Cours de Cryptographie

Résumé de Révision pour l'Examen

Basé sur *The Joy of Cryptography* (Mike Rosulek)

2026-01-12

Table des matières

1 Chapitre 1 : Introduction & Sécurité Parfaite

1.1 Concepts Fondamentaux

Définitions de Base

- **Plaintext** : Message original $m \in \mathcal{M}$
- **Ciphertext** : Message chiffré $c \in \mathcal{C}$
- **Key Space** : Ensemble des clés $\mathcal{K}$
- **Encryption** : Enc : $\mathcal{K} \times \mathcal{M} \to \mathcal{C}$
- **Decryption** : Dec : $\mathcal{K} \times \mathcal{C} \to \mathcal{M}$

1.1.1 Principes de Kerckhoffs

> *"La sécurité d'un système cryptographique ne doit reposer que sur le secret de la clé, et non sur celui de l'algorithme."*

Conséquences :

- Algorithmes publics (auditables)
- Seule la clé doit rester secrète
- Permet l'analyse académique

1.2 Sécurité Parfaite (Perfect Security)

Définition de Shannon (1949)

Un schéma a la **sécurité parfaite** si pour tout $m_0, m_1 \in \mathcal{M}$ et tout $c \in \mathcal{C}$:

$$\Pr[C = c \mid M = m_0] = \Pr[C = c \mid M = m_1]$$

Interprétation : L'observation du ciphertext ne donne **AUCUNE** information sur le plaintext.

1.2.1 One-Time Pad (OTP)

Construction :

- $\text{Enc}(k, m) = m \oplus k$
- $\text{Dec}(k, c) = c \oplus k$
- Key space : $\mathcal{K} = \{0,1\}^n$

Propriétés :

- ✓ Sécurité parfaite (prouvée)
- ✓ Extrêmement rapide (opération XOR)
- × Clé aussi longue que le message
- × Clé à usage unique

Théorème de Shannon

Si un schéma de chiffrement a la sécurité parfaite, alors $|\mathcal{K}| \geq |\mathcal{M}|$
Conséquence : La sécurité parfaite nécessite des clés au moins aussi longues que les messages.

ATTENTION

Attaque Two-Time Pad : Réutiliser la même clé k pour deux messages m_1, m_2 :

$$c_1 \oplus c_2 = m_1 \oplus m_2$$

L'adversaire obtient le XOR des deux messages !
Cas réel : Projet VENONA (1940s) - déchiffrement de messages soviétiques.

2 Chapitre 2 : Cryptographie Symétrique

2.1 Transition : Perfect → Computational Security

Perfect Security	Computational Security
Inconditionnelle	Contre adversaires polynomiaux
Clés ≥ message	Clés courtes (128-256 bits)
Coût : gestion clés	Coût : hypothèses mathématiques

2.2 Pseudorandom Generators (PRG)

Définition

PRG : $\{0,1\}^\lambda \to \{0,1\}^n$ où $n \gg \lambda$ (expansion)
Propriété : La sortie est **indistinguable** d'une chaîne vraiment aléatoire.

Exemples :

× **LCG** (Linear Congruential) - **DANGEREUX**
✓ **ChaCha20** - Recommandé (TLS 1.3, WireGuard)
✓ **AES-CTR** - Standard
✓ **DRBG** (NIST) - Génération aléatoire

2.3 Block Ciphers : AES

Paramètres AES :

— Taille de bloc : **128 bits**
— Tailles de clé : **128, 192, 256 bits**
— Rounds : **10, 12, 14**

Opérations par round :

1. **SubBytes** : S-box (inversion dans GF(2^8) + affine)
2. **ShiftRows** : Permutation des lignes
3. **MixColumns** : Matrice MDS dans GF(2^8)
4. **AddRoundKey** : XOR avec sous-clé

2.4 Modes Opératoires

Mode	CPA-secure	Parallèle	Statut
ECB	×	Oui	**JAMAIS**
CBC	✓	Déchiffrement	OK
CTR	✓	Oui	**Recommandé**
OFB	✓	Non	Moins utilisé

