

# Table of contents

|   |          |
|---|----------|
| <b>Document de Révision : Cryptographie Quantique et Post-Quantique</b> | <b>3</b> |
| Introduction générale . . . . .   | 3        |
| Contexte de la cryptographie quantique . . . . .                        | 3        |
| Ordinateur classique vs Ordinateur quantique . . . . .                  | 3        |
| Ordinateurs classiques . . . . .  | 3        |
| Ordinateurs quantiques . . . . .  | 4        |
| Différence principale . . . . .   | 4        |
| Défis de l'informatique quantique . . . . .                             | 4        |
| Défi 1 : Effondrement (Collapsing) . . . . .                            | 4        |
| Défi 2 : Sensibilité environnementale . . . . .                         | 5        |
| Défi 3 : Portes logiques . . . . .                                      | 5        |
| Défi 4 : Translation d'algorithmes . . . . .                            | 5        |
| Dangers pour la cryptographie actuelle . . . . .                        | 5        |
| Algorithmes asymétriques actuels . . . . .                              | 5        |
| Menace quantique : Algorithme de Shor . . . . .                         | 6        |
| Impact sur AES . . . . .  | 6        |
| Nécessité de solutions post-quantiques . . . . .                        | 6        |
| Notations mathématiques . . . . .                                       | 7        |
| Ensembles de base . . . . .   | 7        |
| Normes et "petits" polynômes . . . . .                                  | 7        |
| Learning Without Errors (LWE) . . . . .                                 | 8        |
| Configuration du problème . . . . .                                     | 8        |
| Problème LWE (sans erreurs) . . . . .                                   | 8        |
| Learning With Errors (LWE) . . . . .                                    | 8        |
| Ajout d'erreurs aléatoires . . . . .                                    | 8        |
| Problème LWE (avec erreurs) . . . . .                                   | 8        |
| Module Learning With Errors (MLWE) . . . . .                            | 9        |
| Problème MLWE . . . . .   | 9        |
| Decision Module Learning With Errors (D-MLWE) . . . . .                 | 9        |
| Problème D-MLWE . . . . .   | 9        |
| Propriétés de D-MLWE . . . . .  | 9        |
| Comparaison MLWE vs D-MLWE . . . . .                                    | 10       |
| Réseaux euclidiens (Lattices) . . . . .                                 | 10       |
| Définition d'un réseau . . . . .  | 10       |
| Problèmes classiques sur les réseaux . . . . .                          | 10       |
| Complexité des problèmes de réseaux . . . . .                           | 11       |
| Lien avec MLWE et D-MLWE . . . . .                                      | 11       |
| CRYSTALS-Kyber : Vue d'ensemble . . . . .                               | 11       |
| Qu'est-ce que Kyber-KEM ? . . . . .                                     | 11       |
| Composantes de Kyber . . . . .  | 12       |

|  |    |
|--|----|
| Kyber-PKE (Public Key Encryption) . . . . .                    | 12 |
| Espace de texte clair . . . . .                                | 12 |
| Propriété importante . . . . .                                 | 13 |
| Schéma général . . . . .                                       | 13 |
| Kyber-KEM : Mécanisme complet . . . . .                        | 13 |
| Fonctions de hachage utilisées . . . . .                       | 13 |
| Structure de Kyber-KEM . . . . .                               | 14 |
| Propriétés de sécurité . . . . .                               | 14 |
| Paramètres de Kyber . . . . .                                  | 15 |
| Fonctions de hachage . . . . .                                 | 15 |
| Tailles de paramètres . . . . .                                | 15 |
| Sécurité de Kyber-KEM . . . . .                                | 15 |
| Base de la sécurité . . . . .                                  | 15 |
| Propriétés des réseaux . . . . .                               | 16 |
| Attaques d'implémentation . . . . .                            | 16 |
| Considérations pratiques . . . . .                             | 16 |
| Crypto-agilité (Crypto-agility) . . . . .                      | 17 |
| Définition . . . . .   | 17 |
| Objectifs d'un système crypto-agile . . . . .                  | 17 |
| Importance pour la transition post-quantique . . . . .         | 17 |
| Recommandations NIST . . . . .                                 | 17 |
| Approche hybride . . . . .                                     | 17 |
| Conclusion sur la crypto-agilité . . . . .                     | 18 |
| Exemple jouet de Kyber-KEM . . . . .                           | 18 |
| Avertissement . . . . .  | 18 |
| Étape 0 : Anneau jouet et représentation . . . . .             | 18 |
| Étape 1 : Génération de clés (Récepteur) . . . . .             | 19 |
| Étape 2 : Encapsulation de clé (Émetteur) - Partie 1 . . . . . | 19 |
| Étape 2 : Encapsulation de clé (Émetteur) - Partie 2 . . . . . | 19 |
| Étape 3 : Décapsulation de clé (Récepteur) . . . . .           | 19 |
| Relation avec le vrai Kyber-KEM . . . . .                      | 20 |
| Flux complet de Kyber-KEM . . . . .                            | 22 |
| Vue d'ensemble du protocole . . . . .                          | 22 |
| Garanties de sécurité . . . . .                                | 23 |
| Avantages et limitations de Kyber . . . . .                    | 23 |
| Avantages . . . . .  | 23 |
| Limitations et défis . . . . .                                 | 23 |
| Comparaison avec d'autres approches post-quantiques . . . . .  | 24 |
| Familles de cryptographie post-quantique . . . . .             | 24 |
| Pourquoi Kyber ? . . . . .                                     | 25 |
| Concepts mathématiques à maîtriser . . . . .                   | 25 |
| Algèbre . . . . .  | 25 |
| Théorie des réseaux . . . . .                                  | 25 |

