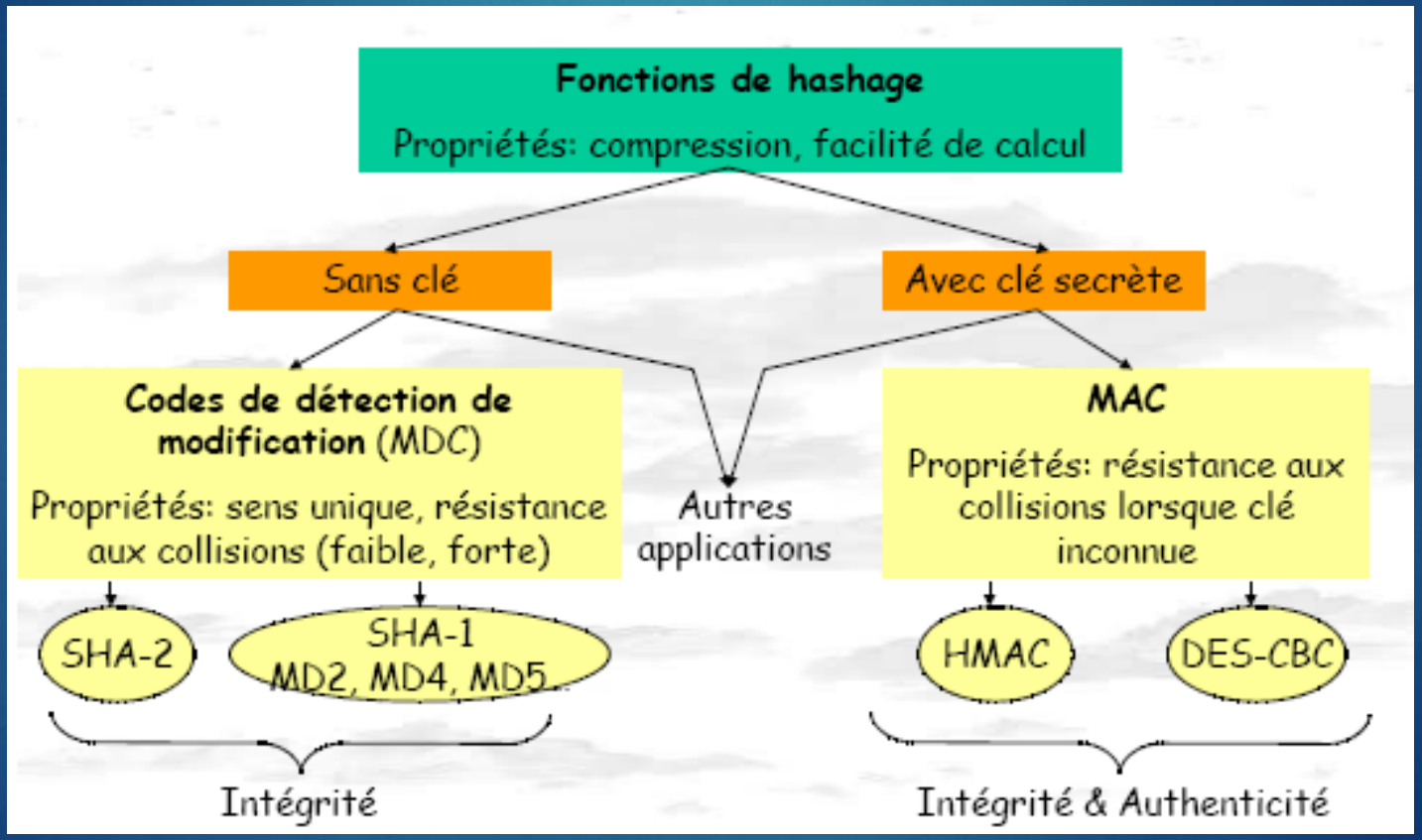
3- authentification & Intégrité

Code d'authentification de message (MAC)



3- authentification & Intégrité

Vue d'ensemble sur les algorithmes de hachage



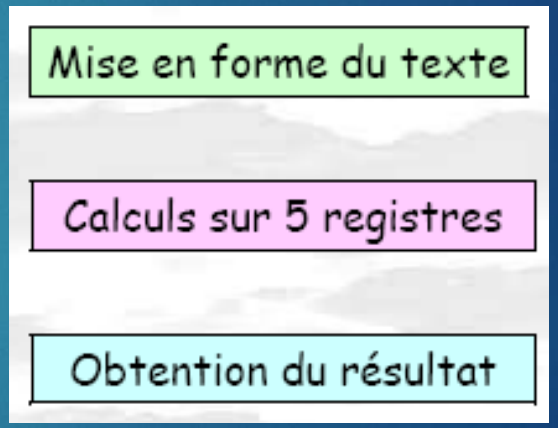
3- authentification & Intégrité

Aperçu du SHA-1

- Historique:
  - conçu en 1995
  - Adopté comme algorithme standard de hachage par le gouvernement américain

Objectifs:

- fonction de hachage qui retourne toujours un résumé sur 20 octets, quelque soit la taille du texte en entrée



3- authentification & Intégrité

SHA

- Points positifs:

⇒ création de SHA2 (SHA-256, SHA-384 et SHA-512).

