Protection des Données : Confidentialité, Intégrité et Authenticité

Comprendre les mécanismes de la cryptographie moderne

Vos données voyagent en terrain hostile

Toute donnée transmise sur un réseau public peut être interceptée par un tiers malveillant. Sans protection, l'information est lisible, vulnérable et exposée.

La Triade de la Sécurité

SÉCURITÉ

Confidentialité

Les données sont accessibles seulement par Alice et Bob. (Personne d'autre ne peut lire).

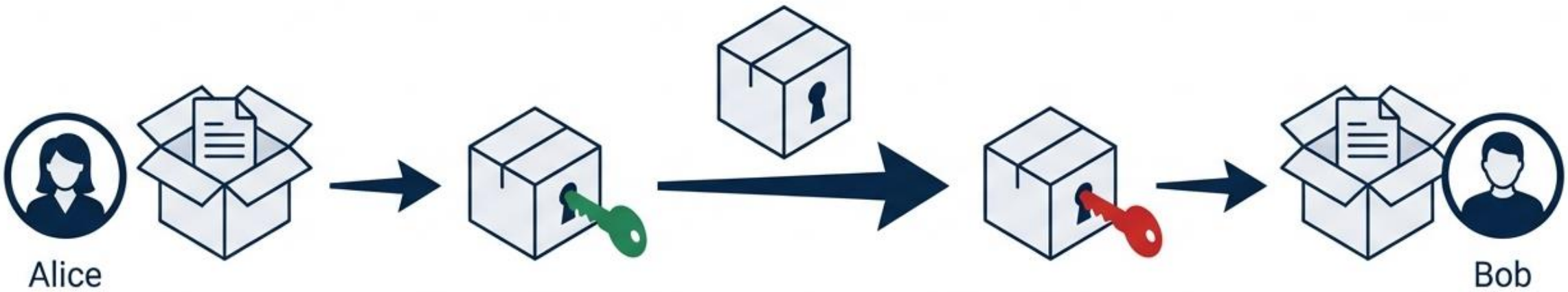
Intégrité

Les données ne sont pas modifiées durant le transit. (Le message arrive intact).

Authenticité

Alice et Bob sont réellement qui ils prétendent être. (Pas d'imposteurs).

Pilier 1 - La Confidentialité (Chiffrement Asymétrique)



Alice chiffre avec la clé publique de Bob.

Bob déchiffre avec sa clé privée.

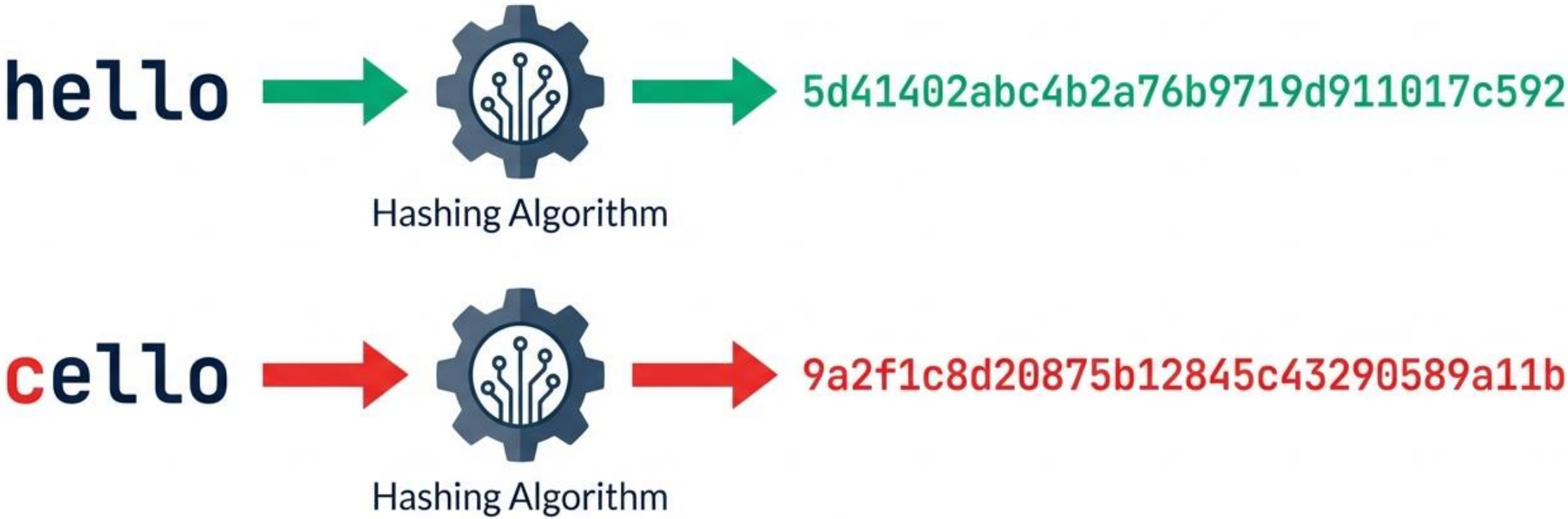
Clé Publique : Sert à verrouiller (chiffrer).