|   |    |
|---|----|
| Cryptographie . . . . .                     | 26 |
| Problèmes de difficulté . . . . .           | 26 |
| Questions d'examen types . . . . .          | 26 |
| Questions conceptuelles . . . . .           | 26 |
| Questions techniques . . . . .              | 27 |
| Questions pratiques . . . . .               | 27 |
| Recommandations pratiques . . . . .         | 28 |
| Pour la transition post-quantique . . . . . | 28 |
| Pour l'implémentation de Kyber . . . . .    | 28 |
| Conclusion générale . . . . .               | 29 |
| Résumé des points clés . . . . .            | 29 |
| Perspectives futures . . . . .              | 29 |
| Glossaire des termes importants . . . . .   | 30 |

## Document de Révision : Cryptographie Quantique et Post-Quantique

---

### Introduction générale

#### Contexte de la cryptographie quantique

Ce cours aborde les **défis posés par l'informatique quantique** à la cryptographie actuelle et les **solutions post-quantiques** développées pour y faire face, en particulier l'algorithme **CRYSTALS-Kyber**.

#### Problématique centrale :

L'avènement des ordinateurs quantiques menace de briser la sécurité des systèmes cryptographiques actuels, nécessitant une transition vers des algorithmes **résistants aux attaques quantiques**.

---

### Ordinateur classique vs Ordinateur quantique

#### Ordinateurs classiques

##### Unité de base d'information : Bit

- Valeurs possibles : **0** ou **1**

- Nombre **fini** d'états
- États distingués par des états **électriques/magnétiques** dans le matériel
- Opérations effectuées via des **portes logiques**

## Ordinateurs quantiques

### Unité fondamentale d'information : Qubit

Un **qubit** est un petit système quantique avec des propriétés spéciales

#### Propriété 1 : Superposition

- Peut exister dans un **nombre infini d'états** (superposition de ses états de base)
- Lors de la **mesure** → le qubit **s'effondre** (décohérence)

#### Propriété 2 : Intrication (Entanglement)

Système de  $n$  qubits :  $n$  qubits intriqués avec  $2^n$  états de base

#### Implications fondamentales :

- Le nombre d'états de base croît **exponentiellement** :  $2^n$
- Opère sur **tous les états de base simultanément** → “**parallélisme naturel**”

## Différence principale

**Point clé** : La différence principale entre informatique classique et quantique réside dans la **représentation de l'information**

#### Informatique quantique :

- S'inspire du **comportement des particules quantiques**
- Nécessite un **ordinateur quantique** réel (pas de simulation classique efficace)

## Défis de l'informatique quantique

### Défi 1 : Effondrement (Collapsing)

**Problème** : Le qubit perd sa propriété de **superposition** lors de la mesure

**Conséquence** : Les mesures doivent être soigneusement planifiées dans les algorithmes

## Défi 2 : Sensibilité environnementale

**Problème :** Grande **sensibilité** du système quantique à son environnement

**Conséquence :** Nécessite une **infrastructure** très particulière (températures extrêmes, isolation)

## Défi 3 : Portes logiques

**Distinction importante :**

- **Portes quantiques logiques** (théoriques)
- **Portes quantiques physiques** (implémentation réelle)

