

# Cryptographie

---

# Les risques réseau

- ♦ *Spoofing* ou *masquarade*
  - Se faire passer pour quelqu'un d'autre
  - Possible dès qu'il y a une association effectuée dynamiquement :
    - ✧ adresse physique-IP
    - ✧ adresse IP-nom
    - ✧ redirection ICMP - routage dynamique en général
- ♦ Attaque active

# Les risques réseau

---

- *Sniffing*
  - Prendre connaissance du contenu des données transitant sur le réseau
  - Utilisation du *spoofing*
  - Ethernet classique diffusion des données
  - Mot de passe en clair sur le réseau pour la plupart des protocoles
- Attaque passive

# Les risques réseau

---

- ♦ *Denial Of Service*
  - Empêcher l'accès à un service
  - Utilisation de toutes les ressources
    - ✧ Ping broadcasté
    - ✧ TCP SYN flood
    - ✧ Message fragmenté avec morceau manquant

# Les risques réseau

---

- Accès, utilisation ou modification de ressources non autorisées
- Destruction de données

# Les services de sécurité

---

- Confidentialité
  - protection des données émises sur le réseau
- Authentification
  - les données reçues proviennent bien de l'entité déclarée
- Intégrité
  - les données reçues n'ont subi aucune modification au cours du transfert

# Les services de sécurité

---

- Contrôle d'accès
  - protège l'accès aux données
- Non répudiation
  - il est possible de prouver que les données ont bien été émises par l'émetteur ou reçues par le récepteur

# Les services de sécurité

---

- Prévention du rejeu de données
  - les données reçues ne sont pas des données précédemment reçues
- Confidentialité du trafic
  - il n'est pas possible de connaître des propriétés sur le trafic :
    - ✧ longueur des données
    - ✧ horaires d'émission
    - ✧ en-tête commune
    - ✧ etc...



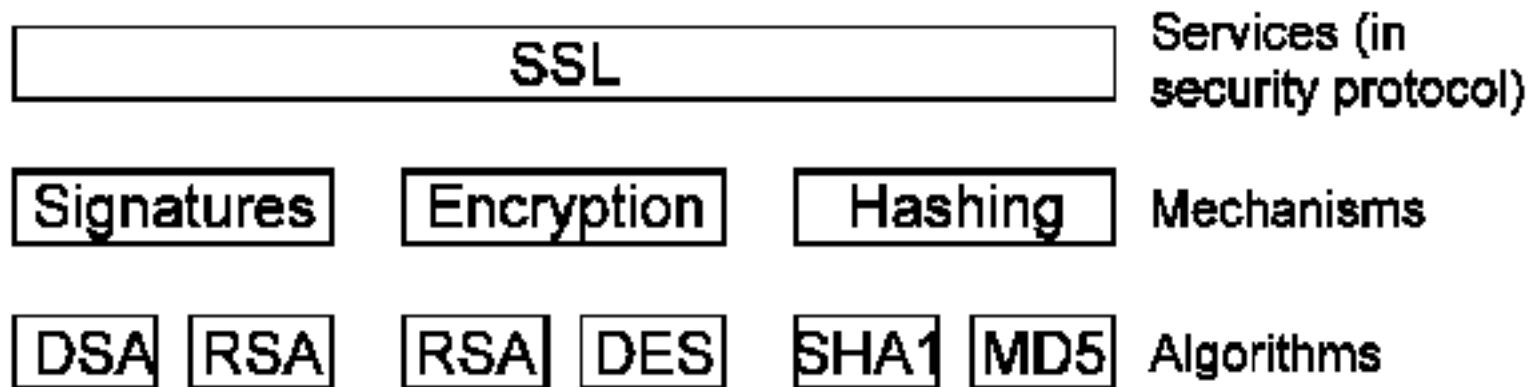
# Les mécanismes de sécurité

---

- Chiffrement : utilisé pour assurer la confidentialité, l'authentification et l'intégrité
- Signature numérique : utilisée pour assurer l'authentification, l'intégrité et la non-répudiation
- Somme de contrôle/algorithmes de hachage : utilisés pour assurer l'intégrité et l'authentification

# Services, mécanismes et algorithmes

- Les services utilisent les mécanismes de sécurité
- Les mécanismes sont implantés grâce à des algorithmes



# La cryptographie

---

- Permet d'assurer confidentialité des données, authentification, intégrité, non répudiation
- Ne sert à rien s'il existe d'autres moyens d'obtenir la même information (*low-teck* attack)
- Systèmes cryptographiques basés sur un algorithme et une valeur secrète appelée «clef»
- Algorithmes connu de tous

