Fondamentaux en matière de cryptographie

Documentation rédigée par : GADONNAUD Ewen, en lien avec les cours de DEMOULIERE Quentin

Formation : BTS SIO 1ère année - Option SISR

Établissement : Lycée Paul-Louis Courier, Tours

Date : Mars 2026

1. Vocabulaire : Hachage

> Critères **DICP** concernés : **Intégrité**

Le hachage est un processus qui consiste à transformer une donnée d'entrée (fichier, mot de passe, chaîne de caractères) en une **empreinte numérique de taille fixe** à l'aide d'un algorithme mathématique.

Caractéristiques principales

Propriété	Description
Déterministe	La même entrée produit toujours la même empreinte.
Irréversible	Il est quasi impossible de retrouver la donnée originale à partir de l'empreinte.
Taille fixe	Peu importe la taille de l'entrée, l'empreinte a toujours la même longueur.

Utilisations courantes

- **Vérification d'intégrité** : Comparer les empreintes avant et après un transfert pour détecter des altérations.
- **Stockage sécurisé des mots de passe** : Seule l'empreinte est stockée, jamais le mot de passe en clair. On peut y ajouter un **sel (salt)** pour renforcer la sécurité.
- **Structures de données** : Tables de hachage (hash tables) pour une recherche rapide.

---

2. Vocabulaire : Chiffrement

> Critères **DICP** concernés : **Confidentialité**

Opération	Description
Chiffrer	Transformer une donnée de façon qu'elle devienne incompréhensible. Seules les entités autorisées pourront la comprendre.
Déchiffrer	Reconstituer la donnée d'origine à partir de la donnée chiffrée. Seules les entités autorisées peuvent effectuer cette opération.

Les deux opérations font appel à **un algorithme et une clé cryptographique**.

Le terme correct est **“déchiffrer”** (avec la clé) et non “décrypter” (sans la clé, par force brute).

---

3. Vocabulaire : Signature

> Critères **DICP** concernés : **Authenticité** → Prouver l'identité de l'émetteur d'une donnée.

Opération	Description
Signer	Créer une signature électronique unique liée à la donnée et à son auteur. Utilise une **clé privée** + le message en entrée → signature en sortie.
Vérifier la signature	S'assurer que la donnée n'a pas été modifiée et que son auteur est authentifié. Utilise la **clé publique** + la signature + le message → verdict **OK** / **NOK**.

Si la signature n'est pas valide, il ne faut pas faire confiance au document.

---

4. Fonction de hachage

Une fonction de hachage permet d'obtenir, à partir d'un fichier ou d'une chaîne de caractères, une **empreinte unique de taille fixe**.

```
Entrée : "Renard"              → Empreinte : DFCD3454
Entrée : "Le renard court..."  → Empreinte : 52ED879E
Entrée : "Le renard marche..." → Empreinte : 46042841
```

- Contrairement au chiffrement, il est **impossible d'obtenir l'entrée originale à partir de l'empreinte** (non-réversibilité).
- Utilisée pour : stockage des mots de passe, contrôle d'intégrité, signature numérique.

Collision

Une **collision** survient quand deux entrées différentes produisent la même empreinte. C'est pourquoi certains algorithmes sont considérés comme **obsolètes**.

Algorithmes

Algorithme	État
SHA-2, SHA-3	État de l'art — à utiliser
MD5, SHA-1	Obsolètes — peuvent générer des collisions

---

5. Chiffrement symétrique

En cryptographie symétrique, **une seule et même clé** est utilisée pour chiffrer et déchiffrer le message.

```
A ──[clé secrète]──► chiffrement ──► B
B ──[clé secrète]──► déchiffrement ──► message original
```

- La clé doit rester **secrète** : seules les personnes habilitées doivent la posséder.
- Si la clé est compromise, la confidentialité du message n'est plus assurée.

Algorithme courant

- **AES-256** (Advanced Encryption Standard, clé de 256 bits) — standard actuel recommandé.