⇒ Plus rapide et plus sûr que la famille MD (2 .. 5)

=> SHA-512 effectue un traitement par mots de 64 bits (et non 32), d'ou une plus grande rapidité.

- Points négatifs:

– Résistance a la collision devenant de plus en plus critique

– Récemment, un groupe de chercheurs a montré qu'il **n'est pas infaillible.**

- **SHA-1 devient obsolète !!**

3- authentification & Intégrité

CRC (Cyclic Redundancy Check)

- code qui permet de détecter les erreurs de transmission
- Il est obtenu en ajoutant aux données transmises (m) les bits de redondance (R) obtenus en divisant C_0 par *G*.
  - *d* le degré de *G*.
  - C_0 étant la chaine des bits obtenue en rajoutant *d* zéros à la fin de m,
  - *G* étant les bits représentant le polynôme générateur
- Le code C envoyé est la concaténation de m et de R. À l'arrivée, le récepteur détecte une erreur si la division de C reçu par G ne donne pas un reste nul.
- Les polynômes générateurs les plus couramment employés sont :
- CRC-32 (Ethernet): $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$
- CRC-12: $X^{12}+X^{11}+X^3+X^2+X+1$
- CRC-16: $X^{16}+X^{15}+X^2+1$

3- authentification & Intégrité

CRC (Exemple)

I = 1101011011
G = 10011
d = 4 (R est composé de 4 bits)
C_0 = I suivi de *d* zéros = 11010110110000
Il faut calculer le reste de la division de C_0 par G

```
11010110110000 | 10011
10011          | 1100001010
 10011
 10011
       10110
       10011
         10100
         10011
           1110
```

On fait une division classique mais en binaire.
R = **1110**

3- authentification & Intégrité

CRC (Exercice)

1- Montrer dans l'exemple précèdent comment le récepteur vérifie s'il y a des modifications dans le message?

2- Calculez le CRC de 10101001001000101010101 avec le polynôme générateur :

$$x^6 + x^5 + x^3 + x + 1$$

3- vérifier si le résultat obtenu dans 2 est correct.

3- authentification & Intégrité

Signatures numériques

Techniques cryptographiques analogues aux signatures manuscrites.

- L'émetteur (Bob) signe le document de manière numérique et établit qu'il est le créateur/propriétaire du document.
- Vérifiable, non falsifiable : le récepteur (Alice) peut prouver à quelqu'un que Bob et personne d'autre n'a signé ce document

3- authentification & Intégrité

Comment "signer" un message ?

- Bob signe m en chiffrant avec sa clé privée K^-_B, créant le message "signé" $K^-_B(m)$
- Supposons qu'Alice reçoit le msg m et la signature numérique $K^-_B(m)$
- Alice vérifie que m a été signé par Bob en appliquant la clé publique de Bob K^+_B à $K^-_B(m)$ et vérifie que $K^+_B(K^-_B(m)) = m$.
- Si $K^+_B(K^-_B(m)) = m$, la personne qui a signé m a forcément utilisé la clé privée de Bob.

Alice vérifie ainsi que :

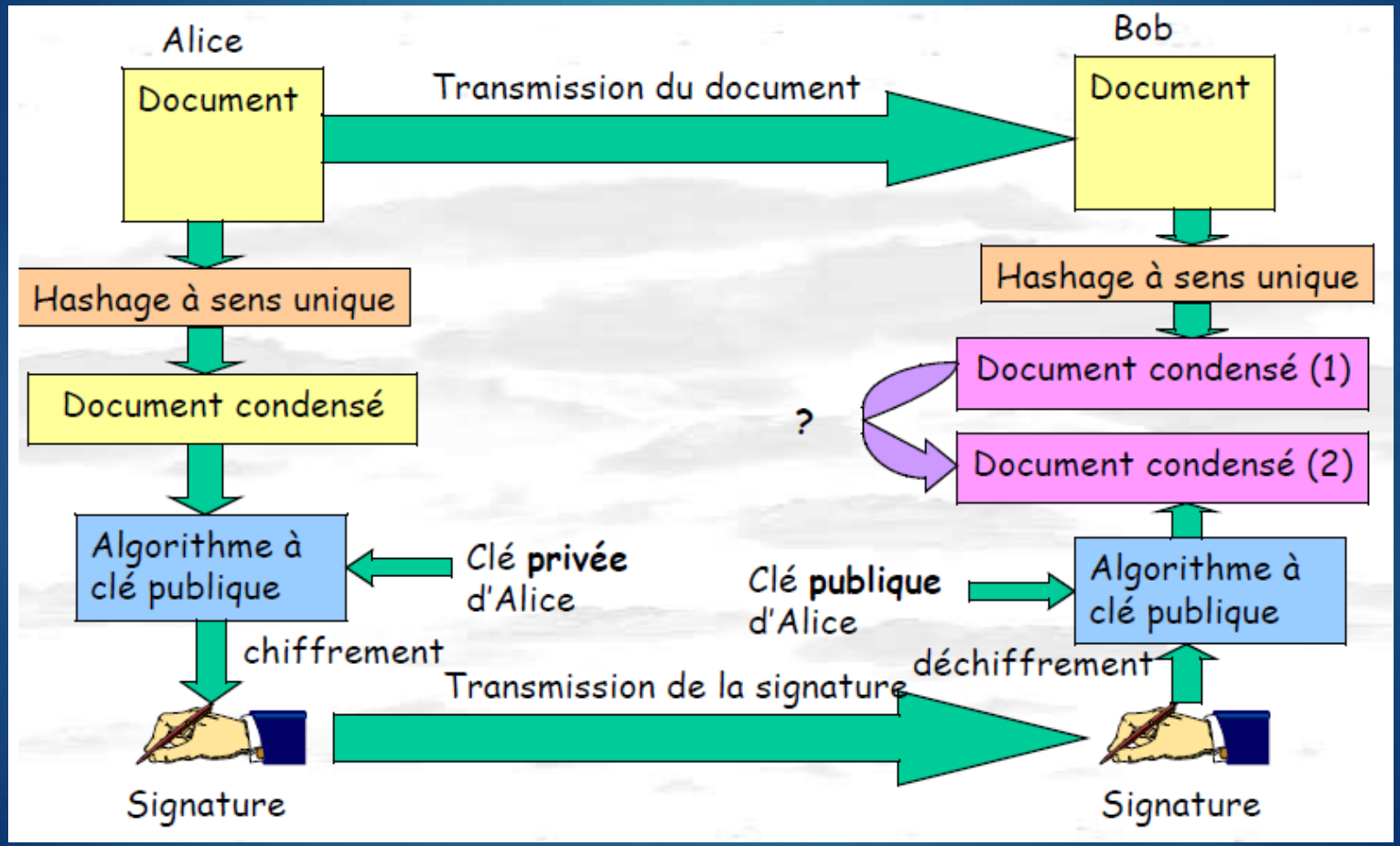
- Bob a signé m.
- Personne d'autre n'a signé m.
- Bob a signé m et pas m'.