Clé Privée : Sert à déverrouiller (déchiffrer).

L’Attaque - Man-in-the-Middle (L’homme du milieu)

Le chiffrement fonctionne techniquement, mais la confiance est rompue. Le pirate lit tout car Bob a utilisé la mauvaise clé.

Pilier 2 - L'Intégrité (Le Hachage)



Le hachage est une empreinte numérique unique. Si on change un seul caractère du message, l'empreinte change radicalement.

Algorithmes de Hachage : Les Standards

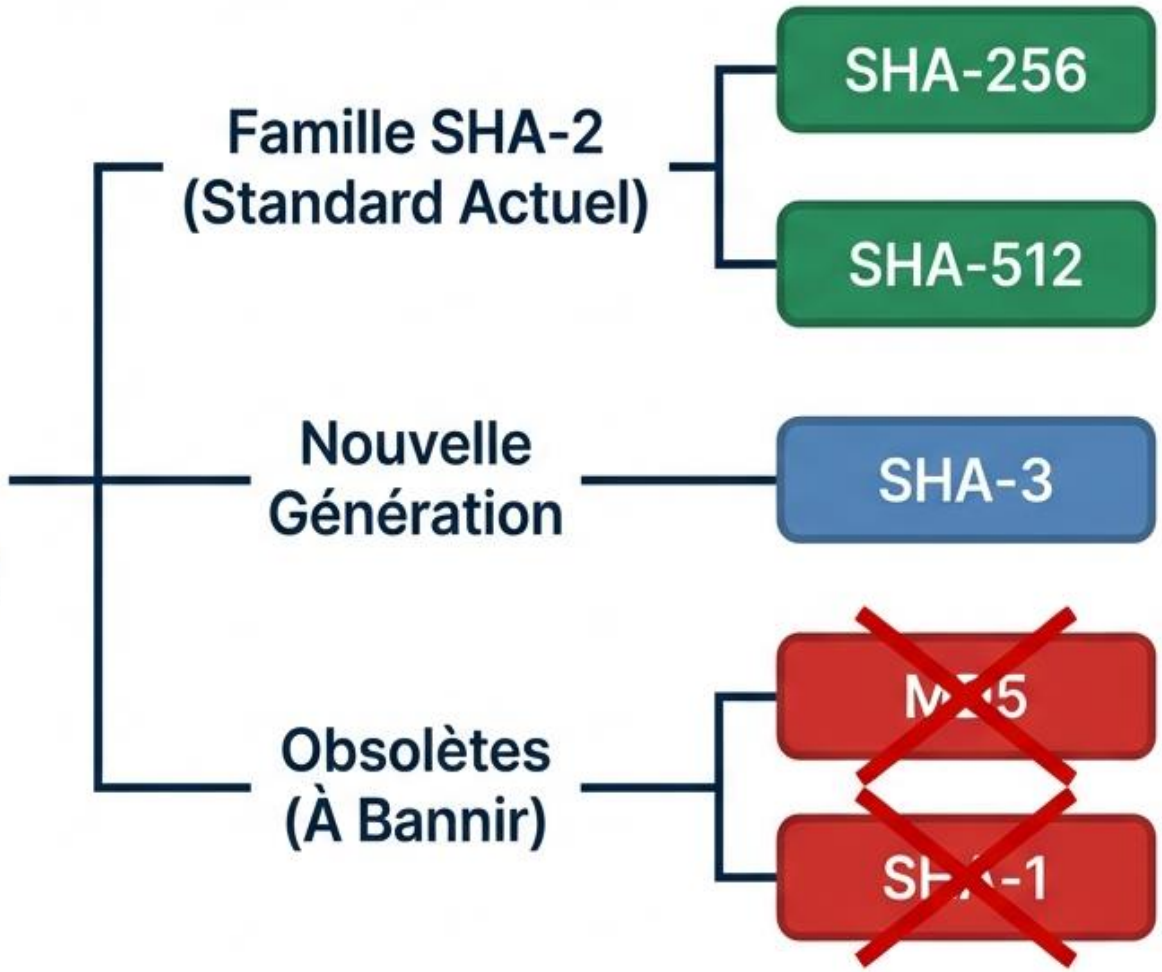
Les 4 Règles d'Or

→ 1. Sens unique (Irréversible)