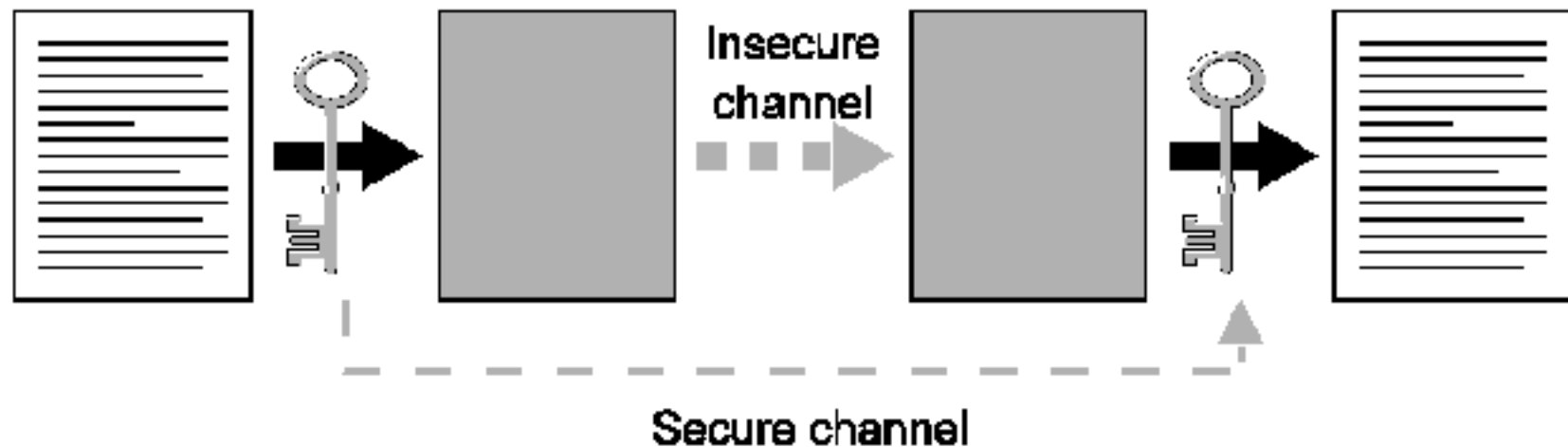
# La cryptographie

---

- Cryptographie basée sur la difficulté de calcul :
  - Toujours possible décrypter en utilisant toutes les clefs (*brute force*)
- Sécurité d'un algorithme = personne n'a (dit avoir) trouvé de méthode plus rapide
- Difficulté souvent liée à la longueur de la clef