Non-répudiation:

- Alice peut emporter m et la signature $K_B(m)$ à un procès et prouver que Bob a signé m.

3- authentification & Intégrité

Comment "signer" un message ? (suite)



3- Infrastructure de gestion des clés

Certificats : pourquoi ?

- chiffrement & signatures supposent l'authenticité des clés publiques
  - signature garantit que le message provient bien du détenteur de la clé privée mais
    - A qui appartient clé privée/publique ?
  - Chiffrement garantit que le message ne pourra être déchiffré que par le détenteur de la clé privée (associée à la clé publique utilisée lors du chiffrement) mais
    - A qui appartient cette clé publique ?

Est-on sûr qu'il ne s'agit pas d'un usurpateur ?

3- Infrastructure de gestion des clés

Certificats : pourquoi ?

Scénario :

- Alice et Bob veulent s'envoyer des messages
- Trudy = pirate

Trudy

- Remplace la clé publique d'Alice par la sienne.
  - Trudy peut maintenant lire les courriers destinés à Alice et signer des message en se faisant passer pour Alice !!
- si Bob envoie un message « M » chiffré à Alice,
  - Bob va chiffrer M avec la clé publique de Trudy (croyant que c'est la clé d'Alice).
- Trudy pourra
  - déchiffrer les messages destinés à Alice avec sa clé privée,
  - lire ainsi le courrier confidentiel d'Alice.

Les signatures et les MACs ne résolvent pas entièrement le problème de l'authenticité.
On introduit alors la notion de certificat !

