Primitives cryptographiques: Chiffrement, Signature électronique et Hachage

Anca Nitulescu
anca.nitulescu@ens.fr

Ecole Normale Supérieure, Paris

Cours 1

Terminologie

Crypto

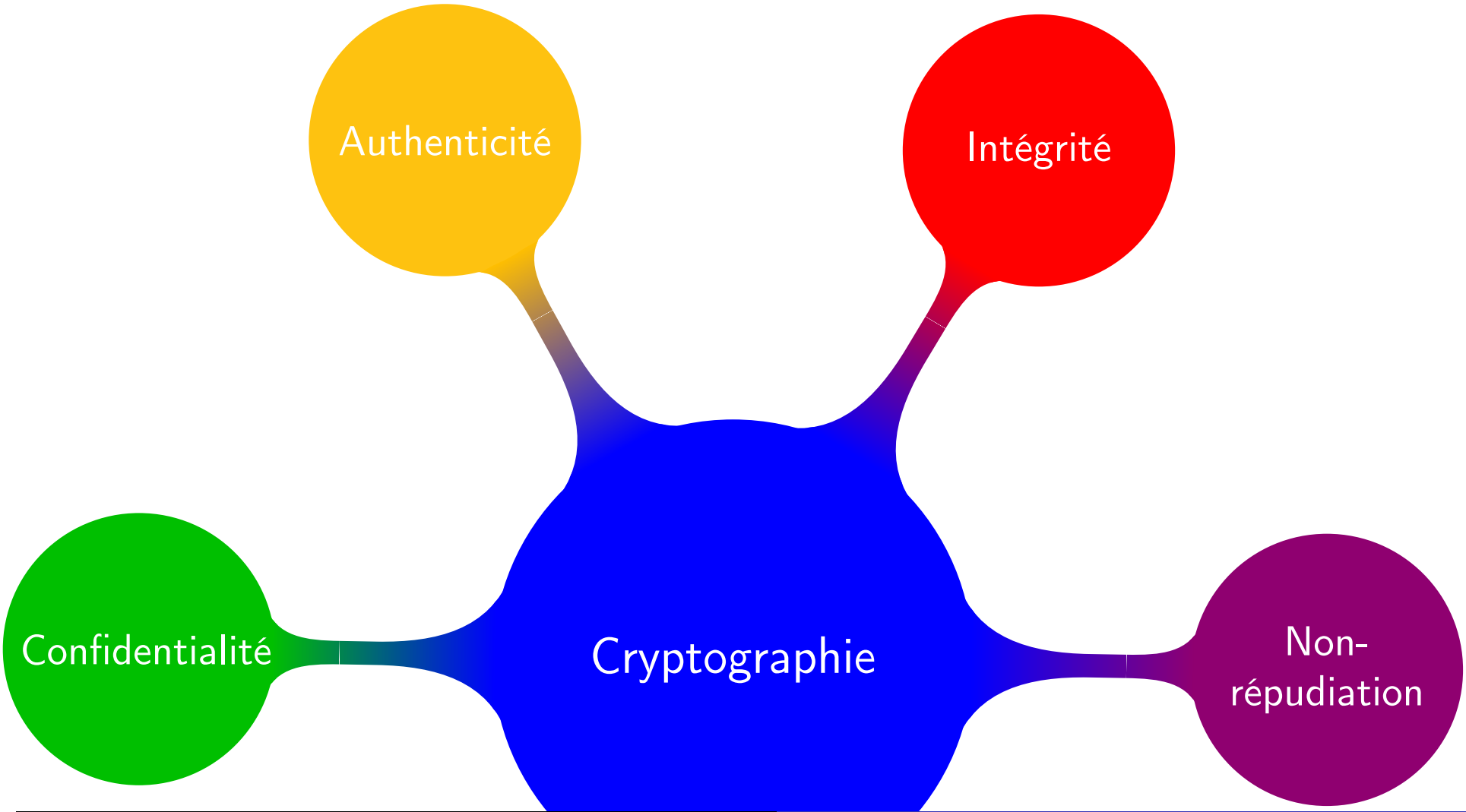
- **Cryptologie** = science du secret
- **Cryptographie** : l'art de rendre inintelligible, de crypter, de coder, un message pour ceux qui ne sont pas habilités à en prendre connaissance.
- **Cryptosystèmes** : mécanismes assurant le chiffrement des messages
- **Cryptanalyse** : art de "casser" des cryptosystèmes

Rappel Cours 1 et 2

Contexte

- **Données informatiques**
  - distribuées (réparties)
  - reliées en réseau
  - accessibles de partout
- **Utilisateurs / services**
  - nombreux utilisateurs, besoins variés
  - services plus ou moins critiques
  - utilisateurs plus ou moins sensibilisés
- **Environnement hostile**
  - menace diffuse
  - risques multiples

Propriétés


Authenticité
Intégrité
Confidentialité
Cryptographie
Non-
répudiation

Propriétés

Confidentialité

Le fait de s'assurer que l'information n'est seulement accessible qu'à ceux dont l'accès est autorisé

Propriétés

Confidentialité

Le fait de s'assurer que l'information n'est seulement accessible qu'à ceux dont l'accès est autorisé

Authenticité

Le fait de s'assurer que l'expéditeur est bien celui qu'il prétend être

Propriétés

Confidentialité

Le fait de s'assurer que l'information n'est seulement accessible qu'à ceux dont l'accès est autorisé

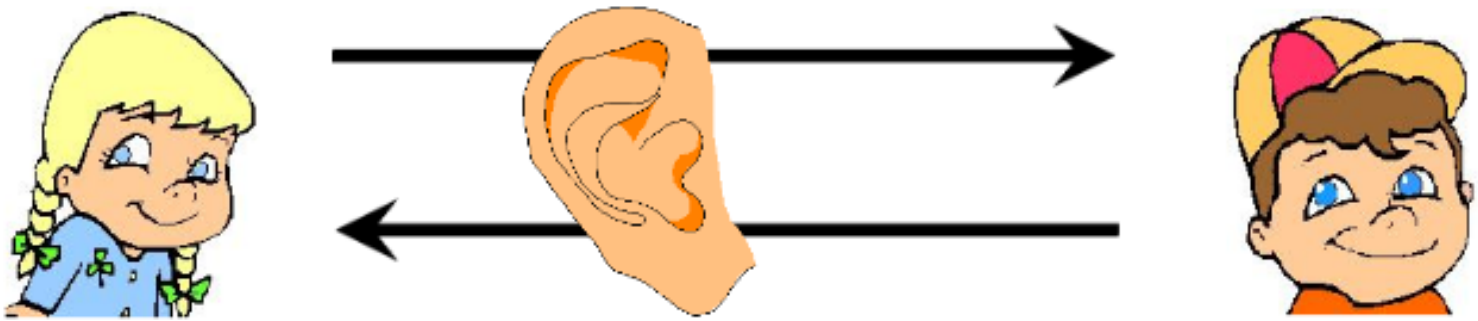
Authenticité

Le fait de s'assurer que l'expéditeur est bien celui qu'il prétend être

Intégrité

Le fait de s'assurer que l'information ne subisse aucune altération ou destruction volontaire ou accidentelle, et conserve le format initial