CTR (Counter Mode) :

$$c_i = E_k(\text{nonce}\|\text{counter}_i) \oplus m_i$$

Propriétés :

✓ Parallélisable

✓ Accès aléatoire

✓ Pas de padding

Nonce JAMAIS réutilisé

3 Chapitre 3 : Intégrité des Messages

3.1 Message Authentication Codes (MAC)

Sécurité : UF-CMA

Unforgeability under Chosen Message Attack
Adversaire peut demander tags pour messages de son choix, puis doit forger un tag pour un nouveau message.
Sécurité : $\Pr[\text{Succès}] \leq \epsilon$ (négligeable)

3.1.1 HMAC

$$\text{HMAC}_k(m) = H((k \oplus \text{opad})\|H((k \oplus \text{ipad})\|m))$$

où opad = 0x5c5c...5c et ipad = 0x3636...36

Propriétés :

✓ Standard (RFC 2104)

✓ Sécurité prouvée

✓ Utilisé partout (TLS, SSH, IPsec, JWT)

3.2 Fonctions de Hachage

3.2.1 Paradoxe des Anniversaires

Théorème

Pour une fonction de hachage à n bits, trouver une collision nécessite $\approx 2^{n/2}$ évaluations (pas 2^n).
Exemple : SHA-256 (256 bits) $\rightarrow$ sécurité 2^{128} contre collisions.

3.2.2 Construction Merkle-Damgård

Algorithme :

1. **Padding** : $m' = m\|1\|0^k\|\langle|m|\rangle_{64}$
2. **Découpage** : $m' = m_1\|m_2\|\cdots\|m_t$
3. **Itération** : $H_i = h(H_{i-1}, m_i)$ avec $H_0 = \text{IV}$

Théorème : Si h résiste aux collisions, alors H aussi.

ATTENTION

Length Extension Attack : Si on connaît $H(m)$, on peut calculer $H(m\|\text{suffix})$ sans connaître m !
Conséquence : $H(k\|m)$ n'est PAS un MAC sécurisé.

3.3 Authenticated Encryption (AEAD)

Composition	Sécurité	Exemple
Encrypt-and-MAC	×	SSH (ancien)
MAC-then-Encrypt		TLS 1.0 (padding oracle)
Encrypt-then-MAC	✓	IPsec

Schémas AEAD modernes :

— **AES-GCM** : CTR + GHASH (standard, TLS 1.3)
— **ChaCha20-Poly1305** : Stream cipher + MAC (mobile, WireGuard)
— **AES-CCM** : CBC-MAC + CTR (WPA2, Bluetooth)
— **ASCON** : Construction éponge (IoT, NIST 2023)

4 Chapitre 4 : Cryptographie à Clé Publique

4.1 Problèmes Difficiles

Hypothèses de Difficulté

$$\text{DLP} \Rightarrow \text{CDH} \Rightarrow \text{DDH}$$

— **DLP** : Logarithme Discret (trouver a depuis g^a)
— **CDH** : Calculer g^{ab} depuis g^a, g^b
— **DDH** : Distinguer (g^a, g^b, g^{ab}) de (g^a, g^b, g^c)

ATTENTION

DDH est **facile** dans $\mathbb{Z}_p^*$ entier (symbole de Legendre) !
Solution : Travailler dans sous-groupe d'ordre premier q de $\mathbb{Z}_p^*$.

