

# Introduction à la Sécurité

Licence 2 – Sem4  
2020

# Mécanismes de sécurité

## Le chiffrement

- Le cryptage utilise spécifiquement l'algorithme appelé chiffrement pour crypter et décrypter le message
- Le chiffrement est une série d'étapes bien défini qui peut être suivi comme une procédure de cryptage ou de décryptage des messages
- Il existe différentes méthodes :
  - La transposition
  - La substitution
  - Masque Jetable

# Mécanismes de sécurité

## Le chiffrement

- Le Chiffrement permet de garantir la **confidentialité des données**
- Différents types d'algorithmes:

- **Algorithme symétrique ou à clé secrète:**

Rapide, utilisation de clé prépartagée

- **Algorithme asymétrique ou à clé publique:**

Utilisation de deux clés différentes (une pour chiffrer et une autre pour déchiffrer)

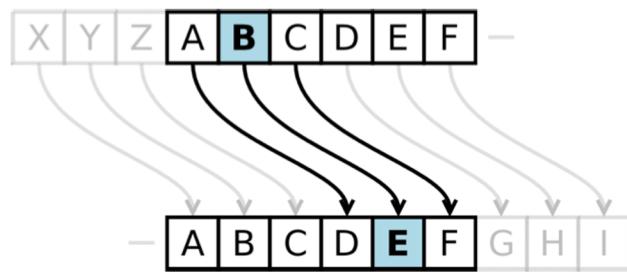
Echange de clé secrètes

Signature

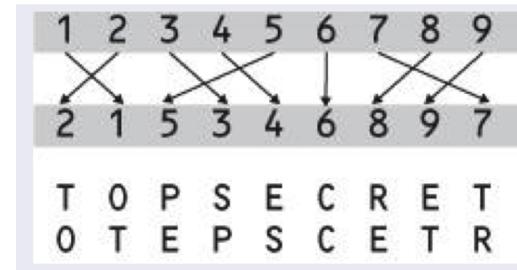
# Mécanismes de sécurité

## Le chiffrement

Chiffrement de césar par substitution



Chiffrement par transposition



## Vigenère ou multi-décalage

On associe aux lettres un entier dans  $\{0, \dots, 25\}$  et on fixe un mot  $C = (c_1, \dots, c_l)$  de  $l$  lettres pour être la clé secrète .

On découpe le message à envoyer en blocs  $B_i = (x_{(i,1)}, x_{(i,2)}, \dots, x_{(i,l)})$  de  $l$  lettres . Le chiffrement/déchiffrement par un multi-décalage sur les  $B_i$ :

# Mécanismes de sécurité

## Le chiffrement symétrique

- Le Chiffrement est dit symétrique ou à clé secrète si pour chiffrer et déchiffrer, la même clé secrète est utilisée.
- Soit  $P$  et  $C$  les alphabets clairs et chiffrés.
- $K$  l'ensemble des clés possibles.

$k \in K ; e_k : P \rightarrow C$  et  $d_k : C \rightarrow P$  avec  $\forall x \in P \ d_k(e_k(x)) = x$ .

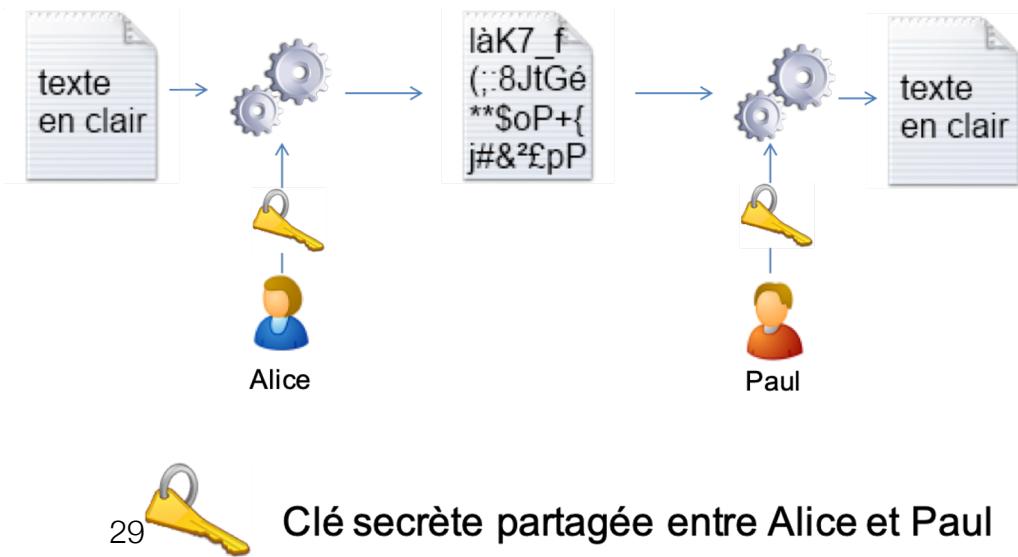
- Les applications  $e_k$  et  $d_k$  nécessitent la connaissance complète de  $K$  pour être définies.