**Impact :** Influence sur les **performances** des circuits quantiques

## Défi 4 : Translation d'algorithmes

**Exemple :** AES (Advanced Encryption Standard)

La translation directe n'est **pas faisable**

**Solution unique :** Développer des **algorithmes appropriés** → nouveau paradigme

**Conclusion :** Beaucoup de questions non résolues dans le domaine

---

## Dangers pour la cryptographie actuelle

### Algorithmes asymétriques actuels

Sécurité basée sur des **problèmes mathématiques difficiles** pour ordinateurs classiques :

- **Factorisation** des grands nombres (RSA)
- **Logarithmes discrets**
- **Problème de Diffie-Hellman**

## **Menace quantique : Algorithme de Shor**

**Problème majeur :** Ces problèmes sont résolus en **temps polynomial** par l'**algorithme de Shor** sur un ordinateur quantique

**Systèmes affectés :**

- Chiffrement/déchiffrement
- Signatures numériques
- Authentification
- Établissement de clés (key establishment)

**Protocoles menacés :**

- **HTTPS**
- **TLS**
- Et beaucoup d'autres

**“Apocalypse quantique” :** L'Internet tel que nous le connaissons aujourd'hui n'existerait plus

## **Impact sur AES**

**AES vs Algorithme de Grover :**

L'algorithme de Grover menace AES mais de manière moins dramatique

**Solution :** Doubler la taille des clés suffit pour maintenir la sécurité

## **Nécessité de solutions post-quantiques**

**Conclusion :** Besoin de **solutions post-quantiques** basées sur des problèmes mathématiques plus difficiles

---

## Notations mathématiques

### Ensembles de base

#### 1. Entiers modulo $q$ :

$$\mathbb{Z}_q = \{0, 1, 2, \dots, q-1\}$$

L'ensemble des entiers modulo  $q$

#### 2. Polynômes à coefficients dans $\mathbb{Z}_q$ :

Pour  $q$  premier :  $\mathbb{Z}_q[X]$  est l'ensemble des polynômes avec coefficients dans  $\mathbb{Z}_q$

**Note importante** : Le degré n'est **pas borné**

#### 3. Anneau de polynômes (Polynomial ring) :

Pour un entier positif  $n$  :

$$R_q = \mathbb{Z}_q[X]/(X^n + 1)$$

C'est l'anneau des polynômes modulo  $(X^n + 1)$

#### 4. Module :

Pour un entier positif  $k$  :

$$R_q^k$$

Un module de rang  $k$  sur  $R_q$

### Normes et “petits” polynômes

#### 5. Norme infinie :

On peut définir la notion de “**taille**” ou **norme infinie** sur tous ces ensembles

#### 6. “Petits” polynômes :

Polynômes avec des coefficients de petite magnitude (proches de 0)

Ces polynômes jouent un rôle crucial dans la sécurité de Kyber

## Learning Without Errors (LWE)

### Configuration du problème

#### Situation :

Alice possède un **vecteur secret**  $s$  (d'entiers)

#### Information publique :

Ensemble d'équations avec **coefficients entiers** auxquelles  $s$  est solution

### Problème LWE (sans erreurs)

**Objectif :** Trouver  $s$  à partir de ces équations

**Méthode :** Utiliser l'**élimination de Gauss**

**Conclusion :** Facile à résoudre  $\rightarrow$  Ne fournit **aucune sécurité**

---

## Learning With Errors (LWE)

### Ajout d'erreurs aléatoires

**Pour rendre le problème plus difficile :**

Ajouter une **erreur aléatoire** (proche de zéro) au **membre de droite** des équations

### Problème LWE (avec erreurs)

**Objectif :** Trouver  $s$  à partir des équations **avec erreurs ajoutées**

#### Difficulté :

Un système **surdéterminé** d'équations rempli d'erreurs n'a presque certainement **pas de solution** (au sens traditionnel)

