
📘 Cours : Chiffrement Asymétrique (RSA, ECC)

🔐 1. Définition du chiffrement asymétrique

Le **chiffrement asymétrique** utilise **deux clés distinctes mais liées mathématiquement** :

- une **clé publique** pour **chiffrer** les données,
- une **clé privée** pour les **déchiffrer**.

☞ Contrairement au chiffrement symétrique, les deux parties **n'ont pas besoin d'échanger la même clé secrète**.

🎯 2. Objectifs

- Permettre des communications **sécurisées sans échange préalable de clé**.
- Garantir **l'authenticité** grâce à la **signature numérique**.
- Fournir **confidentialité, intégrité et non-répudiation**.

🧠 3. Principe de fonctionnement

🔑 Clés :

- La **clé publique** est **diffusée librement** (ex : sur un site web).
- La **clé privée** est **gardée secrète** par le propriétaire.

🔁 Utilisation :

- Pour chiffrer un message : on utilise la **clé publique du destinataire**.
- Pour déchiffrer : seul le destinataire peut utiliser sa **clé privée**.

🔐 4. Algorithme RSA (Rivest-Shamir-Adleman)

a. Historique

- Créé en 1977 par Ron Rivest, Adi Shamir et Leonard Adleman.
- Basé sur la **difficulté de factoriser de grands nombres premiers**.

b. Fonctionnement simplifié

1. Génération de deux grands **nombres premiers** `p` et `q`.
2. Calcul de `n = p × q` (modulus) et `φ(n) = (p−1)(q−1)`.
3. Choix d'un **exposant public** `e` (souvent 65537).
4. Calcul de l'**exposant privé** `d` tel que :
   `d × e ≡ 1 mod φ(n)`

Chiffrement :

- Message `M`, chiffré en `C = M^e mod n`

Déchiffrement :

- `M = C^d mod n`

c. Avantages / Inconvénients

Avantages	Inconvénients
Largement utilisé (HTTPS, email)	Lenteur sur de grands fichiers
Bonne sécurité (>2048 bits)	Nécessite des grandes clés
Signature et chiffrement possibles	Plus lourd que ECC

---

5. Algorithme ECC (Elliptic Curve Cryptography)

a. Principe

- Basé sur les **courbes elliptiques** sur des champs finis.
- Plus **complexe mathématiquement**, mais **plus léger** que RSA.
- Offrant un **niveau de sécurité élevé avec des clés plus petites**.

b. Fonctionnement

- Calcul de points sur une courbe elliptique.
- Basé sur le problème difficile du **logarithme discret sur courbe elliptique**.

c. Exemples de courbes

- **secp256k1** : utilisé par Bitcoin.
- **Curve25519** : très rapide et sécurisé.

d. Comparaison avec RSA

Critère	RSA (2048 bits)	ECC (256 bits)
Sécurité équivalente	Oui	Oui
Taille des clés	Grande	Petite
Vitesse	Plus lent	Plus rapide
Ressources	Plus gourmand	Moins gourmand

🔐 6. Utilisations concrètes RSA / ECC

- **RSA** :
  - Chiffrement d'e-mails (PGP, S/MIME)
  - Échanges de clés dans TLS (HTTPS)
  - Signature de logiciels
- **ECC** :
  - Sécurisation mobile / IoT
  - Cryptomonnaies (Bitcoin, Ethereum)
  - Protocoles modernes (TLS 1.3, SSH)

🛡 7. Forces et limites du chiffrement asymétrique

Forces	Limites
Pas besoin d'échanger de clé secrète	Plus lent que le chiffrement symétrique
Authentification intégrée	Complexité mathématique
Signature numérique possible	Attaques si mauvaise implémentation

---

📌 8. Résumé final

Élément	RSA	ECC
Basé sur	Factorisation	Courbes elliptiques
Taille clé	Longue (2048-4096 bits)	Courte (256-521 bits)
Performance	Plus lent	Très rapide
Usage courant	HTTPS, e-mails	SSL/TLS, cryptomonnaies