# Mécanismes de sécurité

## Le chiffrement symétrique

- Les algorithmes de chiffrement symétriques utilisent la même clé pour chiffrer et déchiffrer les données. Ils partent du principe que chacune des parties engagées dans la communication connaît la clé prépartagée. Ils sont généralement utilisés pour le trafic VPN.

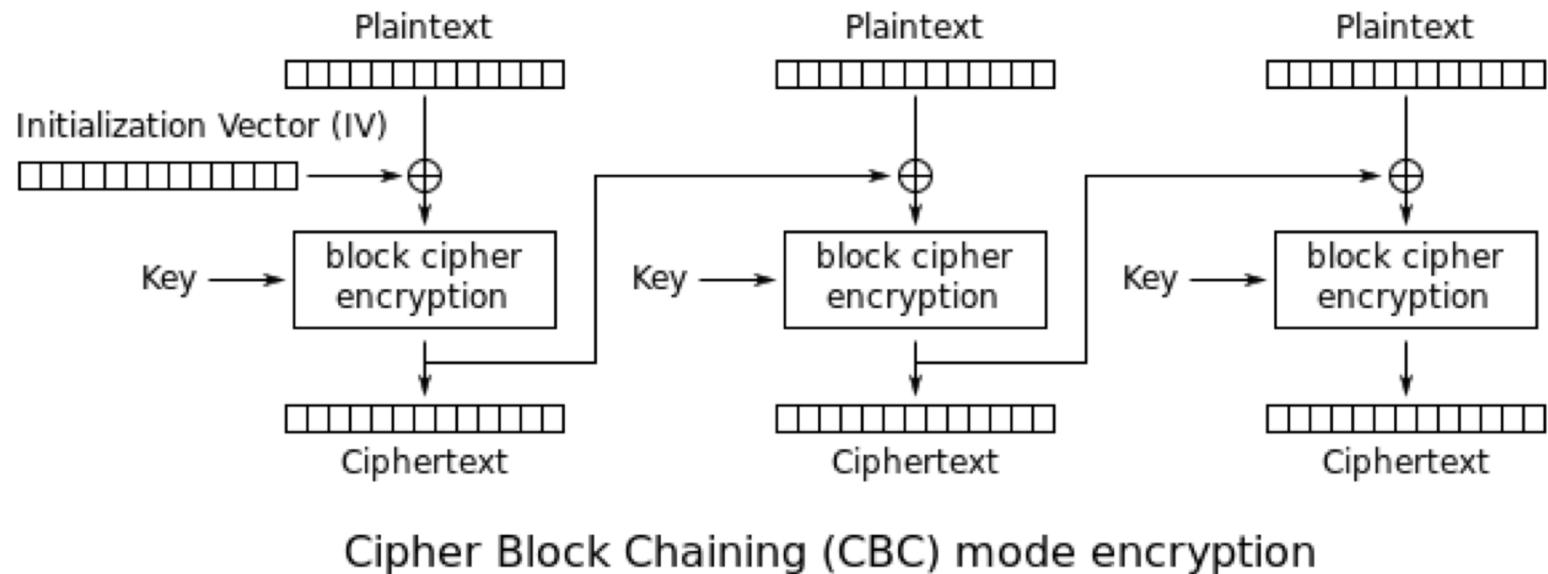
Exemples d'algorithmes de chiffrement symétriques DES, 3DES, AES, IDEA, RC2/4/5/6, et Blowfish.