→ 2. Déterministe (Même message = même empreinte)

→ 3. Effet d'avalanche (Changement mineur = changement majeur)

→ 4. Longueur fixe

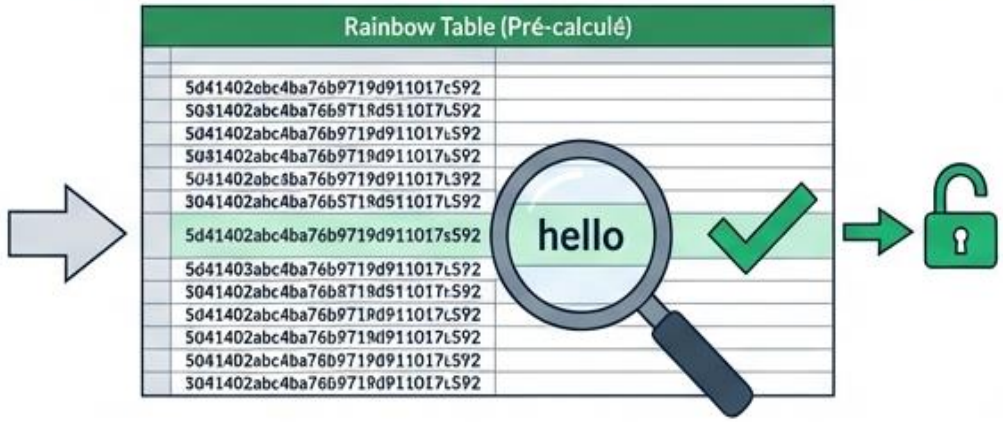


Vulnérabilité - Les Rainbow Tables

Force Brute

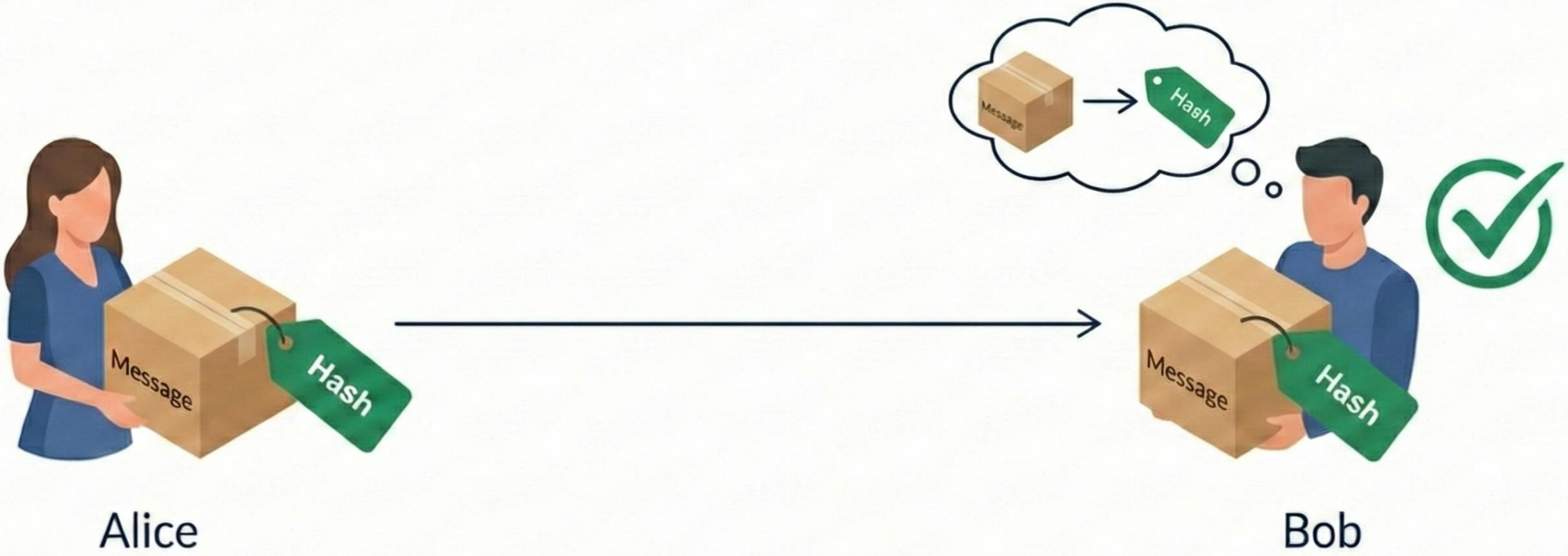
Hash (Empreinte)	Password (Mot de passe)
5d41402abc4ba76b9719d911 17c592	

Rainbow Table (Pré-calculé)



Les pirates utilisent des tables pré-calculées immenses pour retrouver les mots de passe à partir des empreintes, sans avoir à faire le calcul inverse.