Confidentialité

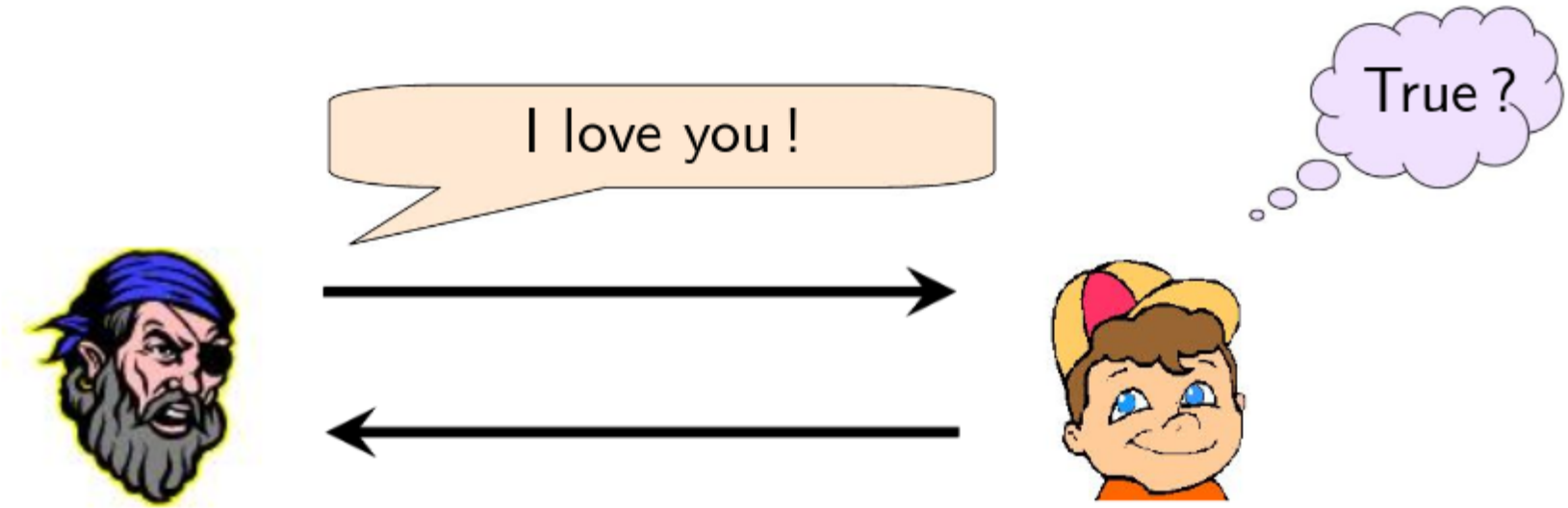


Contexte

S'assurer du caractère secret de l'information

- Echange de messages en présence d'un espion
- Stockage de données sécurisé

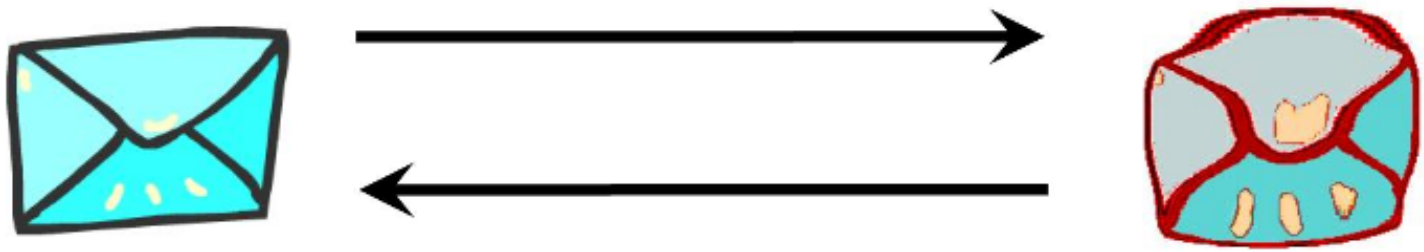
Authenticité



Contexte

S'assurer de la provenance d'un message et de l'authenticité de son émetteur

Intégrité



Contexte

S'assurer de la non-modification d'un message

Non-répudiation

d'origin

L'émetteur ne peut nier avoir écrit le message et il peut prouver qu'il ne l'a pas fait si c'est le cas

Non-répudiation

d'origin

L'émetteur ne peut nier avoir écrit le message et il peut prouver qu'il ne l'a pas fait si c'est le cas

de réception

Le receveur ne peut nier avoir reçu le message et il peut prouver qu'il ne l'a pas reçu si c'est effectivement le cas

Non-répudiation

d'origin

L'émetteur ne peut nier avoir écrit le message et il peut prouver qu'il ne l'a pas fait si c'est le cas

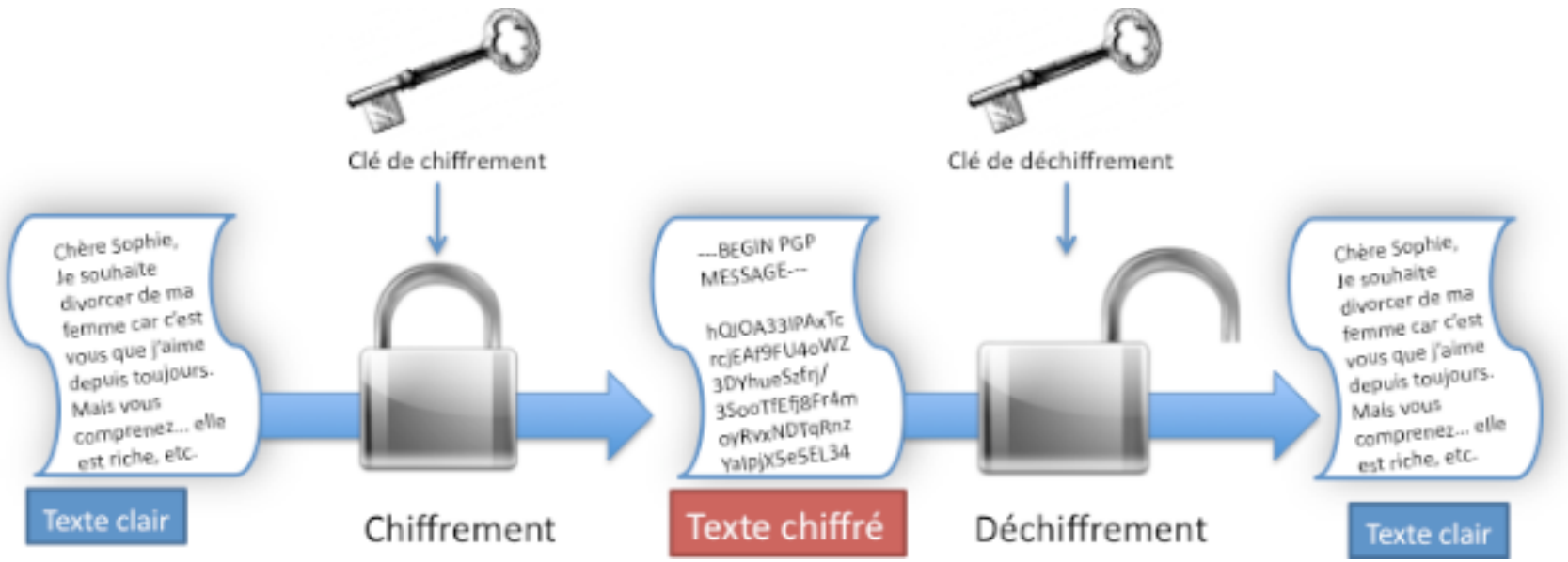
de réception

Le receveur ne peut nier avoir reçu le message et il peut prouver qu'il ne l'a pas reçu si c'est effectivement le cas