# Mécanismes de sécurité

## Le chiffrement symétrique

- Mode opératoire



# Mécanismes de sécurité

## Le chiffrement asymétrique

- Plus connus comme algorithmes à clé publique.
- L'émetteur utilise une clé de chiffrement différente de la clé utilisée par le récepteur pour le déchiffrement
- La longueur de la clé habituelle est de 512 à 4096 bits.
- Algorithmes très lents pour une utilisation intensive car basés sur des algorithmes de calcul complexe
- Exemples d'algorithmes de chiffrement asymétriques RSA, ElGamal, elliptic curves, and Diffie-Hellman.

# Mécanismes de sécurité

## Le chiffrement asymétrique

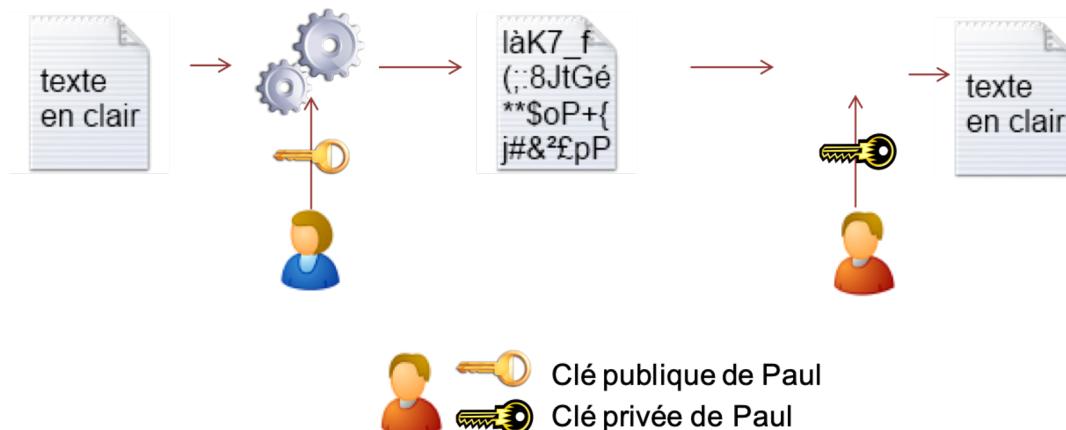
- La clé utilisée pour le chiffrement est différente de celle utilisée pour le déchiffrement. Il est nécessaire d'utiliser 2 clés :
  - **Clé publique** : comme son nom l'indique, cette clé est publique et peut être donnée à tout le monde ;
  - **Clé privée** : cette clé doit être personnelle et connue de son seul propriétaire. Elle ne doit jamais être divulguée !
- Ces deux clés sont mathématiquement liées
  - La connaissance de la clé publique ne permet pas de calculer de manière efficace la clé privée (attention à la taille de la clé, qui doit être suffisamment longue) ;
  - Chaque personne doit donc posséder 2 clés : une clé privée (confidentielle) et une clé publique qu'il peut divulguer à tout le monde.

# Mécanismes de sécurité

## Le chiffrement asymétrique

Exemple : Alice souhaite envoyer un message confidentiel à Paul

- Alice chiffre le message avec la clé publique de Paul ;
- Paul déchiffre le message grâce à sa privée ;
- Notes :
  - Alice ne pourra jamais (et n'aura jamais besoin de) utiliser la clé privée de Paul puisque celle-ci est confidentielle à Paul !
  - Alice n'a pas besoin d'utiliser ses clés personnelles dans cet exemple de chiffrement sans signature



# Mécanismes de sécurité

## Le chiffrement symétrique vs asymétrique

### Chiffrement symétrique

- Rapidité des opérations (adapté à du trafic en temps réel) ;
- Clés courtes (256 bits suffisent actuellement) ;

### Avantages

- Difficulté d'échange sécurisé des clés secrètes : comment le faire en protégeant ce secret ?