**Conclusion :** Maintenant  $s$  est vraiment un **secret**

**Note pratique :** Généralement, les opérations sont effectuées **modulo un grand nombre**

---



## Module Learning With Errors (MLWE)

### Problème MLWE

#### Caractéristiques :

- Extension de LWE aux **modules** sur les anneaux de polynômes
- Travaille avec  $R_q^k$  au lieu de vecteurs d'entiers

#### Propriété fondamentale :

Aucun algorithme efficace n'est connu pour résoudre MLWE

#### Type de problème :

MLWE est un **problème de recherche** (search problem) : trouver  $s$

---

## Decision Module Learning With Errors (D-MLWE)

### Problème D-MLWE

#### Mêmes paramètres que MLWE mais :

Une instance de D-MLWE est donnée par une paire  $(A, t)$

#### Question à résoudre :

Déterminer si  $(A, t)$  est :

- Une instance **valide** de MLWE (où  $t$  dépend d'un secret  $s$ )
- Ou simplement une paire **aléatoire**

#### Propriétés de D-MLWE

#### Difficulté :

Si D-MLWE est difficile  $\rightarrow (A, t)$  ne donne **aucune information** sur  $s$

Aucun algorithme efficace connu pour D-MLWE

#### Type de problème :

D-MLWE est un **problème de décision** (decision problem)

## Comparaison MLWE vs D-MLWE

| Aspect      | MLWE                  | D-MLWE                                 |
|-------------|-----------------------|--|
| Type        | Problème de recherche | Problème de décision                   |
| Objectif    | Trouver $s$           | Distinguer instance valide d'aléatoire |
| Utilisation | Construction de clés  | Preuves de sécurité                    |

## Réseaux euclidiens (Lattices)

### Définition d'un réseau

#### Réseau (Lattice) :

Ensemble de toutes les **combinaisons linéaires entières** de  $n$  vecteurs **linéairement indépendants** dans  $\mathbb{R}^n$  (la base)

#### Notation mathématique :

Si  $\{b_1, b_2, \dots, b_n\}$  est la base, le réseau est :

$$\mathcal{L} = \left\{ \sum_{i=1}^n z_i b_i \mid z_i \in \mathbb{Z} \right\}$$

### Problèmes classiques sur les réseaux

#### Closest Vector Problem (CVP) :

Étant donné un point  $x$  dans  $\mathbb{R}^n$ , trouver le vecteur du réseau **le plus proche** de  $x$

#### Shortest Vector Problem (SVP) :

Trouver un vecteur **non nul** dans le réseau avec **norme euclidienne minimale**

SVP est un **cas particulier** de CVP pour  $x = 0$

## Complexité des problèmes de réseaux

### Propriétés importantes :

- La plupart des problèmes de réseaux sont **NP-difficiles**
- Ils sont **difficiles en moyenne** (average-case hard)

### Différence avec d'autres problèmes NP :

Beaucoup de problèmes NP sont faciles “en moyenne” mais difficiles seulement dans le pire cas. Les problèmes de réseaux sont difficiles **même en moyenne**.

## Lien avec MLWE et D-MLWE

### Reformulation :

MLWE et D-MLWE peuvent être **reformulés comme problèmes de réseaux**

### Difficulté moyenne :

La difficulté moyenne est **au moins celle** de la difficulté quantique du pire cas de certains problèmes de réseaux

### Intérêt pratique :

Ceci est très intéressant pour **évaluer la difficulté** de MLWE et D-MLWE car :

- Les problèmes de réseaux sont bien étudiés
- Leur difficulté est bien comprise
- Même avec un ordinateur quantique, ils restent difficiles