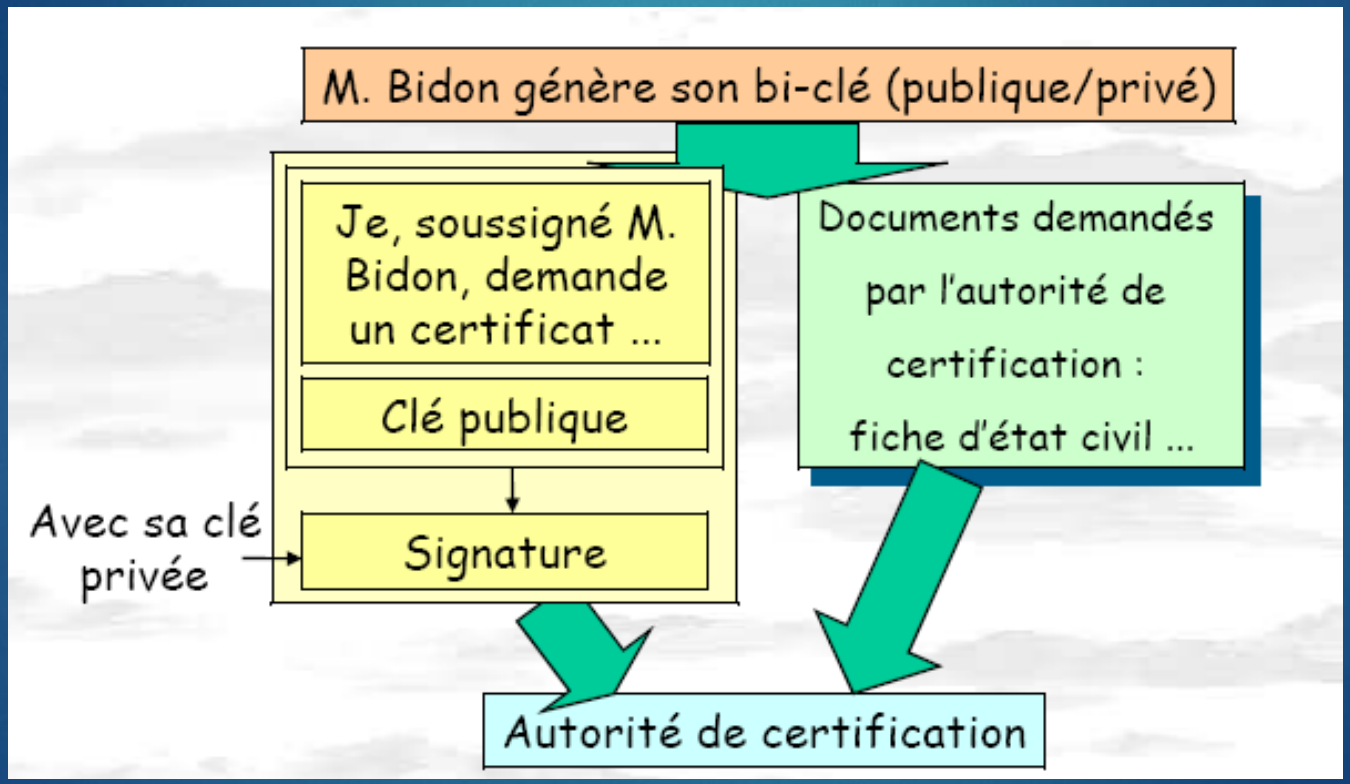
3- Infrastructure de gestion des clés

Certificats : quoi ?

- permet d'établir un environnement de confiance entre deux entités distantes ayant besoin de communiquer entre elles et de s'échanger des informations :
  - non-répudiables (nécessité de signature)
  - confidentielles (application de chiffrement).
- certificat : vise à effectuer un lien entre une personne et une bi-clé (privé/publique)
  - Il est délivré par une autorité de certification (AC)
  - Il est nominatif
  - Il est destiné à un usage unique (signature OU chiffrement)
  - Il a une durée de validité donnée
  - Il est certifié par l'AC
  - Il est révocable
- X.509-v3 : norme proposée par l'ISO (et la plus répandue)

3- Infrastructure de gestion des clés

Processus de certification



3- Infrastructure de gestion des clés

Certificats X 509

Objectif : Certifier qu'une clef publique appartient à une entité identifiée

Ensemble d'informations sur l'utilisateur (qq Ko) signé

- structure ASN.1
- protégé en intégrité
  - encodage DER
  - extension : .der, .cer, .crt, .cert

Typiquement :

- Clef publique
- Propriétaire de la clef : utilisateur, machine, serveur, équipement réseau...
  - qui possède la clef privée
- Durée de validité
- Information sur la signature (signataire/émetteur, algorithme de signature)
- la signature elle-même