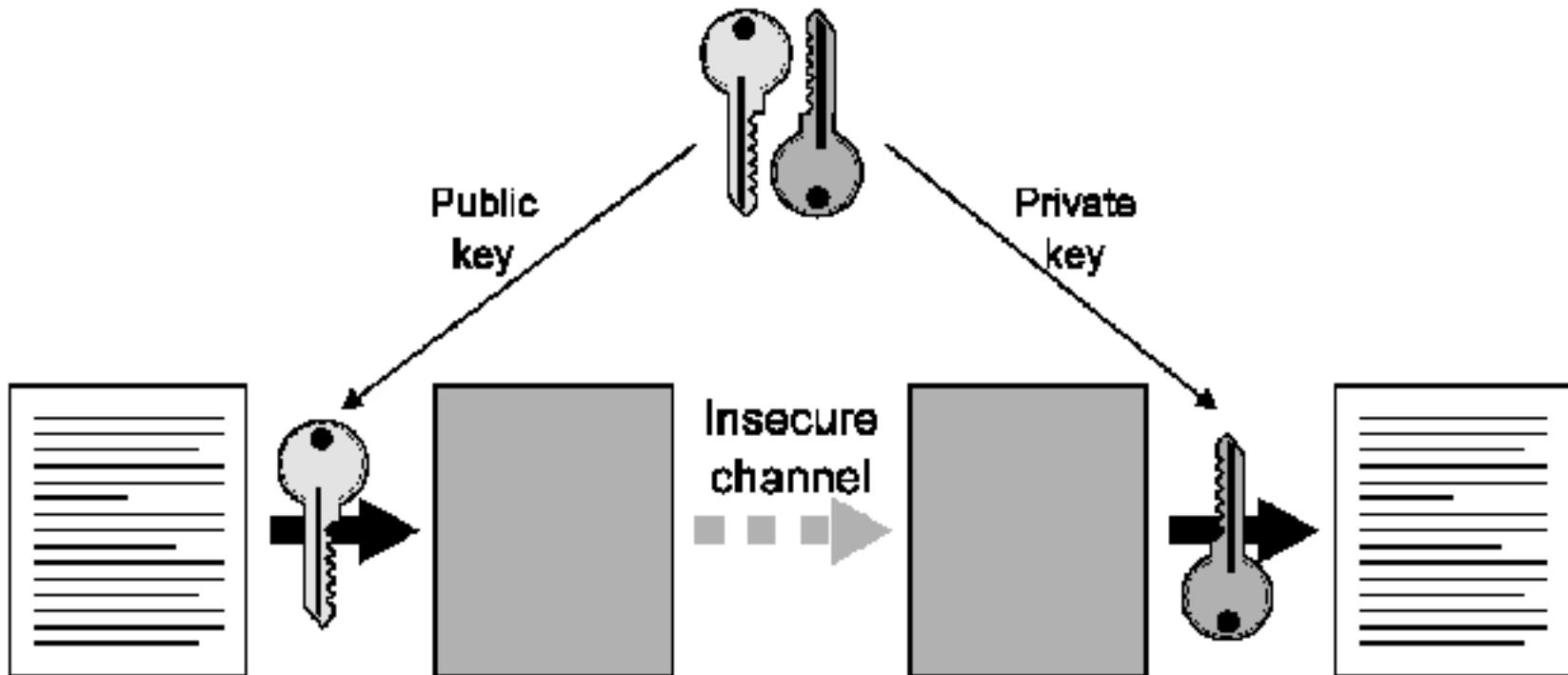
# Chiffrement à clef secrète

- Utilisation d'une clef partagée



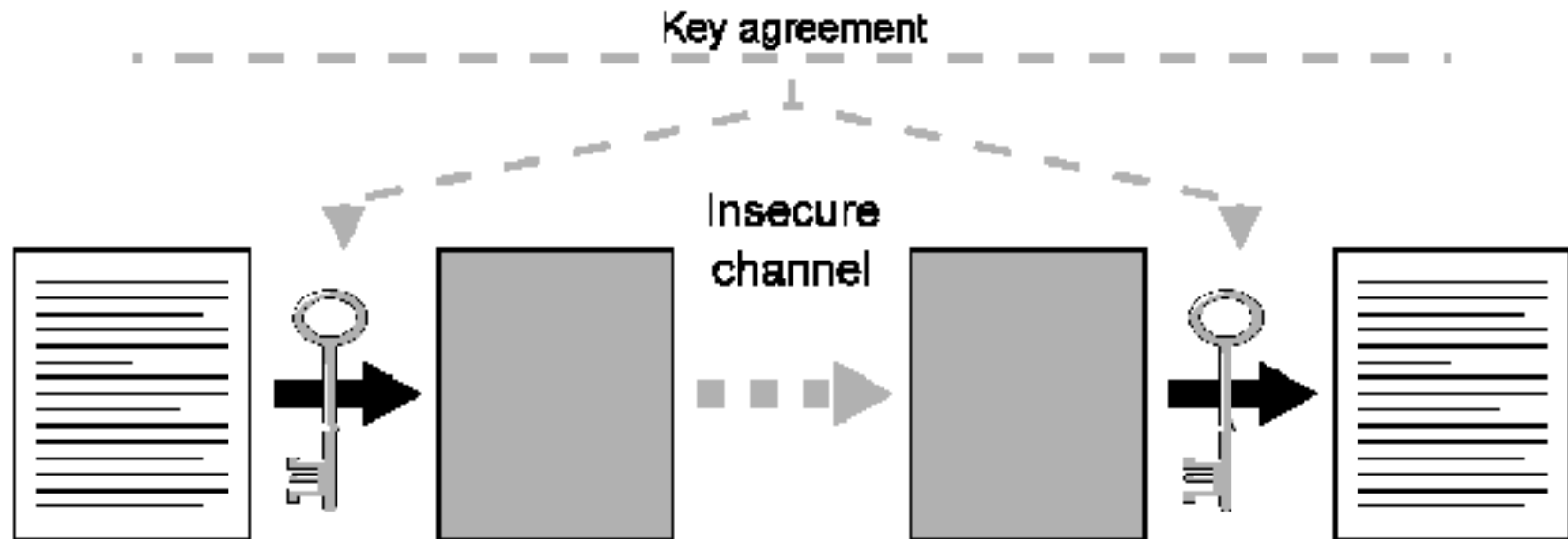
# Chiffrement à clef public

- Tout le monde peut chiffrer avec la clef publique, un seul peut déchiffrer avec la clef privée



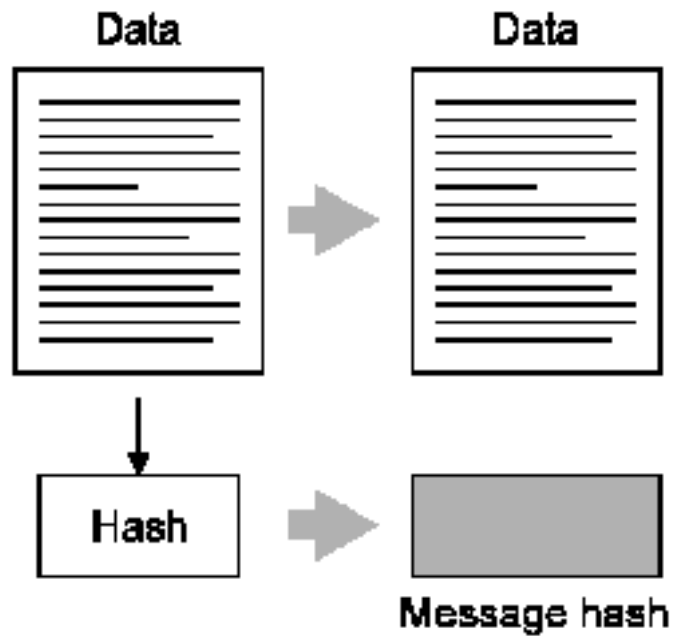
# Accord sur les clefs

- Permet à deux parties de se mettre d'accord sur une même clef



# Hachage

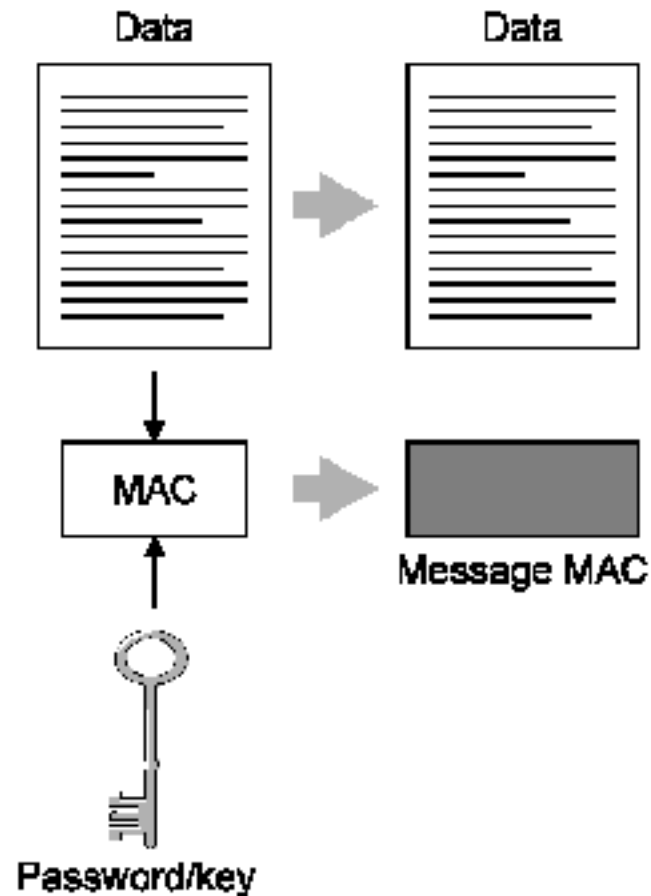
- Crée une valeur (empreinte) unique pour un fichier