---

## CRYSTALS-Kyber : Vue d'ensemble

### Qu'est-ce que Kyber-KEM ?

Kyber-KEM est un **KEM** (Key Encapsulation Mechanism) **IND-CCA2-secure**

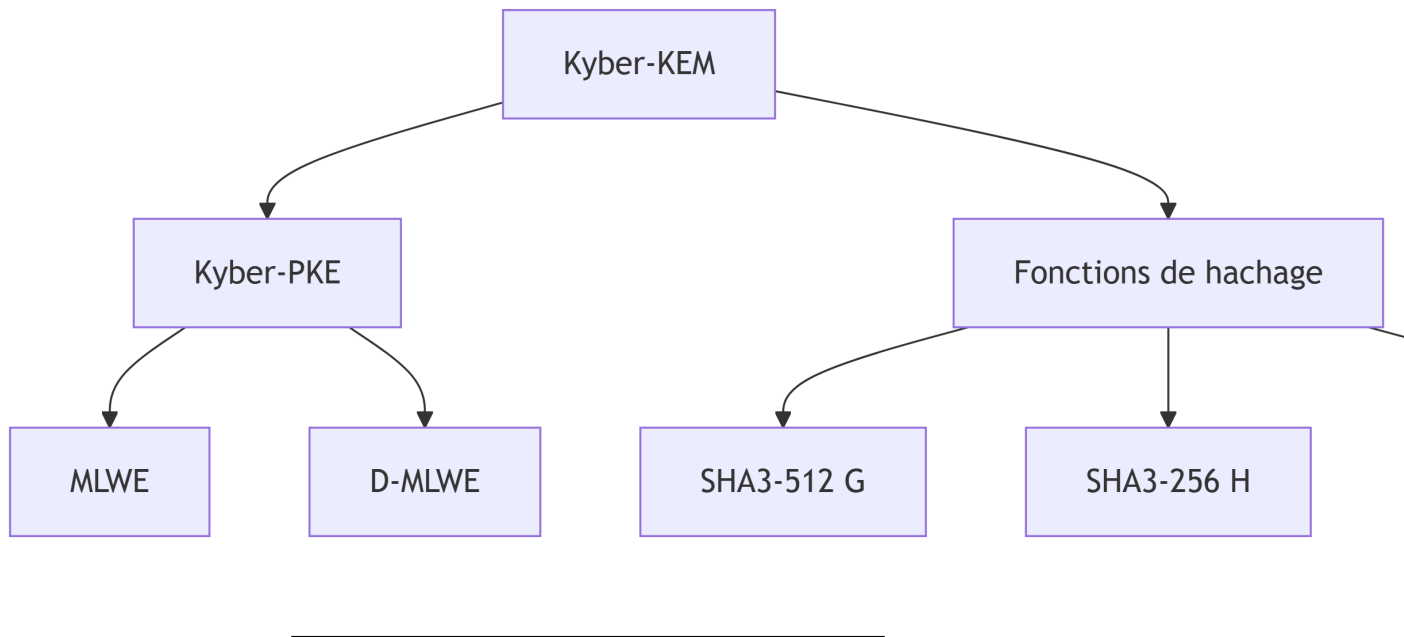
### Caractéristiques principales :

- **Système cryptographique asymétrique**
- Sécurité basée sur la difficulté des problèmes **MLWE** et **D-MLWE**
- Travaille avec :
  - Anneau de polynômes  $R_q$

- Module  $R_q^k$
- Polynômes “petits”
- Cryptosystème **basé sur les réseaux** (lattice-based)

### Composantes de Kyber

Structure en couches :



### Kyber-PKE (Public Key Encryption)

Espace de texte clair

Plaintext space :

$$\{0, 1\}^n \subset R_q$$

Les messages sont des **polynômes** dans  $R_q$  représentant des bits

### Propriété importante

Le déchiffrement peut échouer (avec une **petite probabilité**)

**Raison** : Les erreurs ajoutées peuvent parfois être trop grandes

### Gestion pratique :

Cette probabilité d'échec est rendue **négligeable** par le choix approprié des paramètres

### Schéma général

**Kyber-PKE** est le cœur **cryptographique** utilisé par Kyber-KEM

Il fournit les opérations de base :

- **Génération de clés** (KeyGen)
  - **Chiffrement** (Encrypt)
  - **Déchiffrement** (Decrypt)
- 

### Kyber-KEM : Mécanisme complet

#### Fonctions de hachage utilisées

Trois fonctions de hachage cryptographiques :

- **G** : SHA3-512
- **H** : SHA3-256
- **J** : SHAKE256

Ces fonctions assurent différentes propriétés de sécurité et permettent de dériver des clés

## Structure de Kyber-KEM

### Opérations principales :

#### 1. KeyGen (Génération de clés) :

Génère une paire de clés publique/privée

#### 2. Encapsulation :

- Entrée : Clé publique
- Sortie :
  - Texte chiffré (ciphertext)
  - Clé partagée (shared secret)

#### 3. Decapsulation :

- Entrée : Clé privée + texte chiffré
- Sortie : Clé partagée

## Propriétés de sécurité

### Plaintext awareness IND-CCA

La **conscience du texte clair** (plaintext awareness) implique la sécurité **IND-CCA** (Indistinguishability under Chosen Ciphertext Attack)