3- Infrastructure de gestion des clés

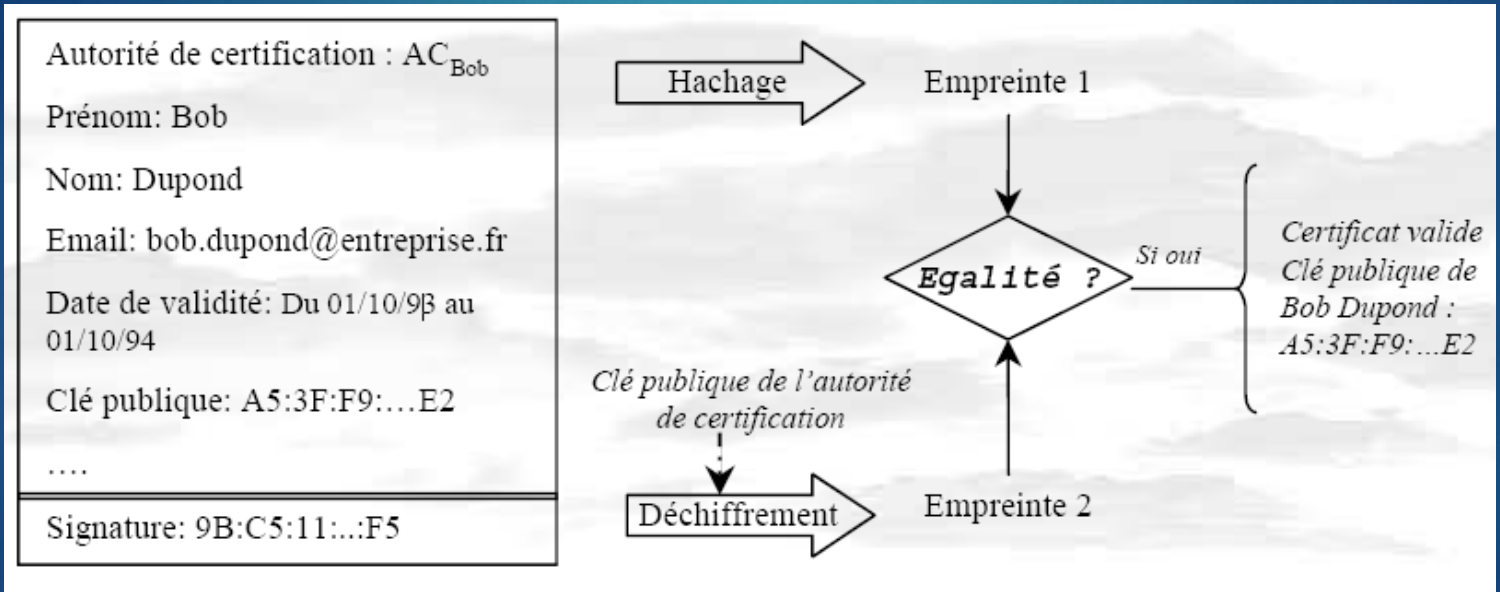
Certificats: exemple

Structure du certificat X.509v3 (1996)

Version du certificat
Numéro de série du certificat
Algo.de signature de l'AC
Nom de l'AC ayant délivré le certificat
Période de validité
Nom du propriétaire du certificat
Clé publique
Algo. à utiliser avec la clé publique
Identification de l'AC (opt)
Identification du propriétaire (opt)
Extensions (opt)
Signature de l'AC

