

## Sécurisation des communications

## Fiche de synthèse

## ① TROIS OBJECTIFS

Une communication sécurisée garantit la **confidentialité**, seuls l'émetteur et le destinataire lisent le message; l'**intégrité**, le message n'est pas modifié en chemin; l'**authentification**, chacun est sûr de l'identité de l'autre.

## ② CHIFFREMENT SYMÉTRIQUE

**Définition 1 (Chiffrement symétrique)** Dans un **chiffrement symétrique**, la même clé, partagée par l'émetteur et le destinataire, sert à chiffrer et à déchiffrer.

Il est rapide, mais il faut d'abord se transmettre la clé sans qu'un tiers l'intercepte.

```
def chiffre_cesar(message, decalage):
    resultat = ""
    for lettre in message:
        rang = ord(lettre) - ord("A")
        nouveau = (rang + decalage) % 26
        code = ord("A") + nouveau
        resultat = resultat + chr(code)
    return resultat

chiffre_cesar("NSI", 3)    # 'QVL'
```

Le chiffrement de César décale chaque lettre d'un même nombre de rangs; on déchiffre avec le décalage opposé. Il n'est pas sûr : il suffit d'essayer les 25 décalages possibles.

## ③ CHIFFREMENT ASYMÉTRIQUE

**Définition 2 (Chiffrement asymétrique)** Dans un **chiffrement asymétrique**, chaque personne possède une paire de clés : une **clé publique**, diffusée à tous, qui sert à chiffrer, et une **clé privée**, gardée secrète, qui seule permet de déchiffrer.

Pour écrire à Jeanne, Serge chiffre avec la clé publique de Jeanne; seule Jeanne, qui détient la clé privée, peut déchiffrer. Aucune clé secrète n'a besoin de circuler.

L'algorithme **RSA** repose sur une difficulté : multiplier deux grands nombres premiers  $p$  et  $q$  est facile, mais retrouver  $p$  et  $q$  à partir de leur produit  $n$  est extrêmement long. La clé publique est le couple  $(n, e)$ ; la clé privée, un nombre  $d$ , ne se calcule qu'à partir de  $p$  et  $q$ .

Il est beaucoup plus lent que le chiffrement symétrique et ne prouve pas l'identité de l'émetteur, puisque tout le monde connaît la clé publique.

## ④ SIGNATURE ÉLECTRONIQUE

## ≡ Signer et vérifier un message.

1. Serge calcule l'**empreinte** du message avec une **fonction de hachage** : un condensé du message, qui ne permet pas de le retrouver.
2. Il chiffre cette empreinte avec sa **clé privée** : c'est la signature, qu'il envoie avec le message.
3. Jeanne déchiffre la signature avec la clé publique de Serge, calcule elle-même l'empreinte du message reçu et compare les deux.

Si les empreintes sont égales, le message vient bien de Serge, seul détenteur de sa clé privée, et n'a pas été modifié : la signature garantit l'authentification et l'intégrité.

## ⑤ CERTIFICATS

Un tiers pourrait publier une clé publique au nom d'une autre personne et se faire passer pour elle : c'est l'**attaque de l'homme du milieu**. Une **autorité de certification** vérifie l'identité du propriétaire d'une clé publique, puis délivre un **certificat électronique** qu'elle signe avec sa propre clé privée. Les autorités et leurs certificats forment une **infrastructure à clés publiques**.

## ⑥ LE PROTOCOLE TLS

TLS, anciennement SSL, est un protocole **hybride** : il utilise le chiffrement asymétrique pour échanger une clé, puis le chiffrement symétrique, plus rapide, pour les données. HTTPS est HTTP sécurisé par TLS.

1. Le client demande une connexion sécurisée au serveur.
2. Le serveur envoie son certificat, qui contient sa clé publique.
3. Le client vérifie le certificat auprès de l'autorité de certification.
4. Le client crée une **clé de session** au hasard, la chiffre avec la clé publique du serveur et l'envoie.
5. Le serveur la déchiffre avec sa clé privée. Dans TLS 1.3, elle est plutôt calculée de part et d'autre, sans circuler.
6. Les échanges sont ensuite chiffrés de façon symétrique avec la clé de session.

Un attaquant qui intercepte l'étape 4 ne peut pas lire la clé de session sans la clé privée du serveur.

## 📌 À retenir.

- Symétrique : une clé partagée. Asymétrique : on chiffre avec la clé publique du destinataire, il déchiffre avec sa clé privée.