**La décapsulation peut échouer** même si honnête (petite probabilité)

### Gestion :

Les paramètres sont choisis pour rendre cette probabilité **négligeable**

---

| Fonction | Algorithme | Usage |
|----------|------------|-------|
|----------|------------|-------|

## Paramètres de Kyber

### Fonctions de hachage

Récapitulatif des fonctions :

| Fonction | Algorithme | Usage                      |
|----------|------------|----------------------------|
| <b>G</b> | SHA3-512   | Dérivation de clés longues |
| <b>H</b> | SHA3-256   | Hachage standard           |
| <b>J</b> | SHAKE256   | Fonction XOF (extensible)  |

### Tailles de paramètres

Kyber propose plusieurs niveaux de sécurité :

Les paramètres varient selon le niveau souhaité, notamment :

- $k$  : rang du module (typiquement 2, 3 ou 4)
- $n$  : degré des polynômes
- $q$  : modulo pour les coefficients

**Compromis** : Plus  $k$  est grand, plus la sécurité est élevée mais plus les clés et calculs sont grands

## Sécurité de Kyber-KEM

### Base de la sécurité

Sécurité IND-CCA sous l'hypothèse :

D-MLWE est **intraitable** (computationally hard)

**Choix des distributions** :

Les distributions d'erreurs sont choisies pour être “**aussi difficiles que**” certains problèmes de réseaux

## Propriétés des réseaux

Réseaux euclidiens sont :

- NP-difficiles
- Difficiles en moyenne (average-case hard)

Ceci fournit une **base solide** pour la sécurité à long terme

## Attaques d'implémentation

Préoccupations pratiques :

Attaques par canaux auxiliaires (side-channel attacks) :

- Analyse de la consommation électrique
- Analyse temporelle
- Analyse électromagnétique

Attaques par injection de fautes (fault attacks) :

- Perturbation délibérée du calcul
- Observation du comportement erroné

Défis :

- Difficile à prévoir toutes les attaques
- Nouvelles techniques d'attaque en développement
- **Soyez prudent** car les algorithmes sont nouveaux

## Considérations pratiques

Points importants :

1. La sécurité théorique est bien fondée
2. L'implémentation pratique nécessite des **contre-mesures**
3. Évaluation continue de la sécurité nécessaire



## Crypto-agilité (Crypto-agility)

### Définition

#### Crypto-agilité :

Capacité de **basculer entre différentes primitives cryptographiques** facilement

#### Objectifs d'un système crypto-agile

##### Permettre :

- **Adaptations rapides** de nouveaux algorithmes sans perturber l'infrastructure
- Sécurité en cas de primitives **cassées/vulnérables**
- Transition en douceur vers de nouvelles solutions

#### Importance pour la transition post-quantique

##### Pourquoi est-ce crucial ?

Se préparer à l'avènement d'un ordinateur quantique consistera en une **migration massive** (probablement la **plus grande de l'histoire**)

#### Recommandations NIST

##### Conseil du NIST :

**Ne pas abandonner** les algorithmes classiques

##### Pourquoi ?

1. **Transition plus douce** + optimisations actuelles conservées
2. **Nouvelles mathématiques** : sécurité empirique, pas encore complètement testée
3. **Incertitude temporelle** : on ne sait pas quand un ordinateur quantique sera réalisé

#### Approche hybride

##### Solution recommandée :

**Combiner** algorithmes classiques et post-quantiques :

- Sécurité garantie si **au moins un** des deux est sûr
- Transition progressive possible
- Flexibilité selon les applications

## Conclusion sur la crypto-agilité

Considérer des systèmes crypto-agiles est très important pour :

- Permettre une **transition en douceur** vers le post-quantique
  - Concevoir des systèmes **flexibles** proposant :
    - Différentes primitives
    - Différents niveaux de sécurité
    - Adaptation selon l'application
- 

## Exemple jouet de Kyber-KEM

### Avertissement

**Important :**

Les paramètres sont **minuscules** et pour **illustration uniquement**

Ils ne sont **PAS** cryptographiquement sûrs

**Objectif :** Comprendre les calculs derrière Kyber-KEM sans se perdre dans les grands nombres

### Étape 0 : Anneau jouet et représentation

**Définition de l'anneau jouet :**

Petit anneau de polynômes avec paramètres réduits pour faciliter les calculs à la main