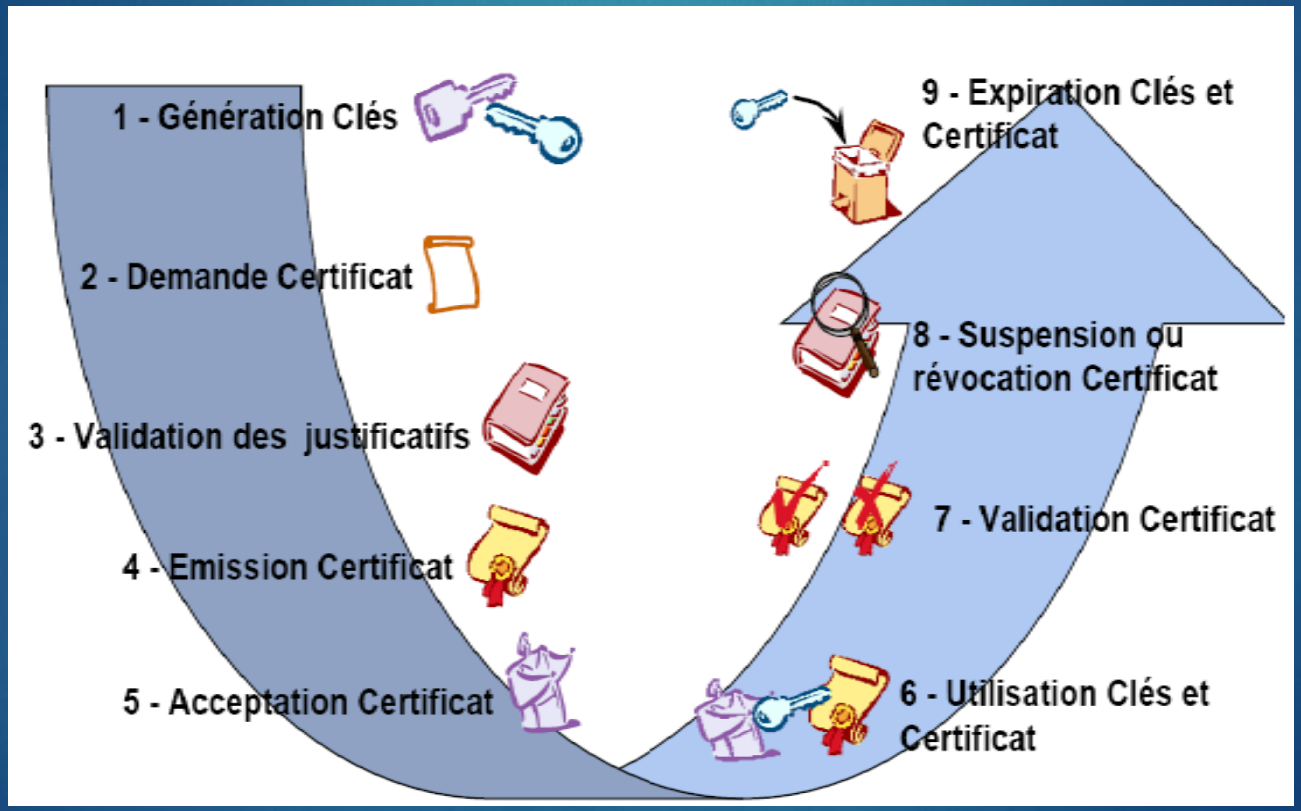
3- Infrastructure de gestion des clés

Certificats : vérification



3- Infrastructure de gestion des clés

Cycle de vie d'un certificat



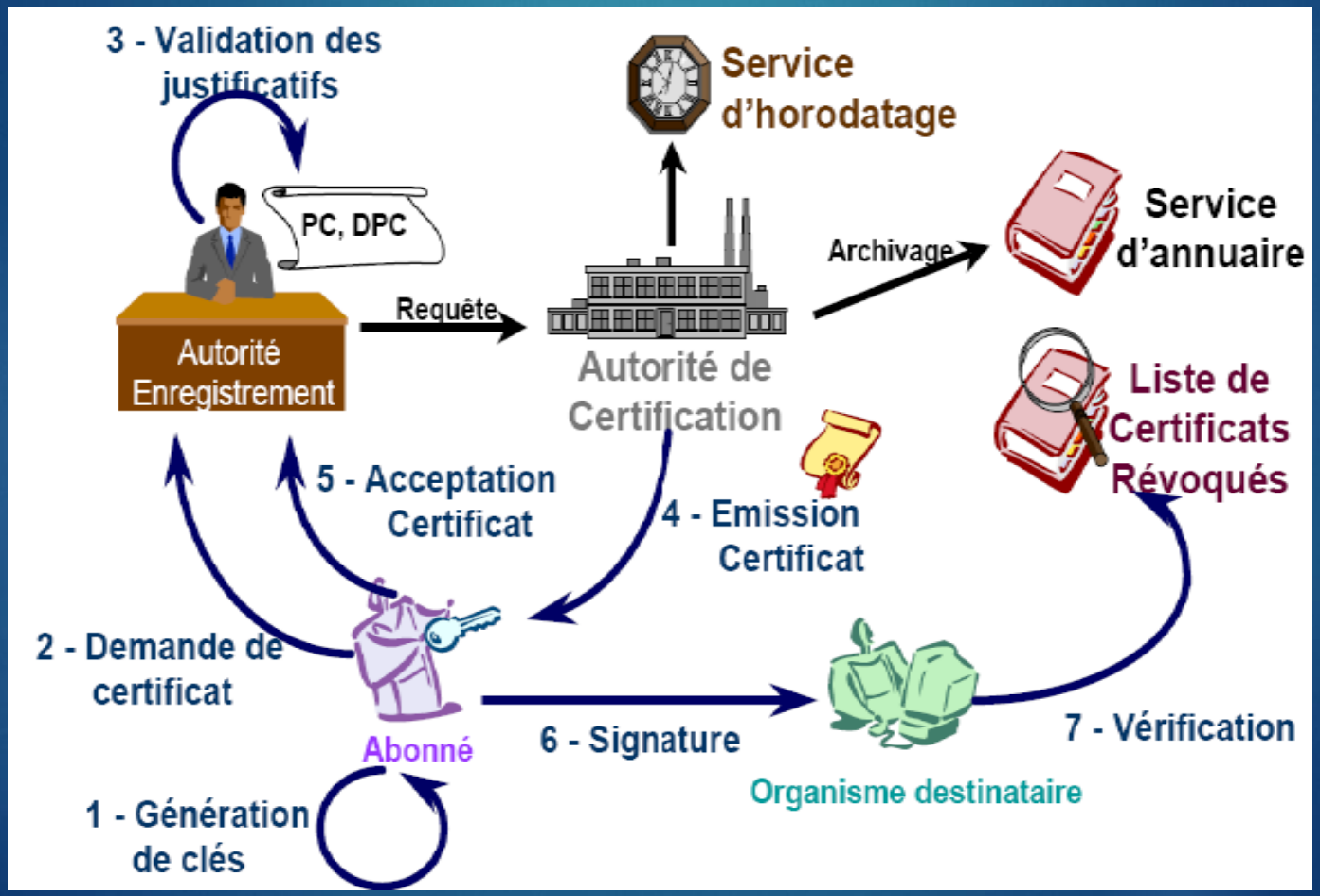
3- Infrastructure de gestion des clés

Infrastructure de gestion de clés

- IGC ou PKI (Public Key Infrastructure) recouvre les services mis en œuvre pour assurer la gestion des clés
  - enregistrement des utilisateurs
  - génération de certificats,
  - vérification des attributs,
  - publication des certificats valides et révoqués,
  - identification et authentification des utilisateurs,
  - archivage des certificats, etc.
- Composants fondamentaux d'une IGC
  - Autorité de certification ;
  - Autorité d'enregistrement,
  - Service de publication ou autorité de validation ;
  - Annuaire qui contient les clés publiques, les certificats distribués, ainsi que les listes de certificats révoqués.

3- Infrastructure de gestion des clés

étapes



3- Infrastructure de gestion des clés

Les acteurs de la certification

- Le porteur
  - Il est référencé par le certificat
  - Il est le seule à posséder la clé privée associée
- L'utilisateur
  - Il utilise le certificat
    - Il vérifie que l'identité indiquée par le certificat est bien son interlocuteur
    - Il vérifie que le certificat n'est pas révoqué (en consultant des listes des certificats révoqués - CRL)
    - Il vérifie que l'utilisation qu'il veut faire du certificat est conforme à son usage (chiffrement, signature, H)
    - authentifie et vérifie l'intégrité du certificat à l'aide de la clé publique de l'AC
- L'autorité de certification
  - Elle émet le certificat
    - Diffuse la valeur de sa clé publique auprès des structures qu'elle connait et des annuaires (e.g., LDAP « Lightweight Directory Access Protocole »)
  - Elle peut optionnellement créer des clés