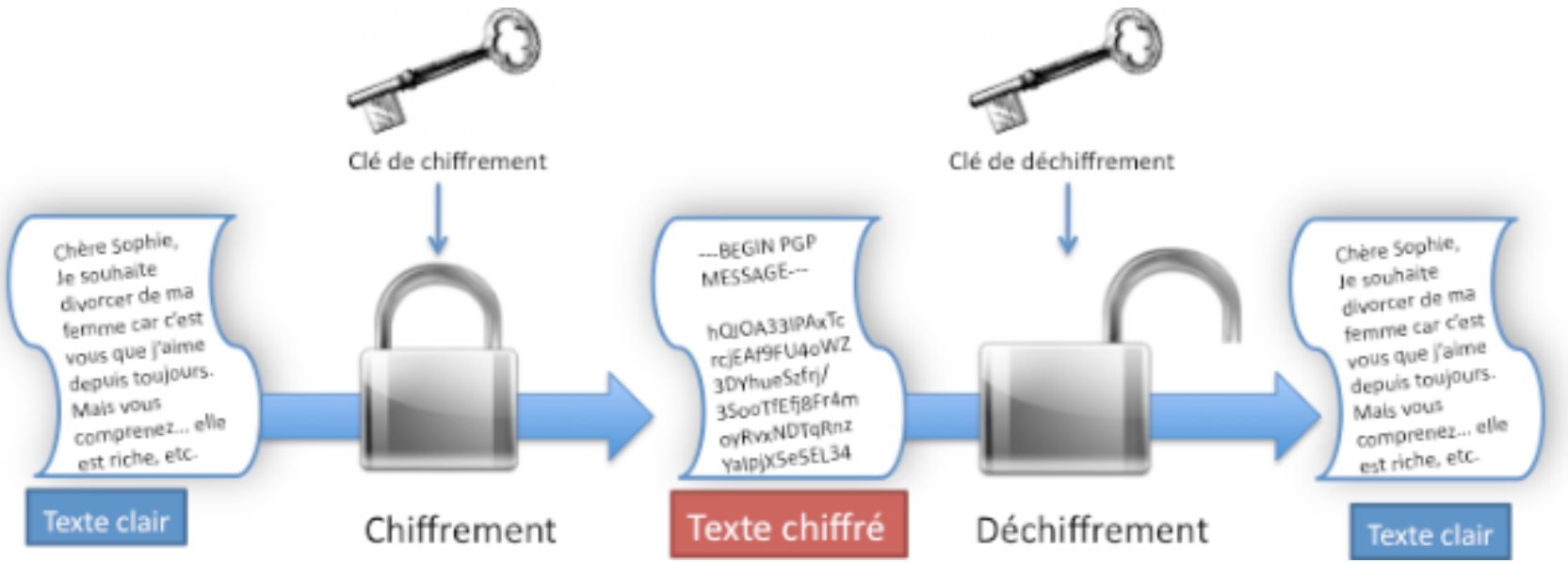
de transmission

L'émetteur du message ne peut nier avoir envoyé le message et il peut prouver qu'il ne l'a pas fait si c'est le cas

Cryptosystème



Cryptosystème



Définition

Un cryptosystème est un dictionnaire entre les messages en clair et les messages chiffrées.

Cryptosystème

Formalisation

Des ensembles finis

- $\mathcal{P}$ les mots en clair
- $\mathcal{C}$ les mots codés
- $\mathcal{K}$ les clefs

Des algorithmes

- $\mathcal{KG} : 1 \to \mathcal{K}$ générateur de clés
- $\mathcal{E} : \mathcal{K} \times \mathcal{P} \to \mathcal{C}$ chiffrement
- $\mathcal{D} : \mathcal{K} \times \mathcal{C} \to \mathcal{P}$ dechiffrement

Cryptosystème

Exemple

Le chiffrement de Caesar peut être représenté en utilisant les congruences sur les entiers.

$$A = 0, B = 1, C = 2, ..., Y = 24, Z = 25$$

Chiffre de Caesar

- $\mathcal{P} = \{0, 1, \ldots 25\}$
- $\mathcal{C} = \{0, 1, \ldots 25\}$
- $\mathcal{K} = \{0, 1, \ldots 25\}$
- Générer la clé = le décalage

$\mathcal{KG}(1) = 3$

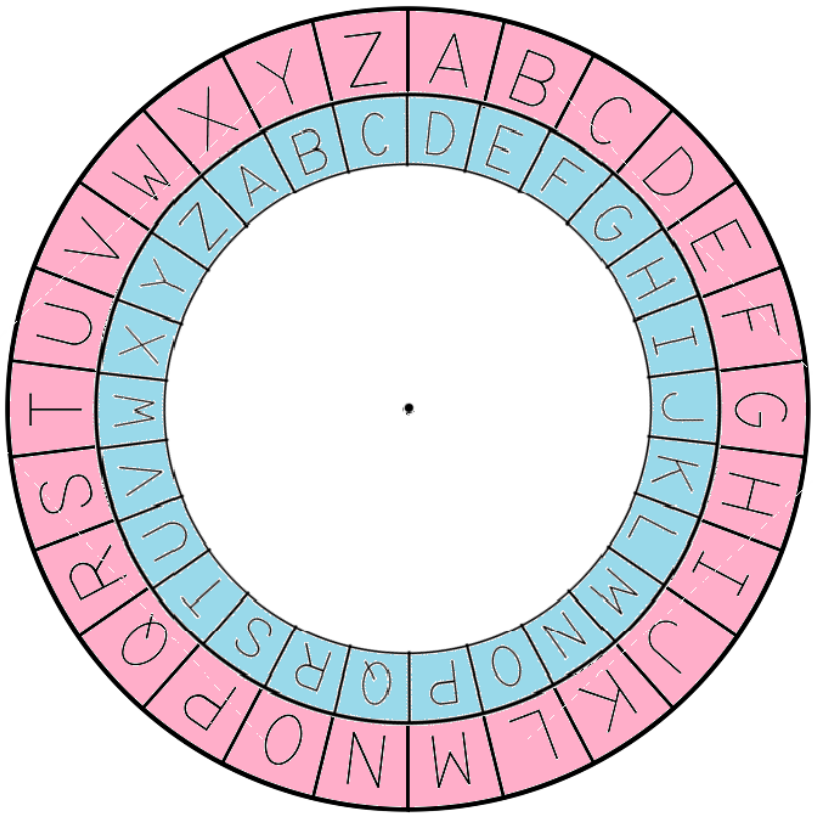
- Le chiffrement = la fonction qui ajoute 3 à chaque lettre du message