4.2 Diffie-Hellman Key Exchange

Protocole :

1. Alice : Choisit a, envoie g^a
2. Bob : Choisit b, envoie g^b
3. Clé partagée : $k = g^{ab}$

Sécurité passive : CDH suffit

ATTENTION

Vulnérable à Man-in-the-Middle !
Solution : Authenticated DH (certificats, signatures)

4.3 RSA

4.3.1 Génération de Clés

1. Choisir deux grands premiers p, q (1024 bits chacun pour RSA-2048)
2. $n = p \cdot q$
3. $\phi(n) = (p-1)(q-1)$
4. Choisir e tel que $\gcd(e, \phi(n)) = 1$ (souvent $e = 65537$)
5. Calculer $d = e^{-1} \bmod \phi(n)$
6. $pk = (n, e)$, $sk = (n, d)$

4.3.2 RSA-OAEP (Sécurisé)

OAEP-Encode(m) :

1. Choisir $r \xleftarrow{\$} \{0,1\}^{k_0}$
2. $s = (m \| 0^{k_1}) \oplus G(r)$
3. $t = r \oplus H(s)$
4. Retourner $s \| t$

Propriétés :

✓ CPA-secure (modèle oracle aléatoire)

✓ Standard PKCS#1 v2.2

Taille message limitée (190 octets pour RSA-2048)

4.4 Signatures Numériques

4.4.1 DSA (Digital Signature Algorithm)

Sign(sk, m) :

1. Choisir $k \xleftarrow{\$} \mathbb{Z}_q^*$ (**unique et aléatoire !**)
2. $r = (g^k \bmod p) \bmod q$
3. $s = k^{-1} \cdot (H(m) + x \cdot r) \bmod q$
4. Retourner (r, s)

ATTENTION

CRITIQUE : Nonce Reuse Attack !
Si deux signatures utilisent le même k :

— On retrouve k depuis $s_1 - s_2$

— On retrouve la clé secrète x !

Cas réels : PlayStation 3 (2010), Bitcoin wallets (2013)

4.4.2 Comparaison Algorithmes

Algorithme	Taille clé	Performance	Statut
RSA-2048	2048 bits	Lent	OK
ECDSA-P256	256 bits	Rapide	Standard
Ed25519	256 bits	Très rapide	**Recommandé**

5 Chapitre 5 : Communication Anonyme

5.1 Mixnets (Chaum 1981)

Principe : Serveurs intermédiaires qui mélangent les messages.
Chaque Mix :

1. Déchiffre sa couche
2. Attend d'accumuler N messages (batch)
3. Mélange aléatoirement
4. Envoie au prochain hop

Propriétés :

✓ Anonymat si **au moins 1 mix honnête**

× Haute latence (attente du batch)

5.2 Tor (Onion Routing)

Différence avec mixnets :

— Pas de batching (faible latence)

— Circuit persistant

— Chiffrement en couches (oignon)

Architecture :

— **Guard nodes** : Premier relais (stable)

— **Middle relays** : Relais intermédiaires

— **Exit nodes** : Dernier relais (sortie vers Internet)

— **Hidden services** : .onion (serveurs anonymes)

Propriétés :

✓ Faible latence (2-3Œ connexion directe)

✓ 7000 relais, 2 millions d'utilisateurs

Vulnérable si adversaire global (traffic correlation)

5.3 Attaques

Attaque	Cible	Mitigation
Traffic correlation	Tor	Padding, cover traffic
Website fingerprinting	Tor	Padding, multiplexing
(n-1) attack	Mixnet	Batches grands
Sybil attack	P2P, Tor	Sélection aléatoire

6 Formules Essentielles

6.1 Algèbre Modulaire

— $a \equiv b \pmod{n} \iff n \mid (a-b)$

— **Inverse** : $a \cdot a^{-1} \equiv 1 \pmod{n}$

— **Euler** : Si $\gcd(a,n) = 1$, alors $a^{\phi(n)} \equiv 1 \pmod{n}$

— **Fermat** : Si p premier, $a^{p-1} \equiv 1 \pmod{p}$

6.2 Probabilités

— **Indépendance** : $\Pr[A \cap B] = \Pr[A] \cdot \Pr[B]$

— **Conditionnelle** : $\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]}$

6.3 Sécurité

— **Avantage** : $\mathrm{Adv}(\mathcal{A}) = \left|\Pr[\mathcal{A} \text{ gagne}] - \frac{1}{2}\right|$