**Exemple :**

- $n$  très petit (par ex.  $n = 4$ )
- $q$  petit (par ex.  $q = 17$ )
- $R_q = \mathbb{Z}_q[X]/(X^n + 1)$

## Étape 1 : Génération de clés (Récepteur)

Le récepteur génère :

- **Clé secrète** : vecteur/polynôme secret  $s$  avec petits coefficients
- **Clé publique** : paire  $(A, t)$  où :
  - $A$  est une matrice/vecteur aléatoire publique
  - $t = As + e$  où  $e$  est une petite erreur

La sécurité repose sur la difficulté de retrouver  $s$  à partir de  $(A, t)$

## Étape 2 : Encapsulation de clé (Émetteur) - Partie 1

L'émetteur :

1. Choisit un message aléatoire  $m$  (qui deviendra la clé partagée)
2. Choisit un vecteur aléatoire court  $r$  et des erreurs  $e_1, e_2$

## Étape 2 : Encapsulation de clé (Émetteur) - Partie 2

L'émetteur calcule :

- $u = A^T r + e_1$  : première partie du chiffré
- $v = t^T r + e_2 + \text{encode}(m)$  : seconde partie du chiffré

Sortie :

- **Ciphertext** :  $(u, v)$
- **Clé partagée** : dérivée de  $m$  via fonction de hachage

## Étape 3 : Décapsulation de clé (Récepteur)

Le récepteur :

1. Calcule  $v - s^T u$
2. Remarque que :

$$v - s^T u = t^T r + e_2 + \text{encode}(m) - s^T (A^T r + e_1)$$

$$= (As + e)^T r + e_2 + \text{encode}(m) - s^T A^T r - s^T e_1$$

$$= \text{encode}(m) + (\text{petites erreurs})$$

3. **Décode** pour retrouver  $m$  (si les erreurs sont assez petites)
4. Dérive la **clé partagée** de  $m$

### Relation avec le vrai Kyber-KEM

Dans le vrai Kyber :

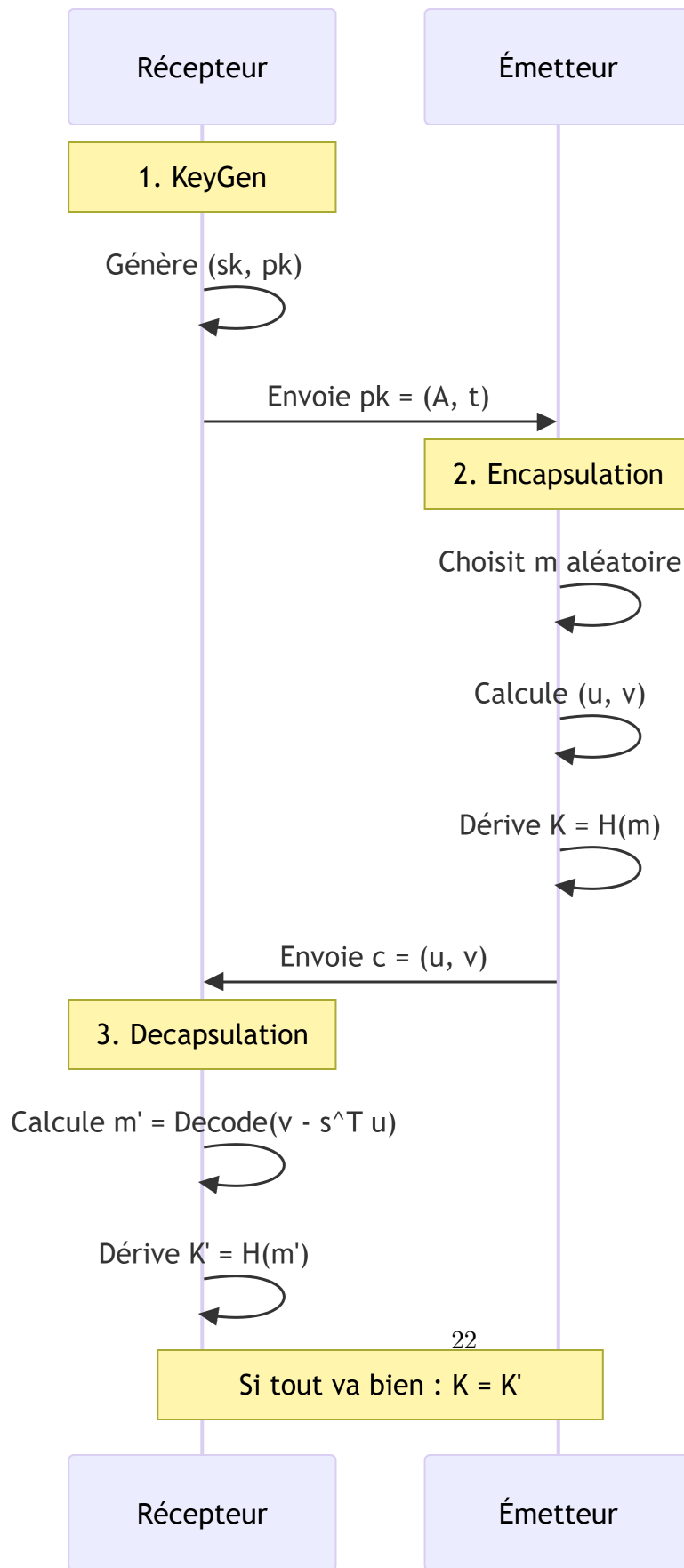
- Les polynômes vivent dans un anneau **beaucoup plus grand**
- Il y a des **vecteurs de polynômes** (module de rang  $k = 2, 3$  ou  $4$ )
- Des **termes de bruit** petits sont ajoutés
- La structure globale suit le **même schéma** :
  - Clé publique de style **module-LWE** :  $(A, t)$
  - Étape d'**encapsulation** utilisant un court  $r$
  - Étape de **décapsulation** qui annule le terme principal et récupère le message (modulo petit bruit)