$\mathcal{E}(Y) = 24 + 3 = 1 = B \mod 26$

$\mathcal{E}(B) = 1 + 3 = 4 = E \mod 26$

- Le déchiffrement = la fonction qui soustrait 3

$\mathcal{D}(Z) = 25 - 3 = 22 = W \mod 26$



Cryptanalyse de chiffre de Caesar

Cryptanalyse

Recherche exhaustive :

Nombre faible de clés possibles (26), on les essaye toutes jusqu'à tomber sur la bonne

Faiblesse

Chaque lettre est remplacée par une autre, toujours la meme
L'ordre des lettres est conservé

Analyse des fréquences :

L'exploitation des caractéristiques linguistiques (redondances, fréquences) permet de cryptanalyser facilement ce type de schéma

Chiffrement à clé secrète

Déscription

- La clé de chiffrement est la meme que la clé de déchiffrement (ou dérivée).
- Alice et Bob doit utiliser une clé secrète pour chiffrer et déchiffrer les messages.
- L' échange d'une clé secrète entre Alice et Bob est nécessaire.

Chiffrement à clé secrète


Alice


Bob

1


Alice veut envoyer un message à Bob

2


Alice et Bob échangent une clé secrète

3


Alice va placer son message dans une boîte, qu'elle fermera à l'aide de la clé

4

Chiffrement à clé secrète

Avantages

- Systémes rapides (implantation matérielle)
- Clés relativement courtes (128 ou 256 bits)

Inconvénients

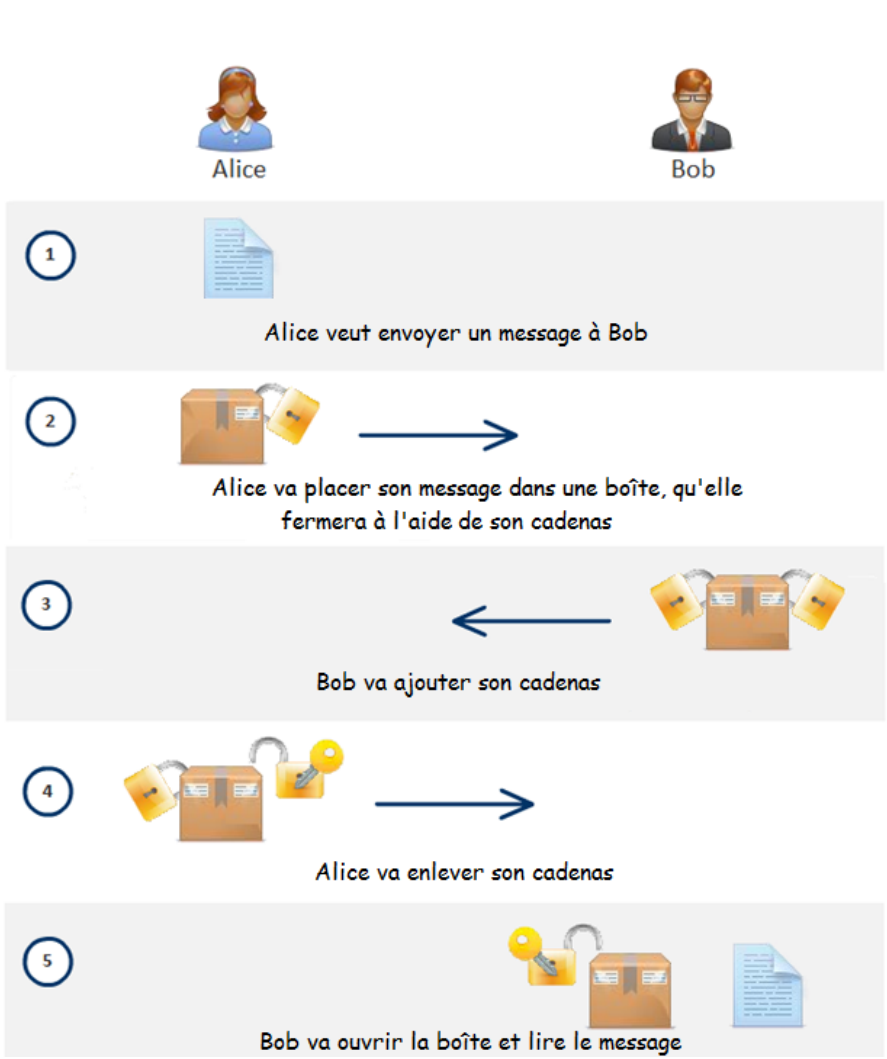
- Gestion des clés diffcile (nombreuses clés)
- Point faible = échange d'un secret

Chiffrement à clé secrète

Exemples

- **Chiffrement par blocs :**
  Les messages sont découpés en blocs
  - DES : blocs de 64 bits, clés de 56 bits
  - IDEA : blocs de 64 bits, clés de 128 bits
  - AES : blocs de 128 bits, clés de 128, 256 bits
- **Chiffrement par flots :**
  Les données sont traitées en flux
  - Pseudo-Vernam : on XOR un pseudo-aléa au flux
  - RC4 : chiffrement octet par octet

Chiffrement sans échange des clés ?



Difficulté

Construction impossible

Les deux clés (cadenas) **NE** commutent **PAS** !

(On doit respecter l'ordre du chiffrement)

Chiffrement à clé publique


Alice


Bob

1


Alice veut envoyer un message à Bob

2


Bob va d'abord envoyer à Alice un cadenas ouvert, dont lui seul possède la clé

3


Alice va placer son message dans une boîte, qu'elle fermera à l'aide de ce cadenas

4

Chiffrement à clé publique

Principe d'asymétrie

Un système cryptographie à clé publique est en fait basé sur deux clés :

- **Une clé publique :**

La clé de chiffrement peut être distribuée librement (comme un cadenas ouvert)

- **Une clé privée :**

La clé de déchiffrement est connue uniquement du receveur, (la clé sécrete qui ouvre le cadenas fermé)

Chiffrement à clé publique

Formalisation

Trois algorithmes

- **Algorithme de génération des clés** $\mathcal{KG}(\ell) = (\mathsf{pk}, \mathsf{sk})$