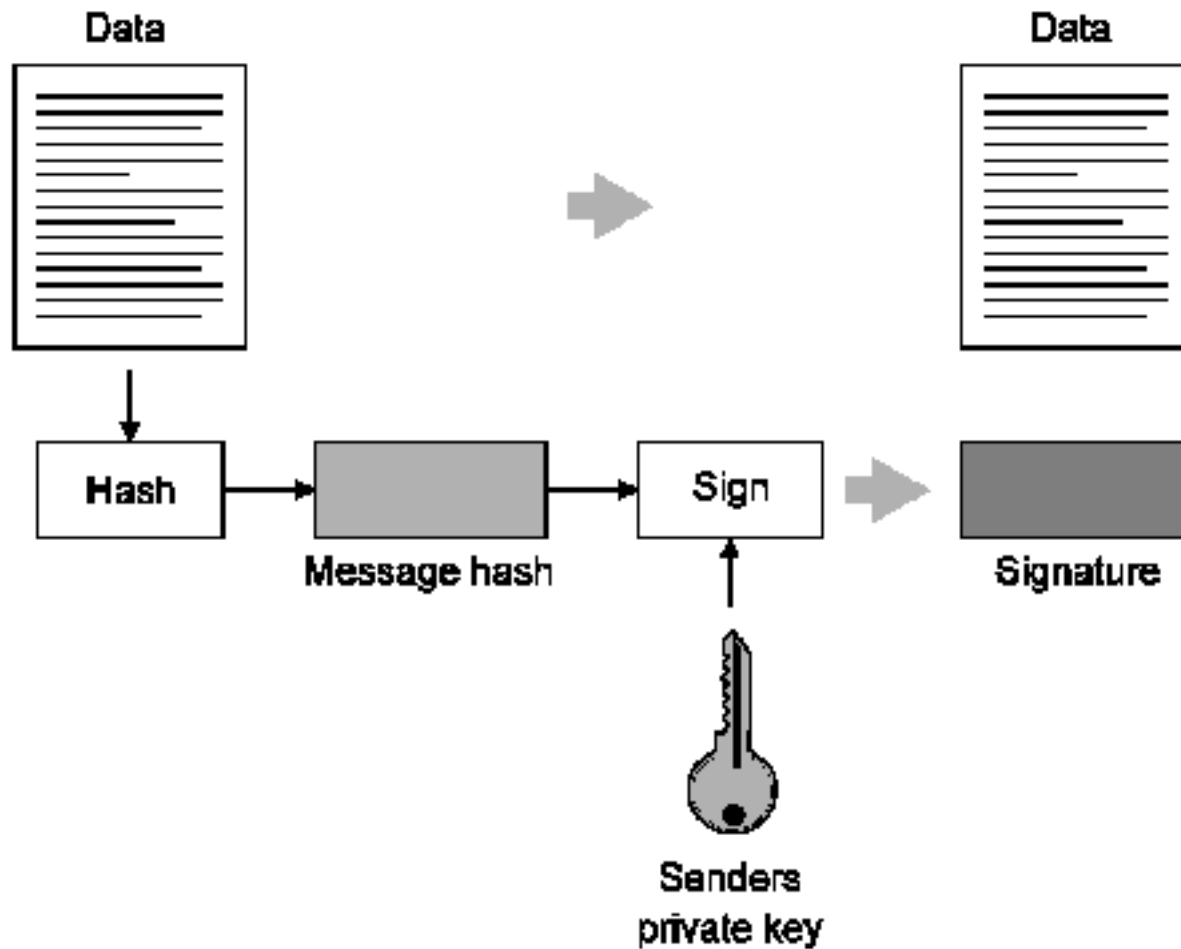
# MAC

- Message Authentication Code, ajoute la clef au hachage



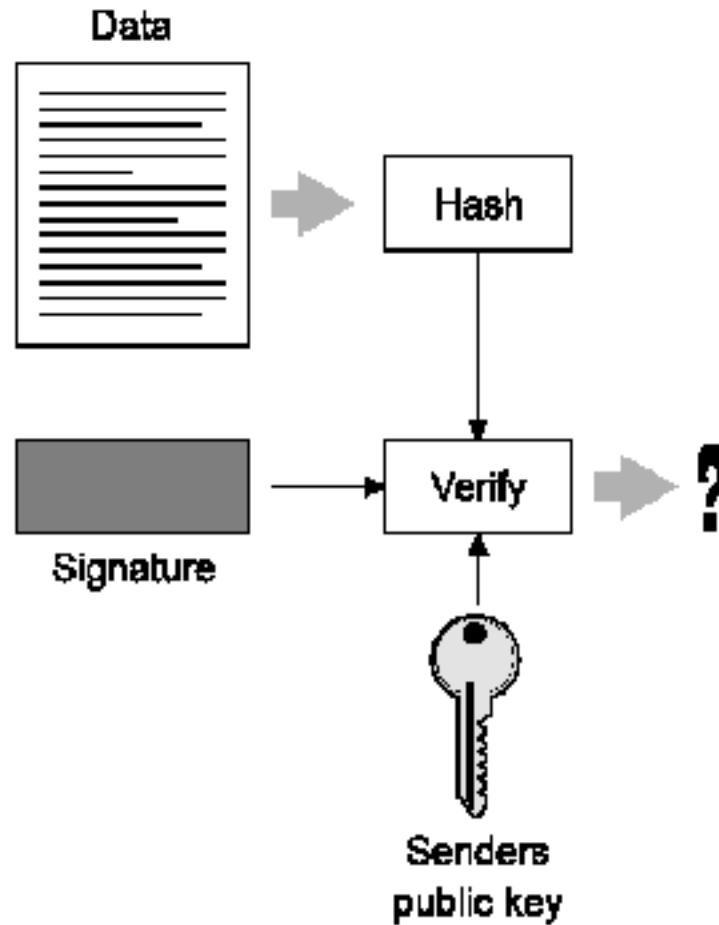
# Signature numérique

- Chiffrement de la valeur de hachage



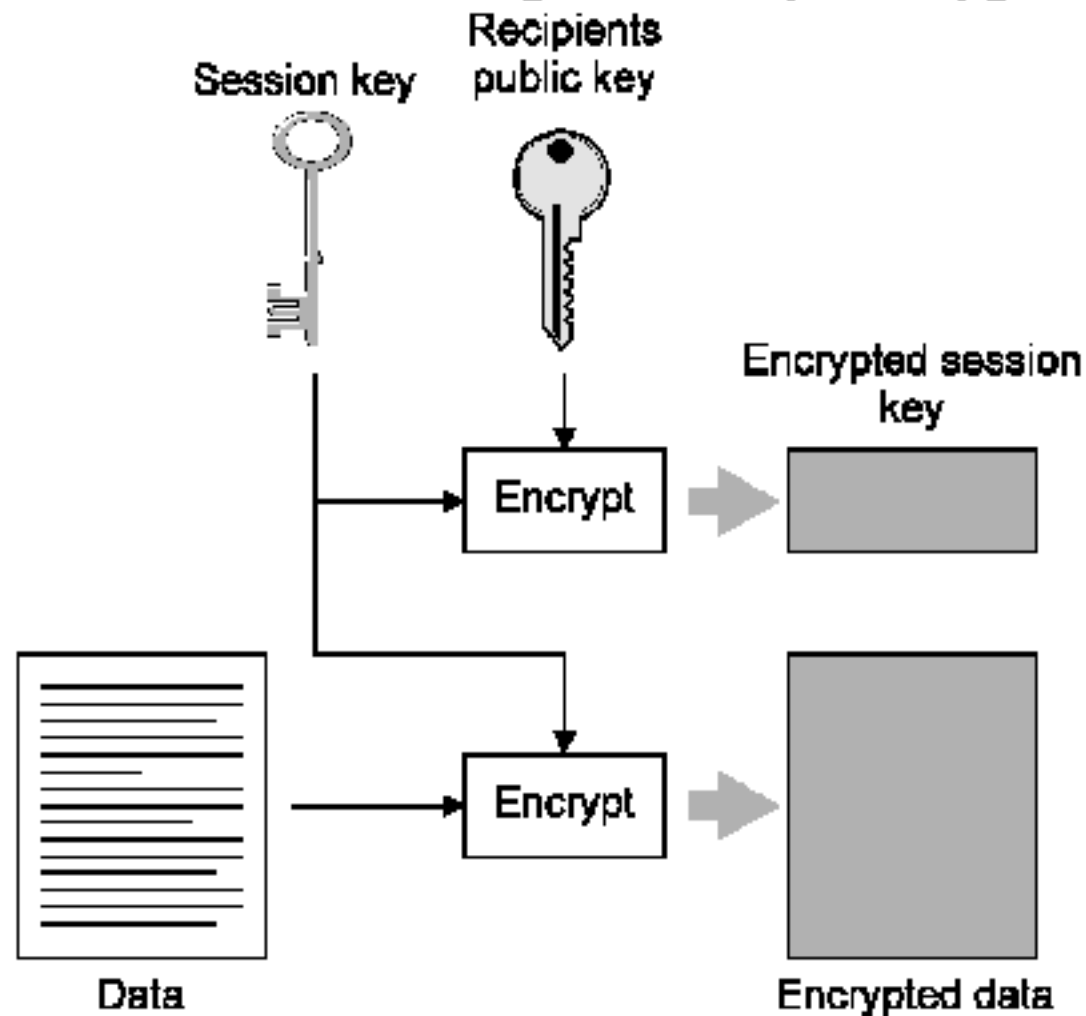
# Signature numérique

- Vérification



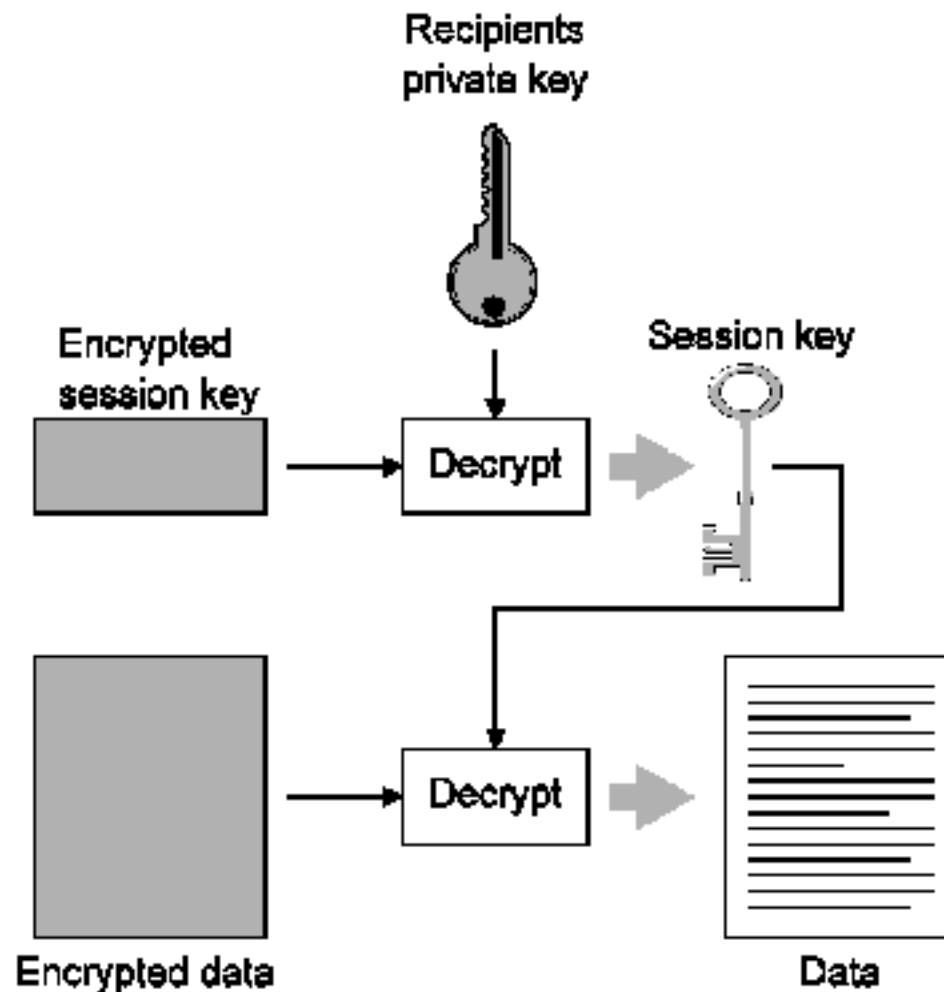
# Chiffrement de messages

- Combine chiffrement à clef secrète et publique



# Chiffrement des messages

- Vérification



# Les algorithmes à clef secrète

---