3- Infrastructure de gestion des clés

Les acteurs de la certification

- L'autorité d'enregistrement
  - Elle dépend de l'AC
  - Elle s'occupe des aspects administratifs:
    - Vérification de l'identité du demandeur
    - Vérification que le demandeur est habilité à recevoir les droits indiqués dans le certificat
    - S'assure que celui possède bien un couple de clés privé-publique.
    - Traitement des demandes de révocation, suspension ou activation d'un certificat
    - Transmission de la demande à l'AC

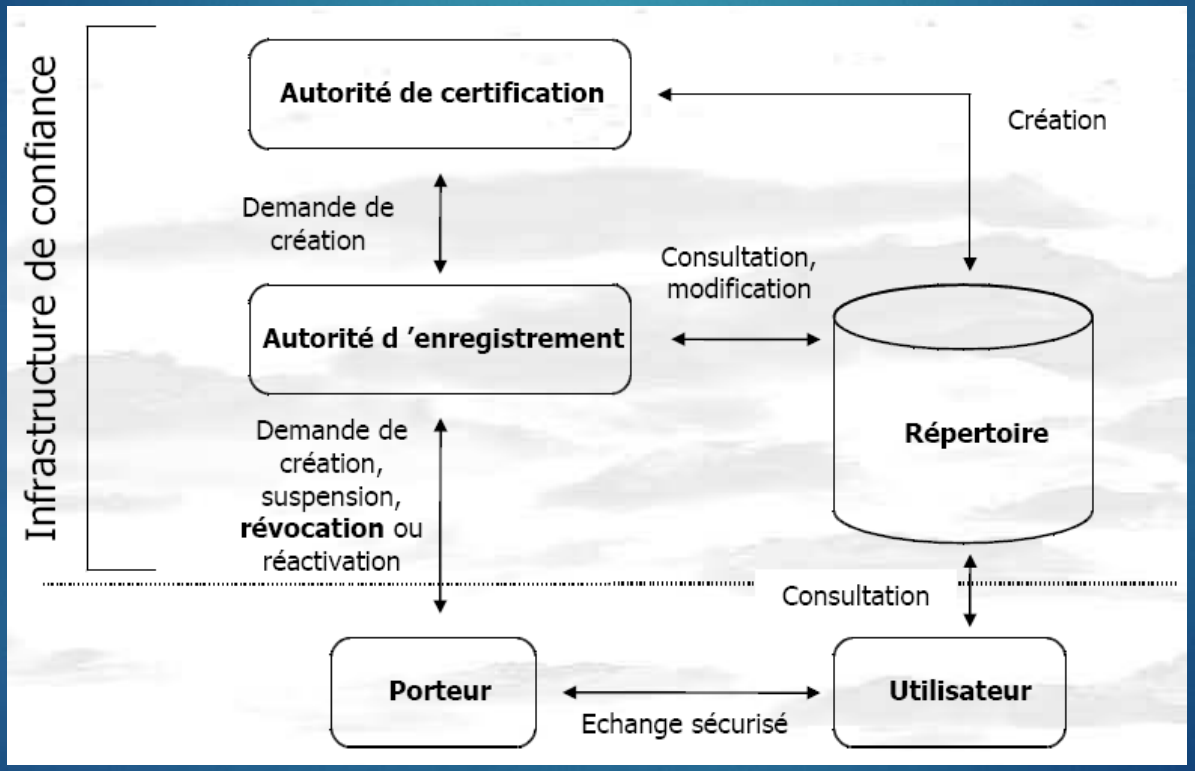
3- Infrastructure de gestion des clés

Les acteurs de la certification

- Le service de publication
  - Publie
    - les certificats
    - les CRL
  - Disponible à tous ceux qui font confiance à l'AC
  - Contraintes sur ce service:
    - à jour
    - Disponible
    - Intégré

3- Infrastructure de gestion des clés

Les acteurs de la certification



3- Infrastructure de gestion des clés

Autres composants

- Autorité d'horodatage (TA Timestamping Authority) – RFC 3161
  - Fournit un service de datation certifiée
  - Empêche les signatures antidatées
- Service de séquestre
  - Conservation des clefs privées de chiffrement
  - Sécurisé!!
- Générateur de bi-clef (KGS, Key Generator System)
  - contraintes cryptographique et de performance
  - Centralisé vs décentralisé
- L'autorité d'approbation des politiques
  - spécifie règles selon lesquelles l'AC est autorisée à délivrer des certs

3- Infrastructure de gestion des clés

La révocation du certificat

- Un certificat est révoqué car :
  - Il a expiré
  - Les clés secrètes ont été perdues ou compromises
  - Le porteur à fait un usage illégal du certificat
  - Il est non valide
- Chaque AC publie périodiquement une liste des certificats révoqués (CRL)