à partir d'un paramètre de sécurité, il produit une paire de clés

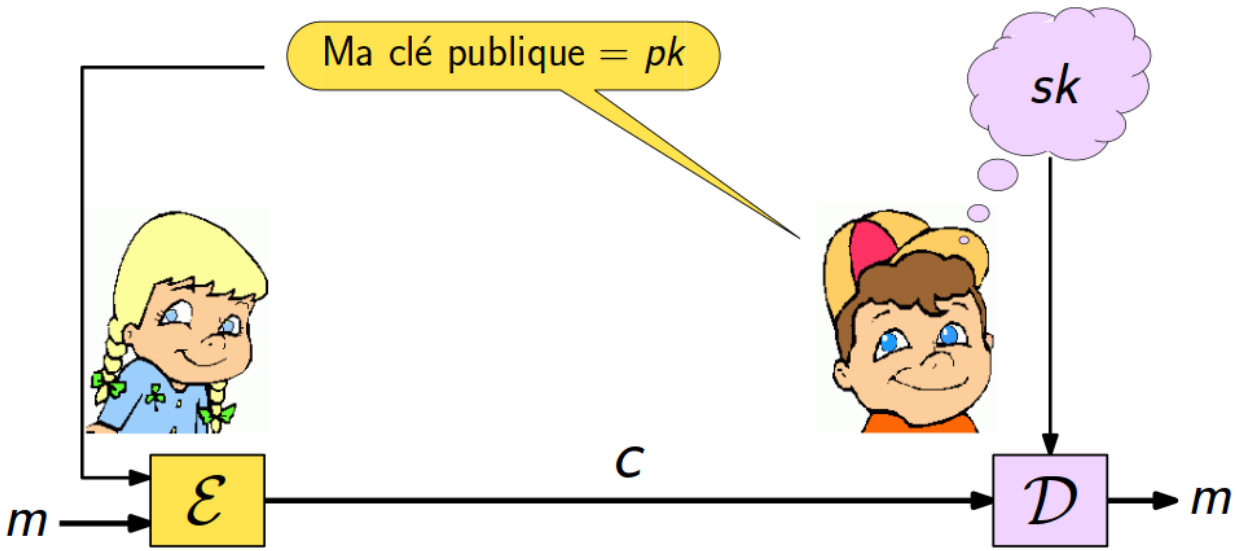
- **Algorithme de chiffrement** $\mathcal{E}(\mathsf{pk}, m) = c$

produit le chiffré d'un message m, par la clé publique

- **Algorithme de déchiffrement** $\mathcal{D}(\mathsf{sk}, c) = m$

utilise la clé sécrete/privée sk pour retrouver m à partir de c

Chiffrement à clé publique



Rémarques

- La clé publique ne permet pas de déchiffrer (en particulier, elle ne permet pas de retrouver la clé privée).
- N'importe qui peut chiffrer un message pour Bob, sans besoin de communication antérieure.

Chiffrement à clé publique

Avantages

La clé privée ne quitte pas son propriétaire :

 Gestion des secrets facilitée

 Pas de secret à transmettre

Inconvénients

La relation clé publique/clé privée impose :

 Des clés plus longues (1024 à 4096 bits)

 Des cryptosystèmes beaucoup plus lents qu'en symétrique

 Gestion de certiffcats de clés publiques

Chiffrement à clé publique

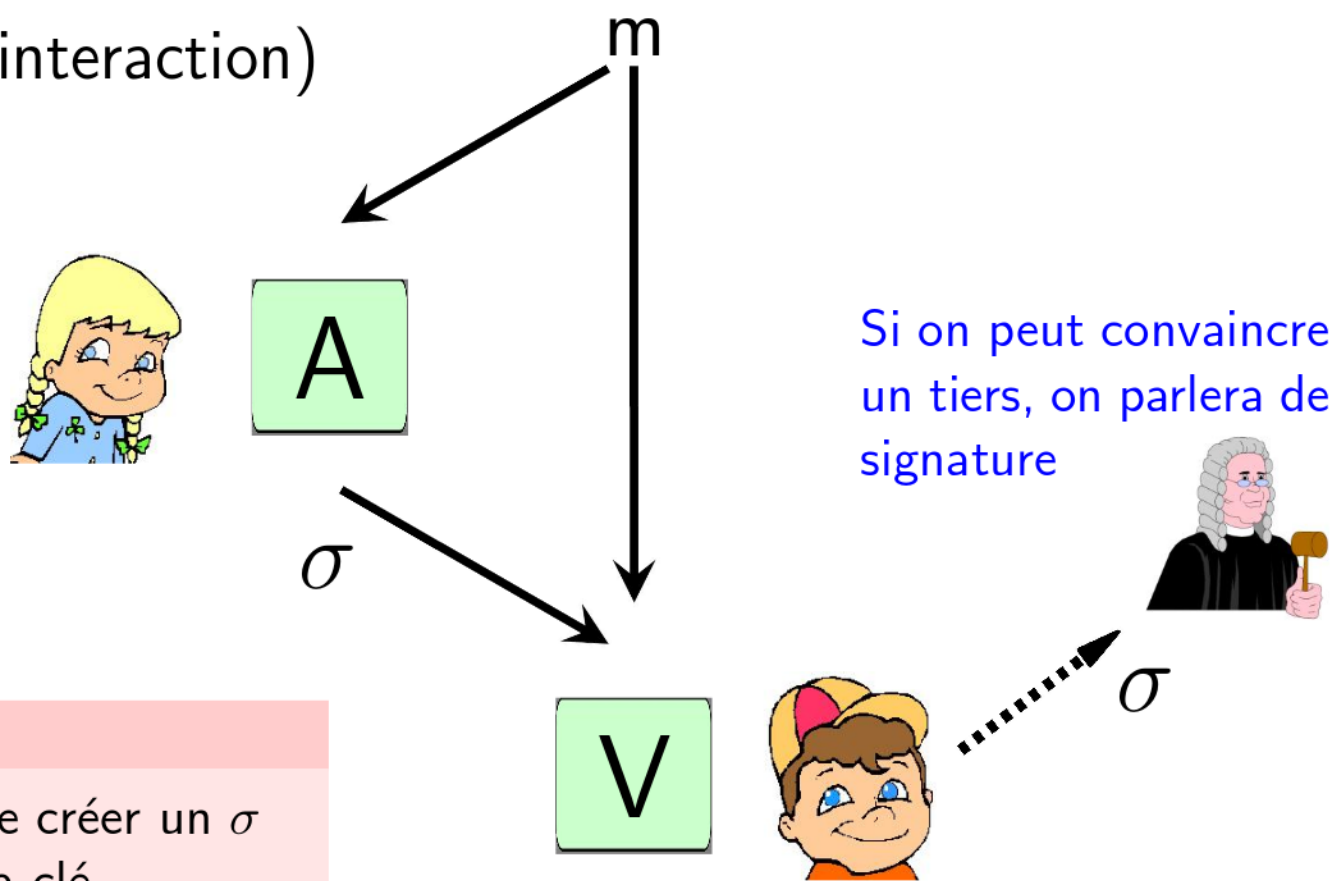
Exemples

- **RSA (Rivest-Shamir-Adleman, 1978) :**
  basé sur les racines modulaires et la decomposition en facteurs prémiers
- **ElGamal (1984) :**
  basé sur le logarithme discret
- **McEliece (1978) :**
  basé sur les codes correcteurs
- **Merkle-Hellman (1978) :**
  basé sur des problèmes combinatoires (sac-à-dos)
- **Hidden Field Equation (Patarin, 1996) :**
  basé sur les systèmes multivariables

Signature électronique

Lier un message à son auteur

(sans interaction)



Si on peut convaincre un tiers, on parlera de signature

Sécurité

Impossible de créer un σ valide sans la clé

Signature électronique : idée générale

Propriétés

Reproduire les caractéristiques d'une signature manuscrite :

- Lier un document à son autheur
- Rendre la signature difficilement imitable
- Responsabilité de l'auteur (juridique)

Les différents acteurs

Le Signataire

Doit pouvoir facilement émettre des signatures en son nom

Les différents acteurs

Le Signataire

Doit pouvoir facilement émettre des signatures en son nom

Le vérifieur

Doit pouvoir vérifier la signature sans avoir d'information confidentielle
Potentiellement tout le monde doit pouvoir vérifier.