- Chiffrement de flux (*Stream Cipher*): basé sur XOR
  - RC4

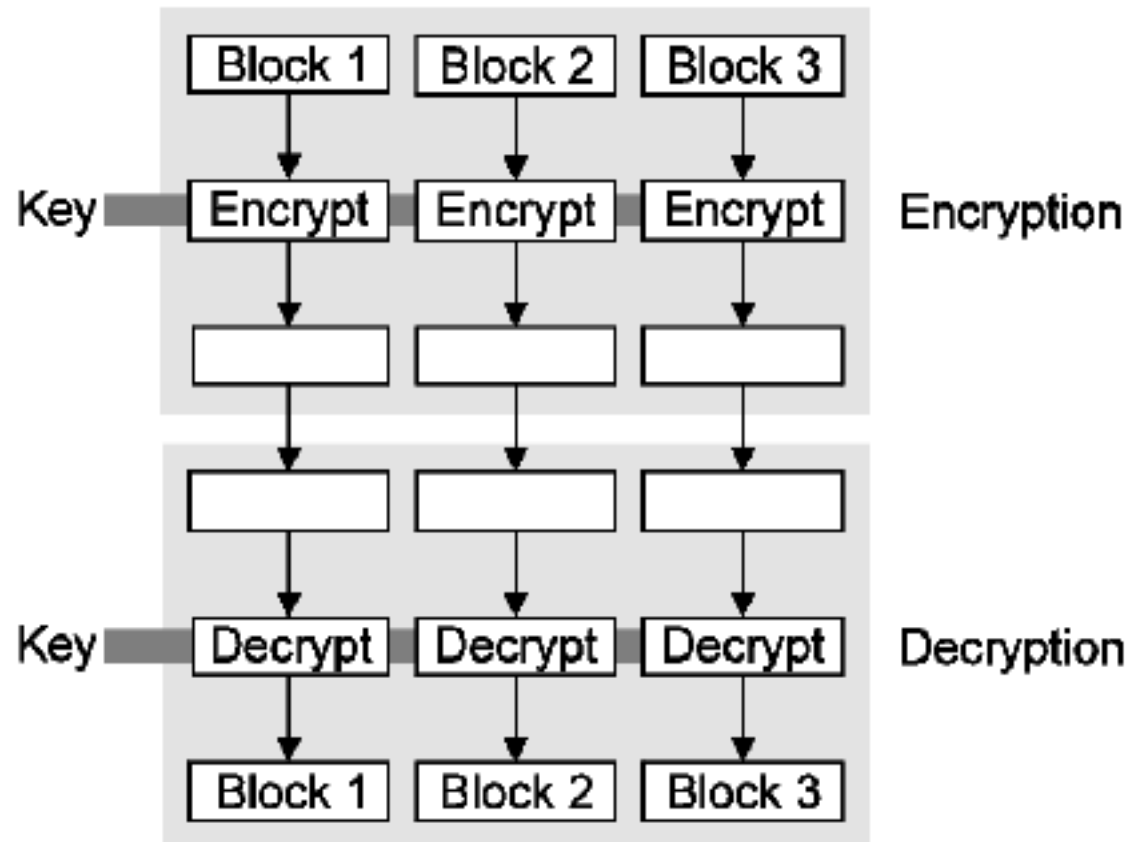
# Les algorithmes à clef secrète

---

- Chiffrement par bloc (*Bulk Cipher*) : basé sur permutation multiple
  - DES
  - 3DES
  - RC2
  - IDEA
  - blowfish
  - CAST-128
  - AES

# Les algorithmes à clef secrète

- Chiffrement de flux avec algorithme sur bloc
  - Electronique Codebook





# Chiffrer des messages longs

---

- Algorithmes sur blocs de taille fixe
- Chiffrement bloc par bloc