### Chiffrement asymétrique

- Facilité d'échange des clés : les seules clés qui ont besoin d'être échangées sont des clés publiques (dont il faut assurer la protection en intégrité) ;

### Inconvénients

- Lenteur des opérations (peu adapté à du trafic en temps réel) ;
- Grande taille des clés (2048 bits minimum actuellement) ;

### Exemples d'algorithmes sûrs (janvier 2015)

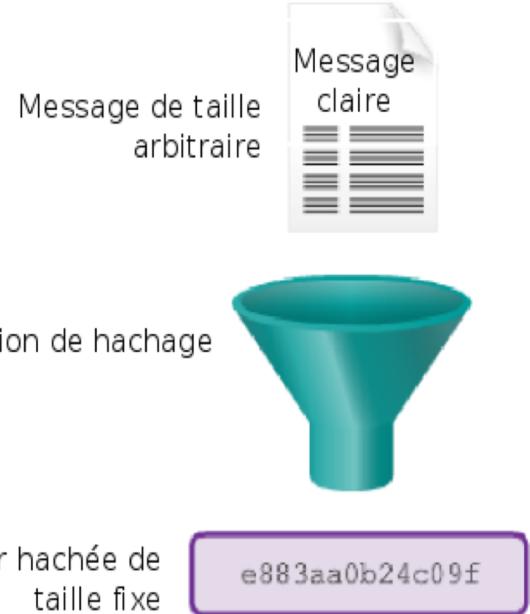
- AES.

- RSA.

# Mécanismes de sécurité

## Le Hachage

- Les **algorithmes de hachage** permet d'assurer L'intégrité et l'authenticité des messages
- L'algorithme de hachage prend un message donné et génère un condensé appelé HASH ou empreinte
- L'algorithme de hachage est basée sur une **fonction mathématique à sens unique** très facile à calculer mais beaucoup plus difficile à inverser
- L'utilisation de **signatures numériques** et de certificats avec les contrôles d'intégrité des données offre aux utilisateurs un moyen de vérifier **l'authenticité et la non-répudiation** des messages et des documents



# Mécanismes de sécurité

## Le Hachage

### Propriétés du hache ou empreinte

- Il n'y a pas de limite de longueur pour le texte : Message de taille arbitraire
- La longueur du résultat est fixe.
- La fonction de hash est unidirectionnelle et irréversible Fonction de hachage
- Deux valeurs d'entrée différentes donneront toujours des valeurs de hash différentes .



valeur hachée de taille fixe e883aa0b24c09f

# Mécanismes de sécurité

## Le Hachage

Les algorithmes de hachage sont appliquées dans différentes situations:

- Fournit une preuve d'authenticité quand il est utilisé avec une clé secrète symétrique d'authentification, tels que la sécurité IP (IPsec) ou l'authentification du protocole de routage
- Fournit une authentification en générant des réponses ponctuelles et à sens unique dans les protocoles d'authentification, tels que le CHAP PPP.
- Fournit une preuve de vérification de l'intégrité des messages, tels que ceux acceptés lors de l'accès à un site sécurisé à l'aide d'un navigateur.
- Confirme que le fichier téléchargé n'a pas été modifié.

# Mécanismes de sécurité

## Le Hachage

Les algorithmes de hachage:

- **Algorithme Message Digest 5 (MD5)**

Développé par Ron Rivest

Produit une valeur de hash de 128 bits.

- **Algorithme Secure Hash** ( SHA-1) développé par le NIST (National Institute of Standards and Technology)

Produit une valeur de hash de 160 bits

Remplacé par SHA-2 en ajoutant 04 fonctions de HASH

La taille du hash est indiqué par le suffixe:

- SHA-224    SHA-256
- SHA-384    SHA-512

# Mécanismes de sécurité

## Le Hachage

### Salage

**Inconvénient du Hash** : Si deux utilisateurs possèdent le même mot de passe, ils auront également les mêmes hashs de mot de passe.

Une valeur de salage, qui correspond à une chaîne aléatoire de caractères, est une entrée supplémentaire du mot de passe avant le hash.

Elle crée un résultat de hash différent pour les deux mots de passe, comme illustré sur la figure. Une base de données stocke le hash et la valeur de salage.

salage	Valeur du hash
Hash (« mot de passe » + <b>QxLUF1blAdeQX</b> )	= <b>b3bab1e5324f057753a4b8d7cef293e4</b>
Hash (« mot de passe » + <b>R9PeIC7sxQXb8</b> )	= <b>713c7beb54841a26a7c81eb06d6cf066</b>

## Mécanismes de sécurité

### Le Hachage

**Hash Message Authentication Code (HMAC)** renforcent les algorithmes de hash en utilisant une clé secrète supplémentaire en entrée de la fonction hash.