Les différents acteurs

Le Signataire

Doit pouvoir facilement émettre des signatures en son nom

Le vérifieur

Doit pouvoir vérifier la signature sans avoir d'information confidentielle
Potentiellement tout le monde doit pouvoir vérifier.

L'ennemi

Il cherche à générer une fausse signature

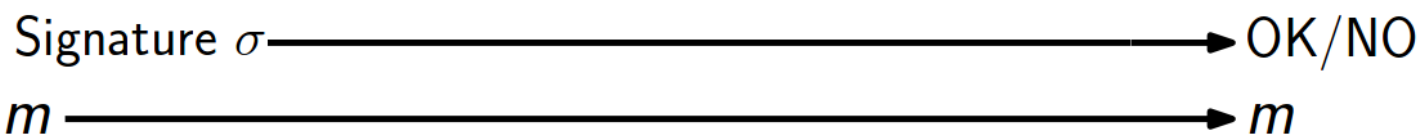
Fonctionnement

Le Signataire

Emet une signature et l'envoie avec le message

Signataire

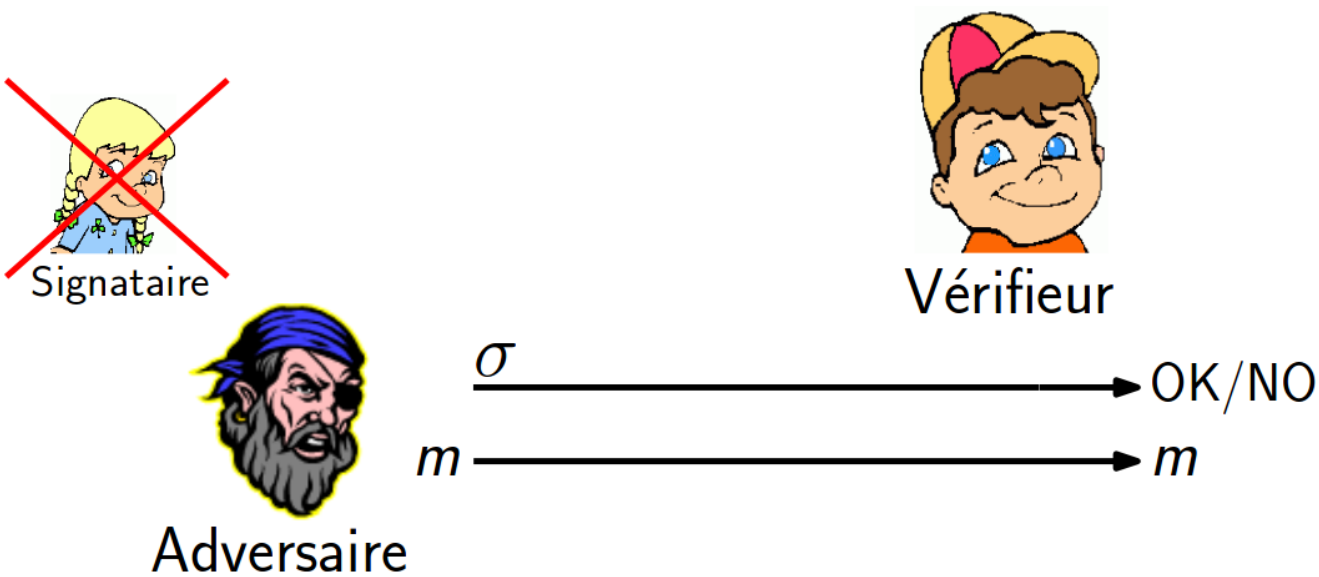
Vérifieur



Fonctionnement

L'adversaire

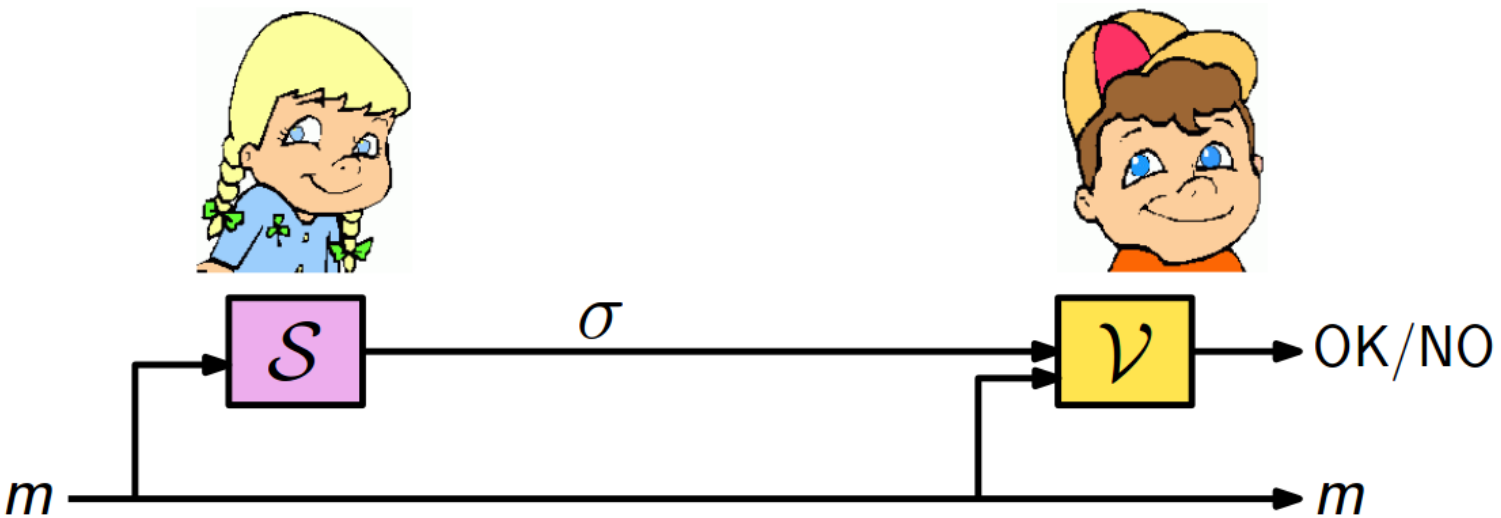
Cherche à fabriquer de fausses signatures