— **Négligeable** : $\epsilon(\lambda) = o(1/\lambda^c)$ pour tout $c > 0$

7 Bonnes Pratiques

7.1 À FAIRE

Primitives :

✓ AES-256-GCM

✓ ChaCha20-Poly1305

✓ SHA-256, SHA-3

✓ HMAC-SHA256

✓ Ed25519

✓ ECDH-X25519

Règles :

✓ AEAD obligatoire

✓ Nonce unique

✓ Clés $\geq$ 128 bits

✓ Bibliothèques auditées

✓ Encrypt-then-MAC

7.2 À ÉVITER ABSOLUMENT

Algorithmes cassés :

× MD5, SHA-1

× DES, 3DES, RC4

× RSA $<$ 2048 bits

Erreurs courantes :

× Mode ECB

× Réutiliser nonce

× Chiffrer sans authentifier

× RSA textbook

× Implémenter sa propre crypto

8 Checklist Avant l'Examen

8.1 Concepts Théoriques

☐ Définir sécurité parfaite (Shannon)

☐ Théorème de Shannon ($|\mathcal{K}| \geq |\mathcal{M}|$)

☐ Distinguer PRG, PRF, PRP

☐ Jeux de sécurité : IND-CPA, UF-CMA, CDH, DDH

☐ Paradoxe des anniversaires ($2^{n/2}$)

☐ Construction Merkle-Damgård + théorème

8.2 Attaques

☐ Two-Time Pad (réutilisation nonce)

☐ Birthday attack sur hash

☐ Length extension (Merkle-Damgård)

☐ Padding oracle (CBC + MAC-then-Encrypt)

☐ Nonce reuse DSA/ECDSA $\rightarrow$ clé secrète !

☐ Man-in-the-Middle (DH)

☐ Traffic correlation (Tor)

8.3 Protocoles

☐ One-Time Pad : Enc, Dec, propriétés

☐ Modes AES : ECB ×, CBC ✓, CTR ✓

☐ HMAC : Construction, sécurité

☐ AES-GCM : CTR + GHASH

☐ Diffie-Hellman : Protocole, CDH, MITM

☐ RSA : Gen, Enc/Dec, OAEP

☐ DSA : Sign, Vrfy, nonce reuse

☐ Tor : Circuit, onion layers

8.4 Comparaisons

- ☐ Perfect vs Computational Security
- ☐ Stream cipher vs Block cipher
- ☐ MAC vs Hash vs Signature
- ☐ Symétrique vs Asymétrique
- ☐ Encrypt-and-MAC vs MAC-then-Encrypt vs Encrypt-then-MAC
- ☐ Mixnets vs Tor
- ☐ RSA vs ECC

9 Conseils pour l'Examen

9.1 Pièges Courants

ATTENTION

Ne pas confondre :

- Hash $\neq$ MAC $\neq$ Signature
- Perfect security $\neq$ Computational security
- CDH $\neq$ DDH (DDH plus fort)
- Encrypt-and-MAC $\neq$ Encrypt-then-MAC
- RSA textbook $\neq$ RSA-OAEP

9.2 Questions Fréquentes

1. **Pourquoi pas ECB ?** $\rightarrow$ Déterministe, révèle patterns, pas CPA-secure
2. **Pourquoi HMAC et pas** $H(k\|m)$ **?** $\rightarrow$ Length extension attack
3. **Pourquoi Encrypt-then-MAC ?** $\rightarrow$ Seule composition toujours sécurisée
4. **Pourquoi RSA-OAEP ?** $\rightarrow$ Textbook déterministe et malleable
5. **Comment éviter nonce reuse DSA ?** $\rightarrow$ Nonce dérivé (RFC 6979) ou Ed25519
6. **Tor garantit l'anonymat ?** $\rightarrow$ Contre adversaire local oui, global non

Bonne chance pour l'examen !

La sécurité repose sur le secret des clés, pas des algorithmes.
— Principes de Kerckhoffs