HMAC permet d'assurer l'authenticité des messages (Authentification et intégrité des messages)

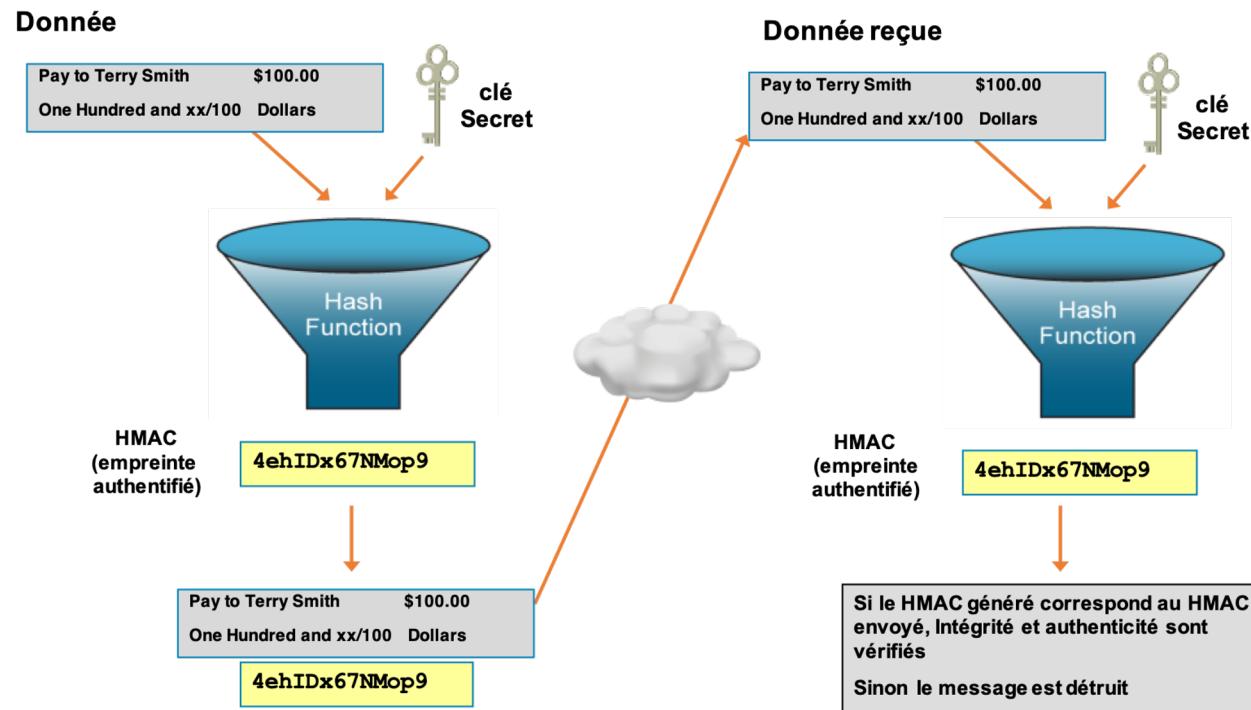
Algorithmes HMAC :

- HMAC-MD5
- HMAC-SHA1

# Mécanismes de sécurité

## Le Hachage

### Hash Message Authentication Code (HMAC)



# Mécanismes de sécurité

## Signature électronique

- Rappel de l'objectif : s'assurer de l'intégrité d'une donnée, et s'assurer de l'identité de son auteur. Si la signature n'est pas valide, cela indique que l'auteur « n'est pas le bon » ou que la donnée reçue n'est pas celle que son auteur avait signé.
- Dans de nombreux pays, les signatures numériques ont la même valeur légale qu'un document signé manuellement.

Notes :

- La signature électronique n'assure pas la confidentialité des données, mais leur intégrité et la notion de preuve ;
- Lorsque l'on chiffre un message, il est fortement recommandé de le signer également afin d'assurer l'intégrité du message.