Fonctionnement

Algorithmes

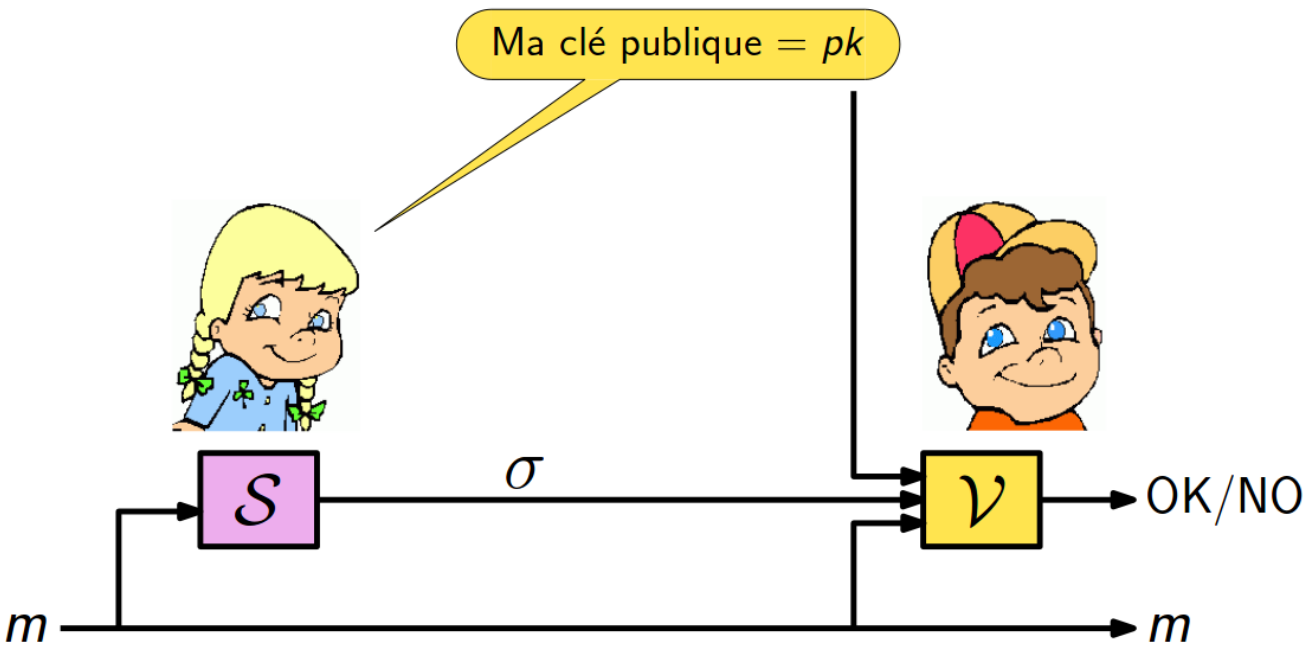
Algorithmes de **signature** et de **vérification**



Fonctionnement

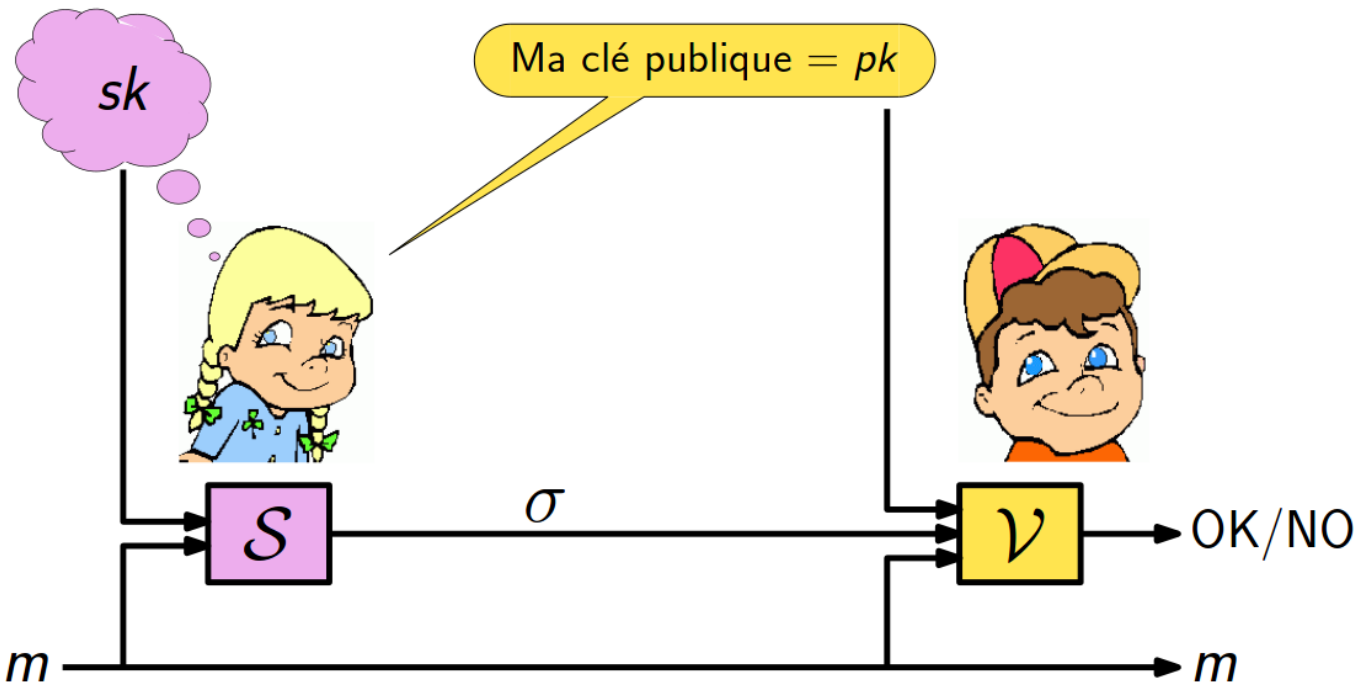
Le vérifieur

Tout le monde peut vérifier : Mécanisme à **clé publique** !



Fonctionnement

La **clé privée** est utilisée pour signer.



Signatures électroniques

Formalisation

Trois algorithmes

- **Algorithme de génération des clés** $\mathcal{KG}(\ell) = (\mathsf{pk}, \mathsf{sk})$

à partir d'un paramètre de sécurité, il produit un couple de clés : clé publique, clé sécrete

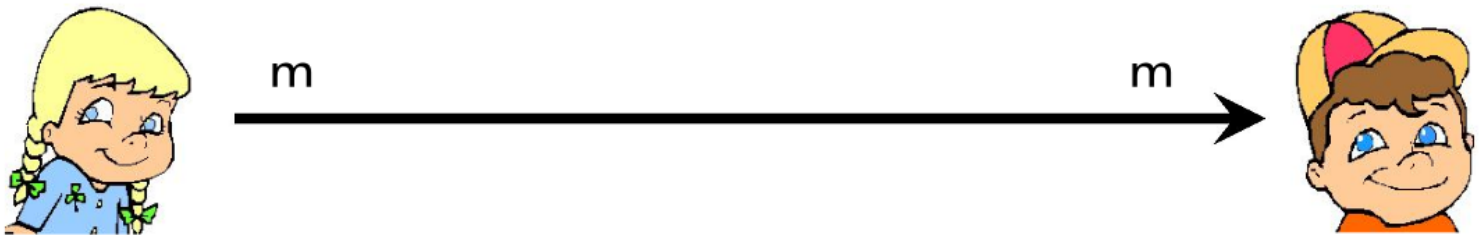
- **Algorithme de signature** $\mathcal{S}(\mathsf{sk}, m) = \sigma$

utilise la clé sécrete sk pour signer le message

- **Algorithme de vérification** $\mathcal{V}(\mathsf{pk}, m, \sigma) = \mathit{yes}/\mathit{no}$