Le hachage pour l'intégrité du message

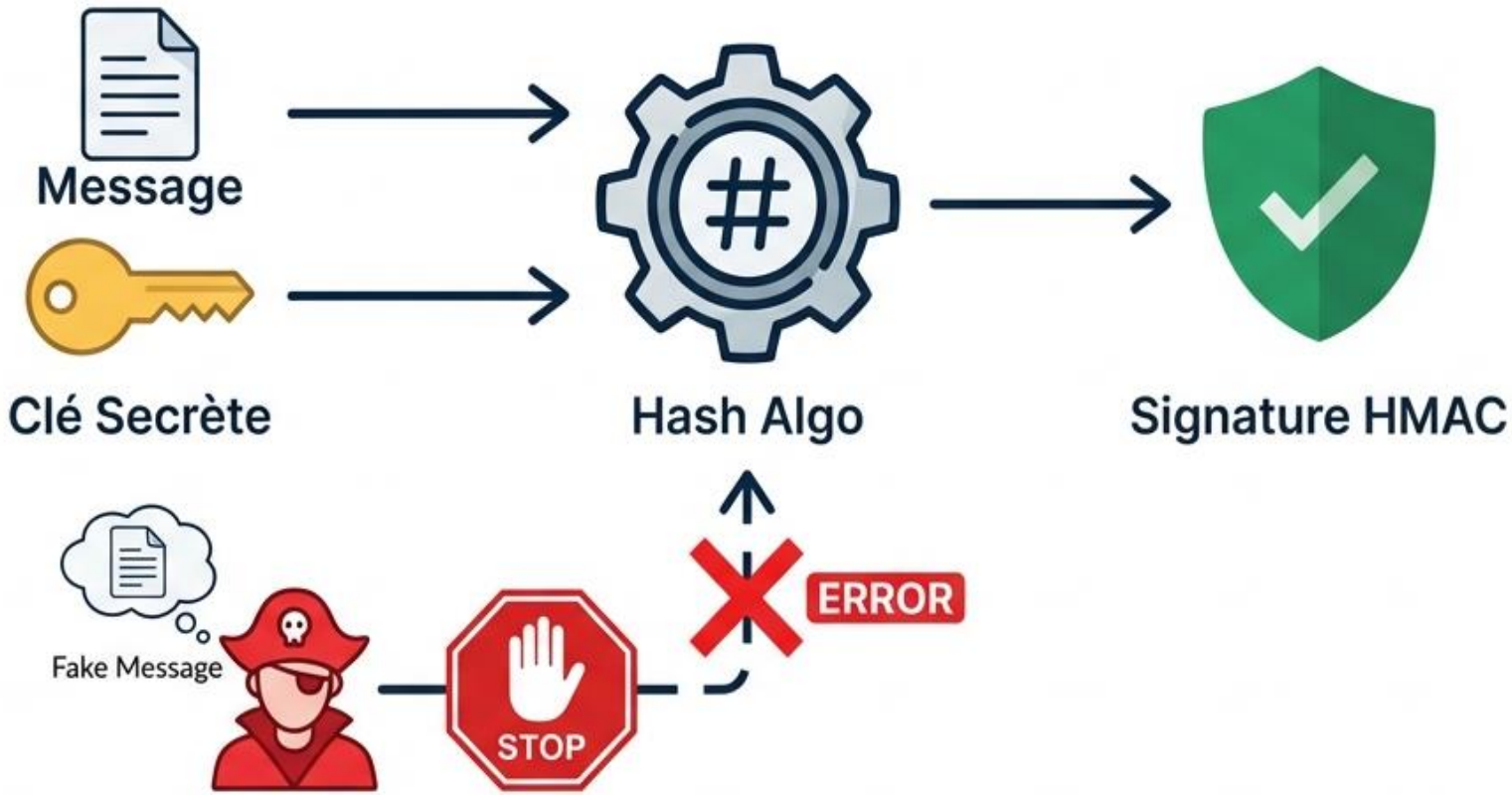


Alice envoie un message haché. Bob reçoit le message, recalcule le hash et le valide car il est identique.