  - Deux blocs identiques donnent un même valeur
    - ✧ donne des informations sur le message

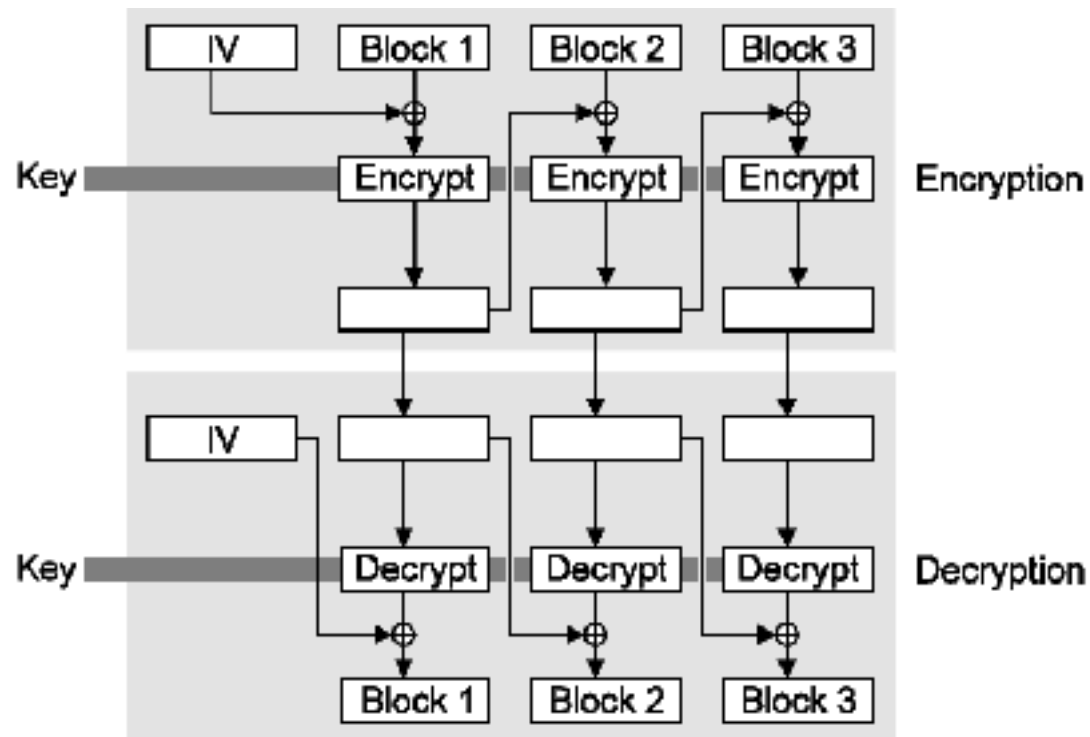
# Chiffrer des messages longs

---

- Chaînage des blocs
  - Obtenir une valeur aléatoire
  - Appliquer ou exclusif avec bloc
  - Chiffrer avec la clef secrète
  - Utiliser la valeur obtenue comme valeur aléatoire pour le bloc suivant
- Nécessité de transmettre la valeur initiale *Initialization Vector (IV)*
  - Si IV toujours identique permet de détecter une différence entre deux messages