Limite principale

Problème de transmission de la clé : comment transmettre la clé secrète de manière sécurisée entre A et B sans qu'un tiers l'intercepte ? C'est précisément ce que résout le chiffrement asymétrique.

---

6. Chiffrement asymétrique

Le chiffrement asymétrique utilise une **paire de clés** :

Clé	Couleur (convention)	Rôle
Clé **publique**	Verte	Partagée librement — utilisée pour **chiffrer**
Clé **privée**	Rouge	Gardée secrète — utilisée pour **déchiffrer**

Déroulement

1. **Étape 1** : Alice génère une paire de clés. Elle envoie sa **clé publique** à Bob et conserve sa **clé privée**.
2. **Étape 2** : Bob chiffre le message avec la **clé publique d'Alice**.
3. **Étape 3** : Alice déchiffre le message avec sa **clé privée**.

Algorithmes liés

- **RSA**
- **ECDSA**
- **ED25519**

---

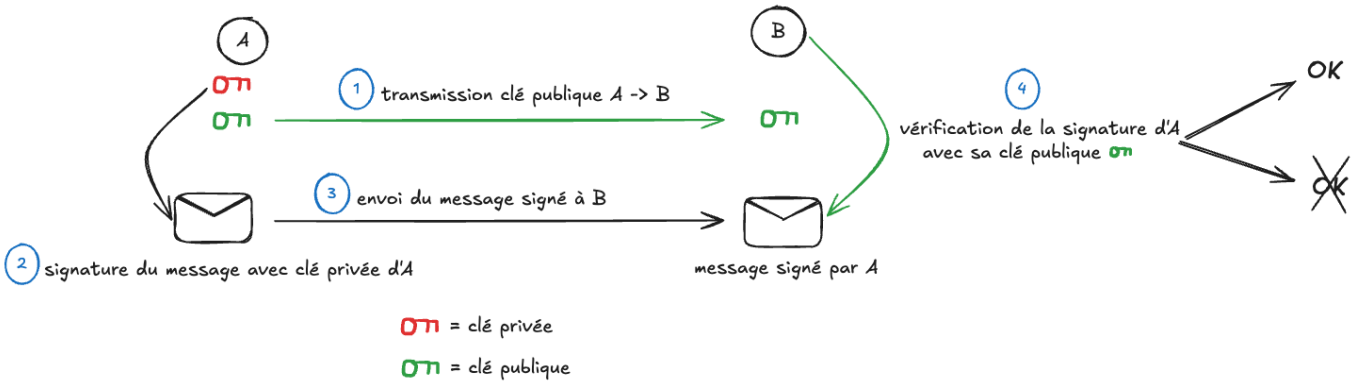
7. Signature à l'aide d'une paire de clés

La signature numérique repose sur le chiffrement asymétrique mais dans le sens **inverse** du chiffrement.

Récapitulatif des rôles des clés

Clé	Opération
Clé privée	Déchiffrer, **Signer**
Clé publique	Chiffrer, **Vérifier une signature**

Flux de la signature (4 étapes)



1. **A transmet sa clé publique** à B au préalable.
2. **A signe le message** avec sa clé privée.
3. **A envoie le message signé** à B.
4. **B vérifie la signature** avec la clé publique d'A → résultat **OK** (signature valide) ou **NOK** (signature invalide / message altéré).

Outil : GPG

GPG (GNU Privacy Guard) permet la création de paires de clés pour diverses actions : signature, chiffrement, authentification.

GPG est l'implémentation libre du standard OpenPGP. Couramment utilisé pour signer des commits Git ou chiffrer des emails.

Récapitulatif général

Mécanisme	Objectif DICP	Clé(s)	Réversible ?
Hachage	Intégrité	Aucune	❌ Non
Chiffrement symétrique	Confidentialité	1 clé secrète partagée	✅ Oui
Chiffrement asymétrique	Confidentialité	Paire pub/priv	✅ Oui
Signature numérique	Authenticité	Paire pub/priv	N/A