3- Infrastructure de gestion des clés

Organisation des PKI

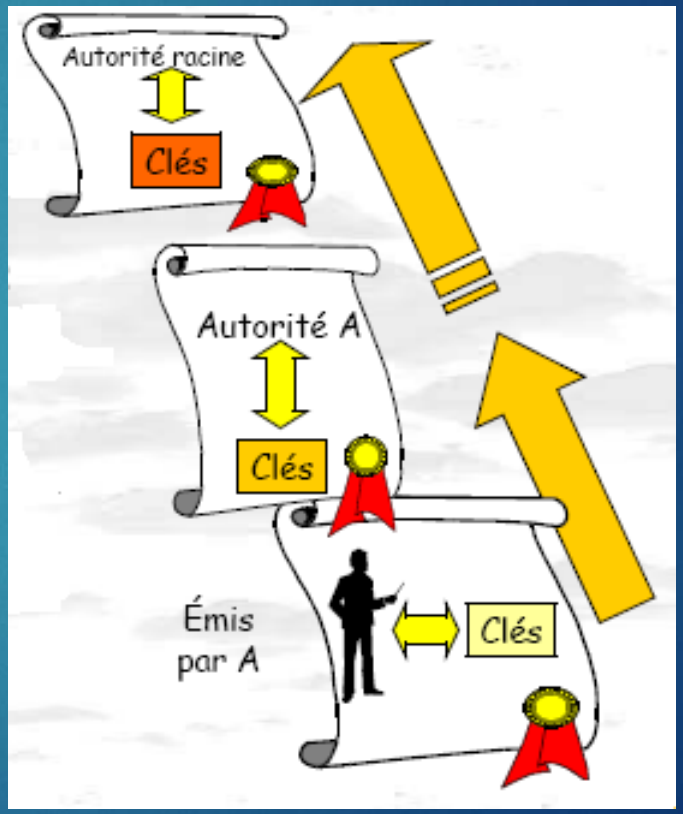
- le certificat ne peut garantir seul la validité d'une transaction, le sérieux de la PKI ayant délivré le certificat est important également

- Etant donné qu'il n'existe pas qu'une seule PKI, il arrive un moment ou une relation de confiance doit être établie entre les intervenants dépendant de plusieurs PKI

3- Infrastructure de gestion des clés

Chaîne de certification

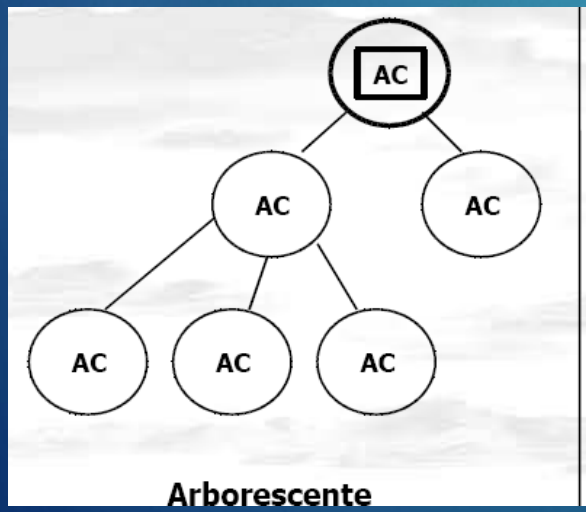
- La confiance est déplacée vers l'autorité de certification
- Des autorités peuvent en certifier d'autres : on aboutit à une chaîne de certification
  - à la racine, on a forcément un certificat "racine" auto-certifié …
  - si on ne fait pas confiance à cette autorité racine, toute la chaîne de certification ne sert à rien
  - généralement, il s'agit d'une organisation très réputée

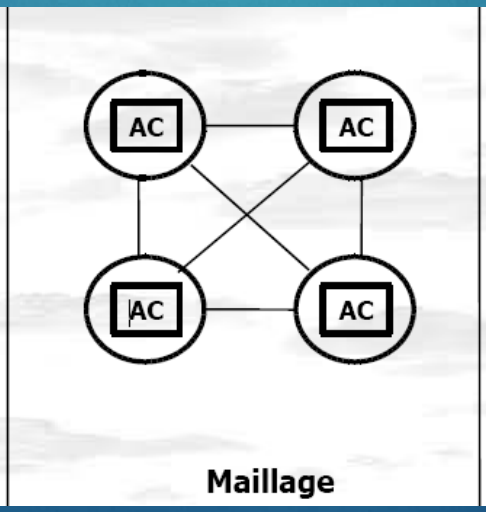


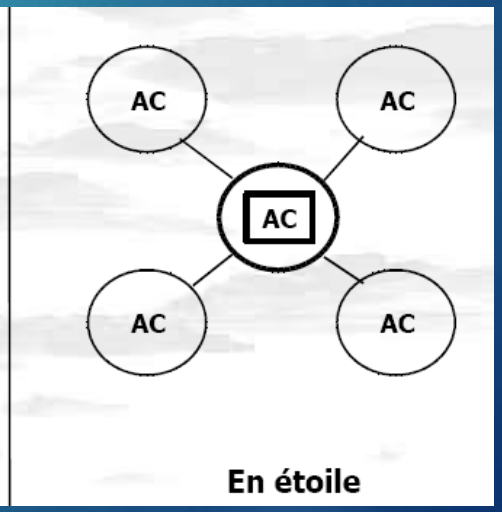
3- Infrastructure de gestion des clés

Organisation des PKI

- les AC peuvent être organisées de manière hiérarchique.. Plusieurs organisations sont possibles :
  - Organisation arborescente
  - Organisation en maillage
  - Organisation en étoile (avec point focal)







3- Infrastructure de gestion des clés

Organisation des PKI

- Exemple : IGC hiérarchique