# Les algorithmes à clef secrète

- Chaînage du chiffrement
  - CBC (*Cipher Block Chaining*)



# Les performances

- Rapide

RC4

Blowfish, AES, CAST-128

DES, IDEA, RC2

3DES

- Lent

- RC4 10 fois plus rapide que 3DES (~Mbit/s)

# Algorithme à clef publique

---

- Utilisé pour chiffrement, signature, échange de clefs
  - RSA : chiffrement, signature, échange
  - Deffie-Hellman : échange
  - elgamal : signature, chiffremetn
  - DSA : signature
- Utilise au minimum des clefs de 1024 bits (pas 512)

# Algorithmes de hachage

---

- Réduit l'entrée en une taille fixe
  - MD2 128bits ne plus utiliser
  - MD4 128bits cassé
  - MD5 128bits faiblesse
  - SHA-1 160bits

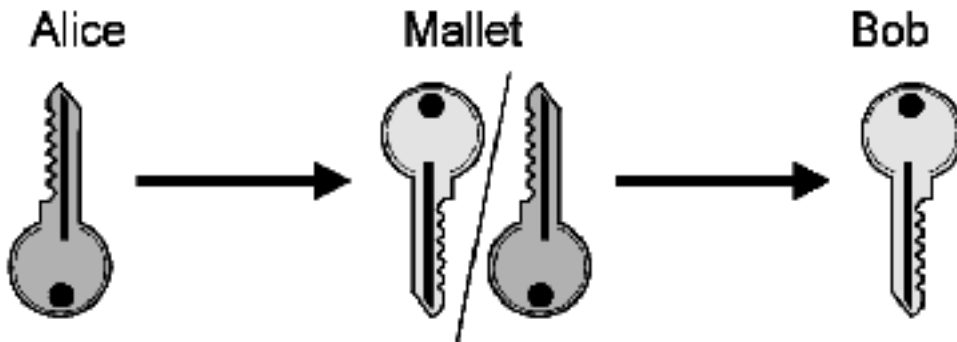
# Algorithmes MAC

---

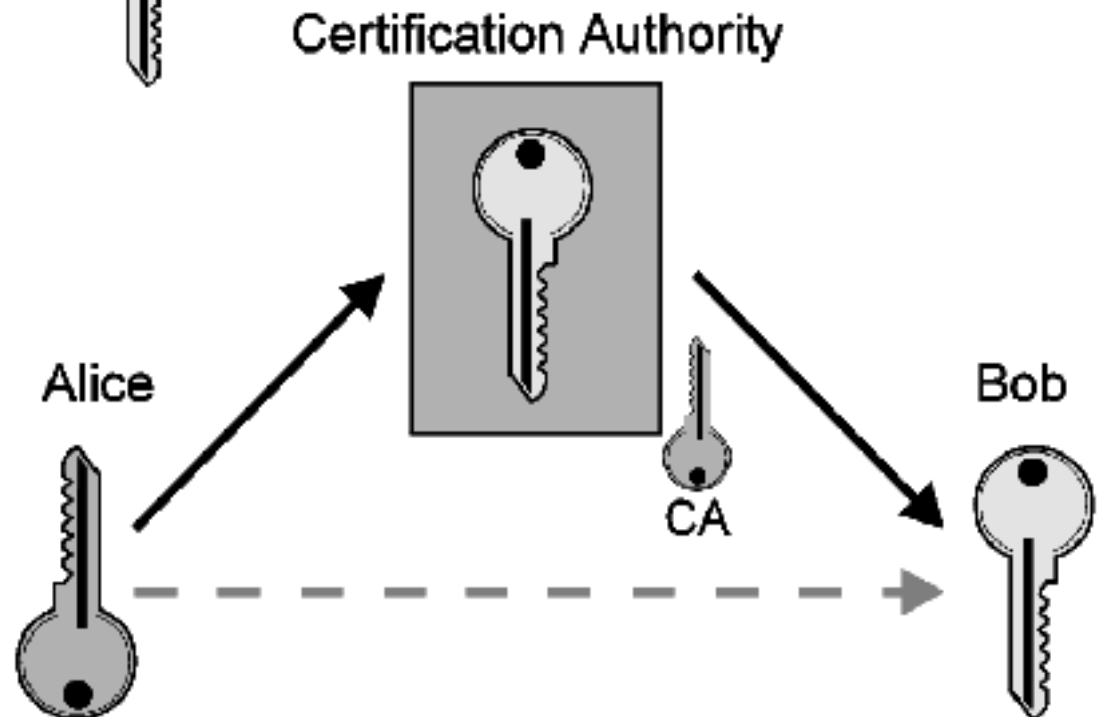
- HMAC
  - $\text{hash}(\text{key}, \text{hash}(\text{key}, \text{data}))$
- HMAC-MD5 ou HMAC-SHA

# Distribution des clefs publiques

- Problème de Man-In-The-Middle



- Utilisation d'un tiers de confiance



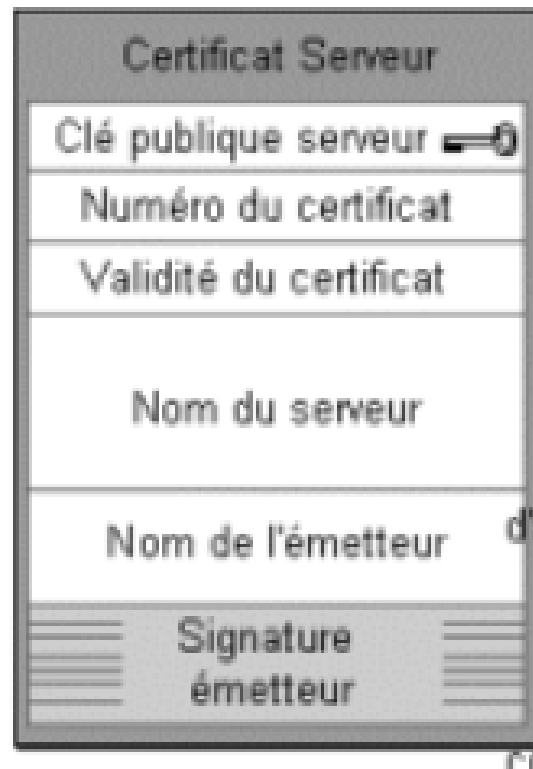


# Certificats

- Certificats X509
- Utilisation de la hiérarchie de nommage X500
  - Notion de *Distinguished Name* (DN) :
    - \* C country =FR
    - \* SP state-province
    - \* L locality = Champs
    - \* O organisation = UMLV
    - \* OU organisational unit = IGM
    - \* CN Common Name = Gilles Roussel
    - \* EMAILADDRESS mail = rousssel@univ-mlv.fr

