Le chiffrement est une technique essentielle en sécurité informatique pour protéger les données contre l'accès non autorisé. Il existe deux principaux types de chiffrement : le chiffrement symétrique et le chiffrement asymétrique. Voici une explication des concepts de base de chacun d'eux.

Chiffrement Symétrique

Définition:
Le chiffrement symétrique utilise une seule clé pour chiffrer et déchiffrer les données. Cette clé doit être partagée entre les parties communicantes de manière sécurisée.

Caractéristiques:

- **Clé Unique** : La même clé est utilisée pour le chiffrement et le déchiffrement.
- **Rapidité** : Les algorithmes de chiffrement symétrique sont généralement plus rapides et nécessitent moins de ressources que les algorithmes asymétriques.
- **Sécurité de la Clé** : La sécurité repose entièrement sur le secret de la clé. Si la clé est compromise, les données chiffrées le sont également.

Exemples d'Algorithmes Symétriques:

- **AES (Advanced Encryption Standard)** : Utilisé largement dans diverses applications pour sa robustesse et sa rapidité.
- **DES (Data Encryption Standard)** : Plus ancien et moins sécurisé que l'AES, mais encore utilisé dans certains contextes.
- **3DES (Triple DES)** : Amélioration du DES en appliquant le chiffrement trois fois pour renforcer la sécurité.

Exemple de Chiffrement Symétrique avec OpenSSL:

1. **Chiffrement** :

```
openssl enc -aes-256-cbc -salt -in fichier.txt -out fichier.txt.enc
```

2. **Déchiffrement** :

```
openssl enc -aes-256-cbc -d -in fichier.txt.enc -out fichier.txt
```

Chiffrement Asymétrique

Définition:
Le chiffrement asymétrique utilise une paire de clés : une clé publique pour chiffrer les données et une clé privée correspondante pour déchiffrer les données. La clé publique peut être partagée librement, tandis que la clé privée doit rester secrète.

Caractéristiques:

- **Paire de Clés** : Utilisation de deux clés distinctes, une publique et une privée.
- **Sécurité** : La sécurité repose sur la difficulté mathématique de dériver la clé privée à partir de la clé publique.
- **Lenteur Relative** : Les algorithmes asymétriques sont généralement plus lents et consomment plus de ressources que les algorithmes symétriques.

Exemples d'Algorithmes Asymétriques:

- **RSA (Rivest-Shamir-Adleman)** : L'un des algorithmes asymétriques les plus connus et les plus utilisés.
- **ECC (Elliptic Curve Cryptography)** : Fournit un niveau de sécurité équivalent à RSA mais avec des clés plus petites et une meilleure performance.

Exemple de Chiffrement Asymétrique avec OpenSSL:

1. **Génération d'une paire de clés** :

```
openssl genpkey -algorithm RSA -out clé_privée.pem -pkeyopt
rsa_keygen_bits:2048
openssl rsa -pubout -in clé_privée.pem -out clé_publique.pem
```

2. **Chiffrement avec la clé publique** :

```
openssl rsautl -encrypt -inkey clé_publique.pem -pubin -in fichier.txt -
out fichier.txt.enc
```

3. **Déchiffrement avec la clé privée** :

```
openssl rsautl -decrypt -inkey clé_privée.pem -in fichier.txt.enc -out
fichier.txt
```

Comparaison entre Chiffrement Symétrique et Asymétrique

Caractéristique	Chiffrement Symétrique	Chiffrement Asymétrique
Nombre de Clés	1 clé unique	2 clés (publique et privée)
Vitesse	Rapide	Plus lent
Sécurité de la Clé	Dépend du secret de la clé unique	Clé privée doit rester secrète
Usage Typique	Chiffrement de données en masse	Échange de clés, signatures digitales
Complexité	Relativement simple	Plus complexe

Utilisations Courantes

- **Chiffrement Symétrique** : Utilisé pour chiffrer de grandes quantités de données, comme dans le cas du chiffrement des disques durs ou des bases de données.
- **Chiffrement Asymétrique** : Utilisé pour sécuriser les échanges de clés dans les communications SSL/TLS, pour les signatures numériques et pour les certificats numériques.