# Mécanismes de sécurité

## Signature électronique -Principe

1.Le signataire d'un message génère – grâce à un algorithme cryptographique spécifique – une valeur unique calculée à partir du message que l'on souhaite signer : un condensat (un haché) ;

- Les algorithmes de calcul de condensat sont publics et ne gèrent pas de secret, donc tout le monde peut les utiliser et calculer les mêmes condensats à partir d'un même message ;
- Deux messages différents ne peuvent pas donner lieu au même condensat.

2.Le signataire utilise l'algorithme de signature, qui prend en entrée sa clé privée et le condensat précédent, pour produire une signature électronique ;

3.Le signataire envoie (ou stocke) le message et la signature électronique, permettant ainsi à un lecteur d'en prendre connaissance ;

4.Le lecteur calcule lui-même le condensat du message en clair ;

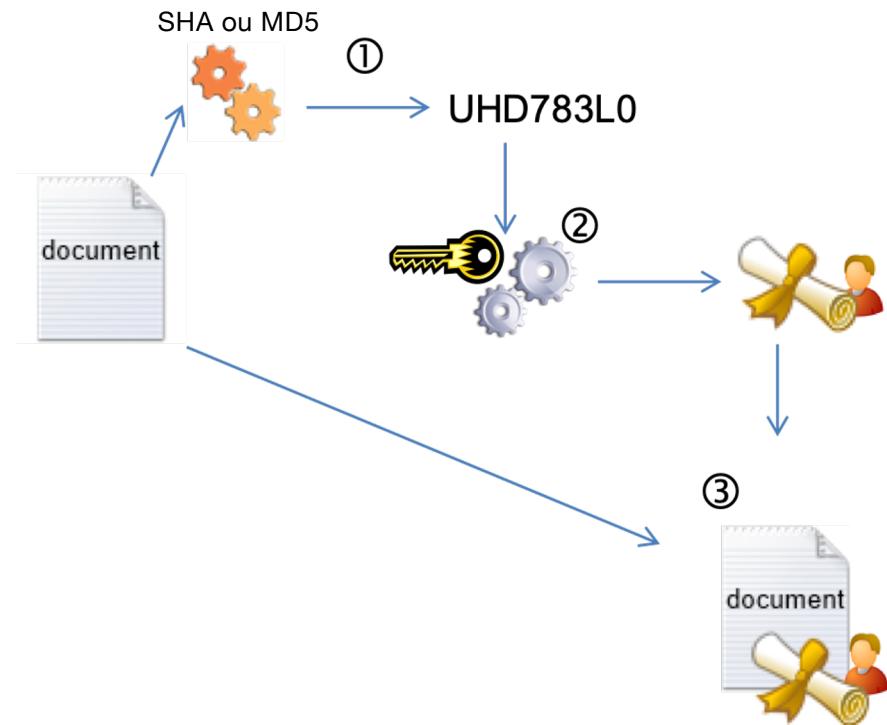
5.Le lecteur utilise l'algorithme de vérification de signature, qui prend en entrée la clé publique du signataire, le condensat et la signature, pour rendre un verdict. Si le verdict est négatif, alors il ne faut pas faire confiance au message reçu (celui-ci ne correspond pas – pour une raison que l'on ignore – au message du signataire).

# Mécanismes de sécurité

## Signature électronique -Illustration

### Etapes de la signature :

1. Le signataire génère le condensat unique associé au message ;
2. Le signataire utilise l'algorithme de signature, qui prend en entrée sa clé privée et le condensat précédent, pour produire une signature électronique ;
3. Le signataire envoie (ou stocke) le message et la signature électronique, permettant ainsi à un lecteur d'en prendre connaissance ;
4. La vérification par le destinataire/lecteur est décrite sur la diapositive suivante.



# Mécanismes de sécurité

## Signature électronique -Illustration

### Etapes de la vérification de la signature par un lecteur/destinataire :

4. Le lecteur calcule le condensat du message en clair ;

5. Le lecteur utilise l'algorithme de vérification de signature, qui prend en entrée la clé publique du signataire, le condensat et la signature, pour rendre un verdict. Si le verdict est négatif, alors il ne faut pas faire confiance au message reçu (celui-ci ne correspond pas — pour une raison que l'on ignore — au message du signataire).