Pourquoi le hachage seul est insuffisant

L'intégrité sans authenticité est inutile face à un pirate actif qui peut recalculer l'empreinte.

La Solution - HMAC (Hash-based Message Authentication Code)



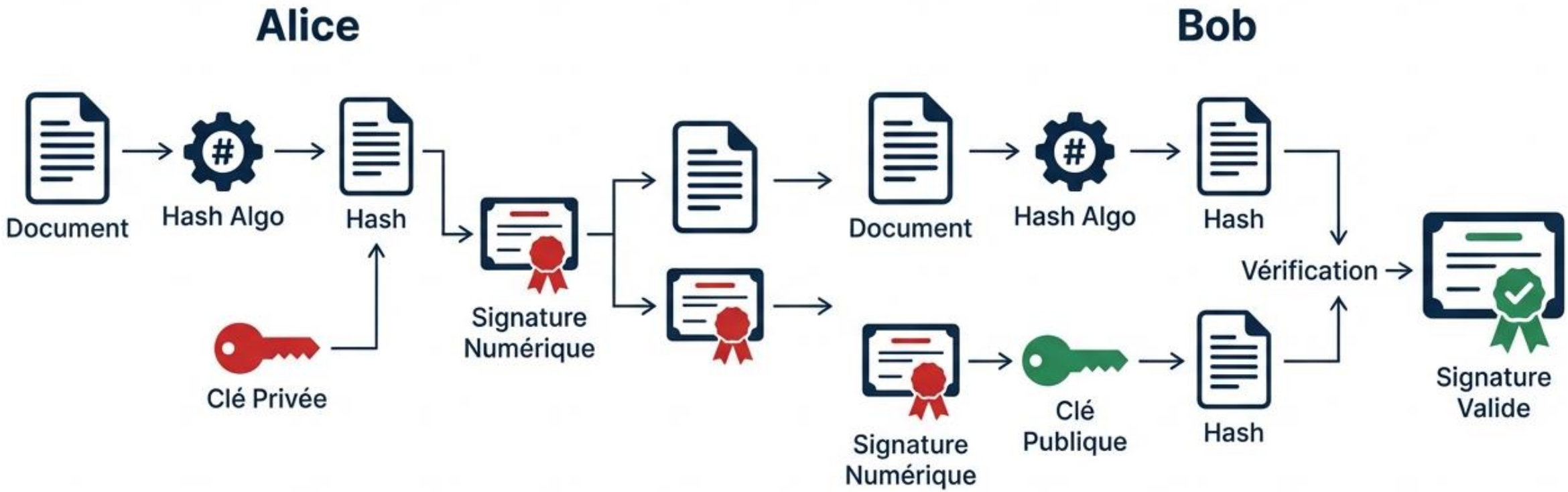
En combinant le message avec une clé secrète partagée, le **pirate ne peut pas générer une signature valide.**

HMAC en Pratique

```
echo -n "Message important" | openssl dgst -sha256 -mac
hmac -macopt hexkey:[CLE_SECRETE]
```

Bonne pratique : La longueur de la clé doit être comparable à la longueur de sortie du hachage (ex: 32 octets pour SHA-256) pour éviter les attaques par force brute sur la clé elle-même.

Pilier 3 - L'Authenticité (Signature Numérique)



Si la clé publique déchiffre la signature, cela prouve que seule Alice (détentrice de la clé privée) a pu l'émettre.

Signature Numérique avec OpenSSL

1. Générer les clés 🔑

```
openssl genpkey -algorithm RSA -out private.pem
```

2. Signer le message 📝

```
openssl dgst -sha256 -sign private.pem -out signature.bin message.txt
```

3. Vérifier la signature ✅

```
openssl dgst -sha256 -verify public.pem -signature signature.bin message.txt
```

Synthèse des Protections

	Le Problème	La Solution
1	Interception (Lecture)	Chiffrement Asymétrique (RSA)
2	Altération (Modification)	Hachage (SHA-256)
3	Imposture (Usurpation)	HMAC ou Signature Numérique

Sécuriser la chaîne de confiance

- Combinez les méthodes : Confidentialité + Intégrité + Authenticité. Aucune méthode ne suffit à elle seule.
- Vérifiez vos algorithmes : Les standards évoluent. Bannissez MD5/SHA-1.
- Protégez vos clés privées à tout prix.