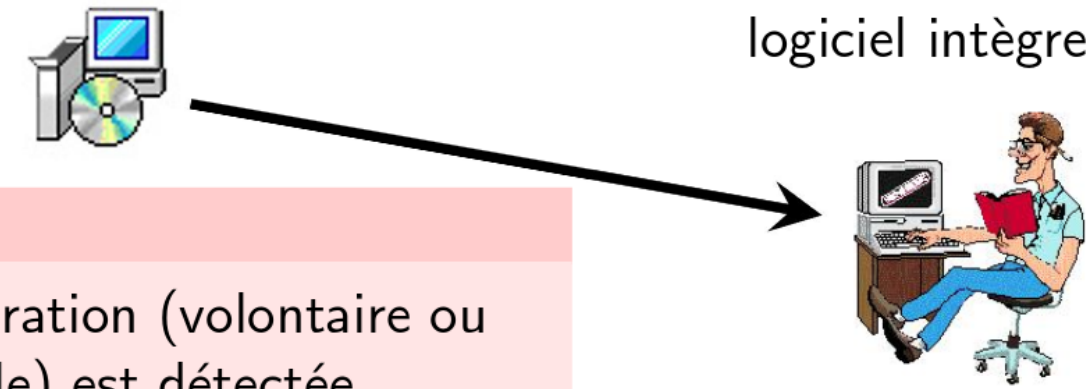
utilise la clé publique seulement

Idée générale



Non modification du message pendant sa transmission

Téléchargement d'un logiciel intègre



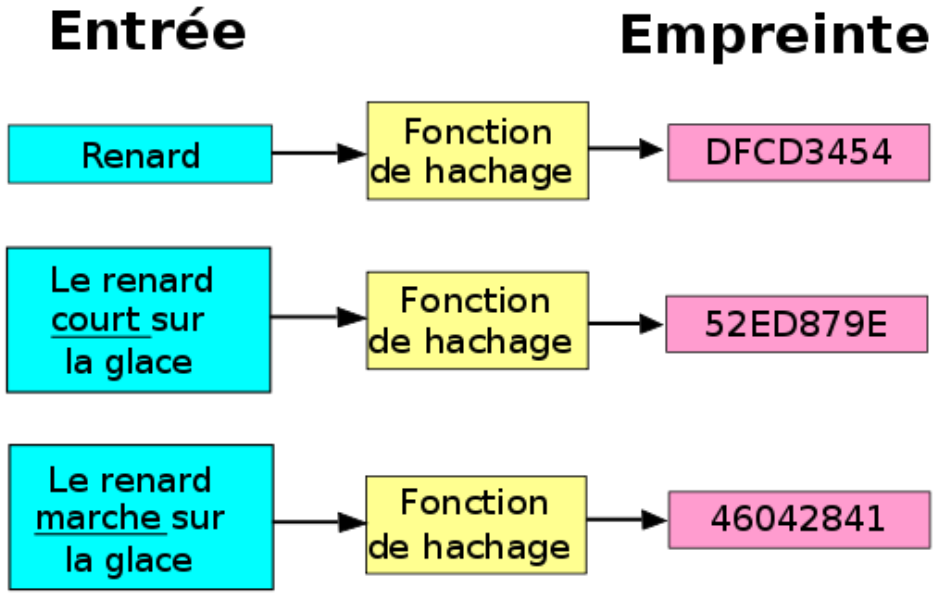
Sécurité

Toute altération (volontaire ou accidentelle) est détectée

But

L'intégrité des données

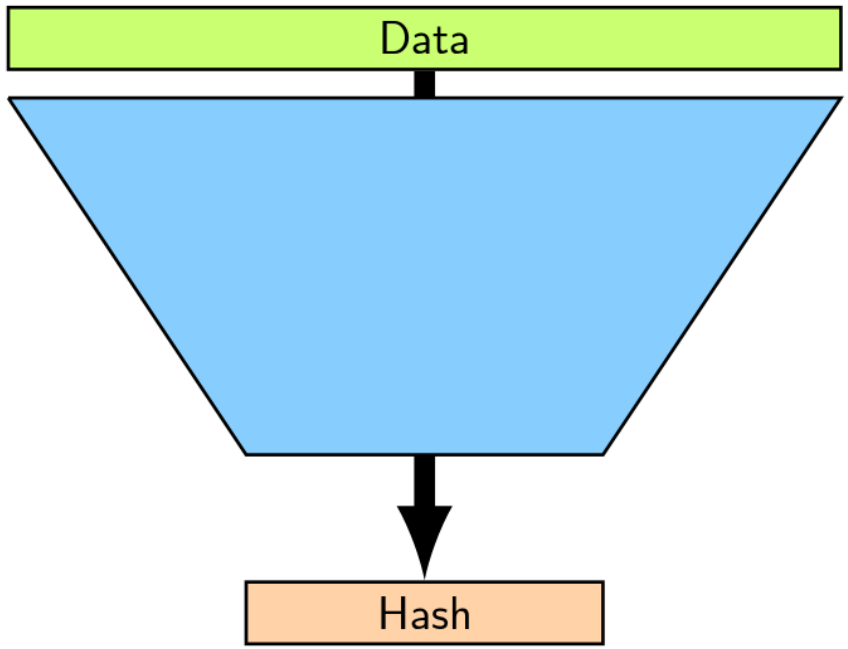
- Vérifier que les infos transmises n'ont pas subi d'altérations.
- Produire une empreinte condensée qui soit facilement vérifiable.



Définition

Fonction de hachage

Une fonction qui, à partir d'une donnée arbitraire, calcule une empreinte servant à condenser et identifier rapidement la donnée initiale



Fonctions de hachage

Propriétés

- **Compression et facilité de calcul.**
- **Résistance à la préimage :**
  étant donné y, il est difficile de trouver x tel que $y = H(x)$
- **Résistance à la seconde préimage :**
  étant donné x, il est difficile de trouver $x' \neq x$ tel que $H(x) = H(x')$
- **Résistance à la collision :**
  il est difficile de trouver x et x' tels que $H(x) = H(x')$

Fonctions de hachage

Types des fonctions

- **Fonction de Hachage à Sens Unique**
  résistance à la préimage et à la seconde préimage
- **Fonction de Hachage résistante aux collisions**
  résistance à la seconde préimage et à la collision

Fonctions de hachage

Types des fonctions

- **Fonction de Hachage à Sens Unique**
  résistance à la préimage et à la seconde préimage
- **Fonction de Hachage résistante aux collisions**
  résistance à la seconde préimage et à la collision

Fonctions de hachage résistante aux collisions

Absence de collisions : permet à garantir qu'une altération des données produira une modification de l'empreinte

Fonctions de hachage

Exemples

- **MD5 (Message Digest 5) :**
  Empreinte de 128 bits
- **SHA (Secure Hash Algorithm) :**
  - Norme NIST
  - Empreinte de 160 bits
- **SHA-1 (révision publiée en 1994)**
  Considéré comme plus sûr que MD5
- **SHA-2 (octobre 2000)**
  Agrandit la taille de l'empreinte