# Les certificats

- Format des certificats X509
- Possibilité de chaînage des certificats

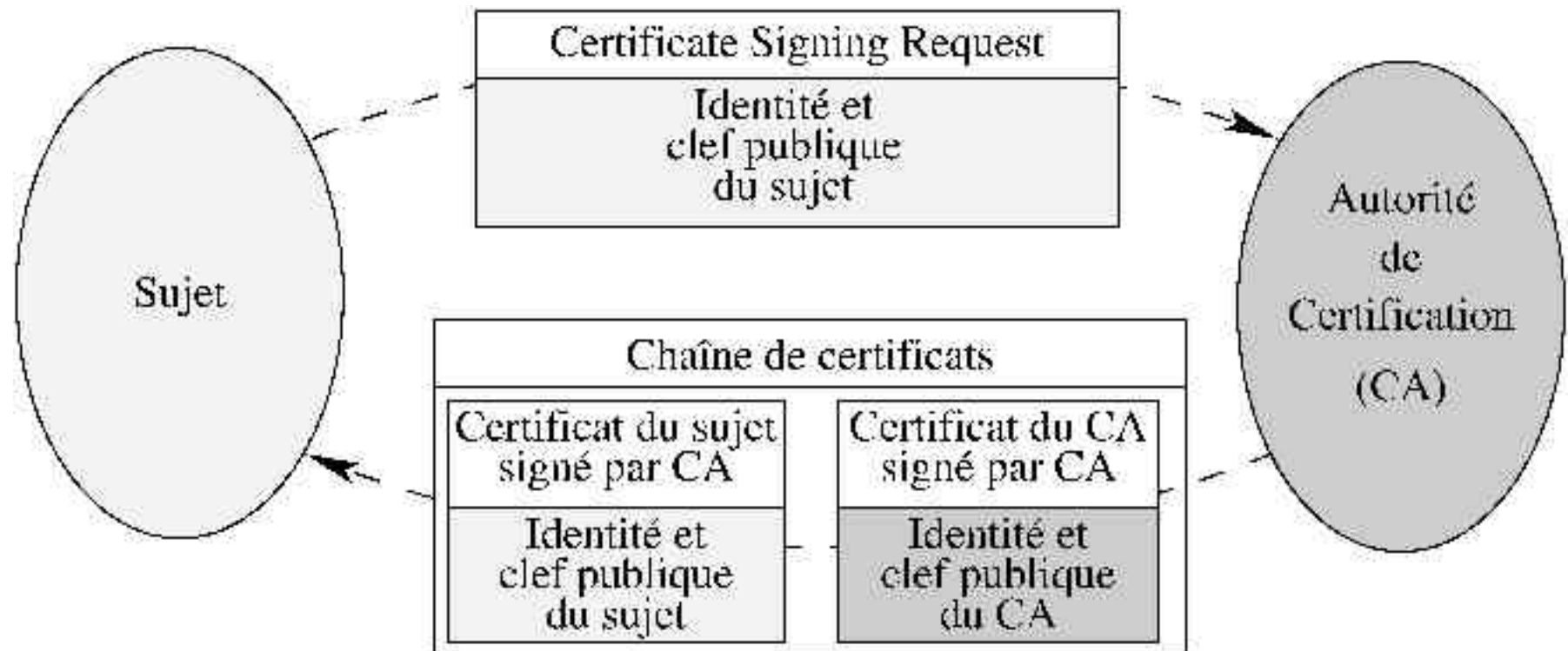


# Les certificats

---

- Par défaut les certificats sont autocertifiés
- Demande de certification par un tiers de confiance
  - Verisign, Thawte, Entrust, Certimonis, ....
- *Certificate Signing Request* est envoyé à l'autorité de certification (CA)

# Les certificats



# openssl

- Création d'une paire de clefs RSA de 1024 bits protégée par un mot de passe et chiffrée en DES3

```
$ openssl genrsa -des3 -out server.key 1024
```

- Affichage paramètres clefs

```
$ openssl rsa -noout -text -in server.key
```

# openssl

- **Création d'une requête de certification**

```
$ openssl req -new -key server.key 1024 -out  
server.csr
```

```
Country Name (2 letter code) [GB]:FRState or
```

```
Province Name (full name) [Berkshire]:.
```

```
Locality Name (eg, city) [Newbury]:Champs
```

```
Organization Name (eg, company) [My Company  
Ltd]:UMLV
```

```
Organizational Unit Name (eg, section) []:  
IGM
```

```
Common Name (eg, your name or your server's  
hostname) []:Gilles Roussel
```

```
Email Address []:rousseau@univ-mlv.fr
```

# Créer son propre CA

- **Création de la clef privée du CA**

```
$ openssl genrsa -des3 -out ca.key 1024
```

- **Auto-signature d'un certificat**

```
$ openssl req -new -x509 -days 365 -key ca.key  
-out ca.crt
```

```
Country Name (2 letter code) [GB]:FR
```

```
State or Province Name (full name) [Berkshire]:.
```

```
Locality Name (eg, city) [Newbury]:Champs
```

```
Organization Name (eg, company) [My Company Ltd]:
```

```
UMLV
```

```
Organizational Unit Name (eg, section) []:IGM
```

```
Common Name (eg, your name or your server's  
hostname) []:CA
```

```
Email Address []:ca@univ-mlv.fr
```

# openssl

---

- Utilisation d'un script `mod_ssl` pour la signature
- Signature d'un *Certificate Signing Request*  

```
$ ./sign.sh server.csr
```