---



## Flux complet de Kyber-KEM

### Vue d'ensemble du protocole



## Garanties de sécurité

Si D-MLWE est difficile :

1. Un adversaire ne peut pas distinguer  $(A, t)$  d'aléatoire
2. Un adversaire ne peut pas retrouver  $s$  de  $(A, t)$
3. Un adversaire ne peut pas retrouver  $m$  de  $(u, v)$  sans  $s$
4. La clé partagée  $K$  est indistinguable d'une clé aléatoire

**Résultat :** Sécurité **IND-CCA2** (la plus forte notion de sécurité pour KEM)

---

## Avantages et limitations de Kyber

### Avantages

**Sécurité post-quantique :**

Résistant aux attaques d'ordinateurs quantiques

**Performances :**

- Clés relativement petites
- Calculs efficaces
- Peut être implémenté sur matériel limité

**Fondements mathématiques solides :**

Basé sur problèmes de réseaux bien étudiés

**Standardisation :**

Sélectionné par le NIST pour standardisation

### Limitations et défis

**Tailles de clés :**

Plus grandes que RSA/ECC classiques (mais acceptables)

**Nouveauté :**

- Moins de temps pour analyse cryptanalytique
- Implémentations peuvent avoir des vulnérabilités

### **Attaques d'implémentation :**

Nécessite des contre-mesures spécifiques

### **Probabilité d'échec :**

Très faible mais non nulle (contrairement aux schémas classiques)

---

## **Comparaison avec d'autres approches post-quantiques**

### **Familles de cryptographie post-quantique**

#### **1. Basés sur les réseaux (Lattice-based) :**

- **Kyber** (KEM)
- Dilithium (signatures)
- **Avantages** : Efficaces, bien étudiés
- **Kyber appartient à cette famille**

#### **2. Basés sur les codes (Code-based) :**

- Classic McEliece
- **Avantages** : Très mature
- **Inconvénients** : Très grandes clés

#### **3. Basés sur les hash (Hash-based) :**

- SPHINCS+
- **Avantages** : Sécurité minimale bien comprise
- **Inconvénients** : Signatures volumineuses

#### **4. Basés sur les isogénies (Isogeny-based) :**

- SIKE (cassé en 2022)
- **Inconvénients** : Moins mature, vulnérabilités découvertes



## Pourquoi Kyber ?

Kyber a été choisi par le NIST pour :

- Bon équilibre **performance/sécurité**
  - Flexibilité (plusieurs niveaux de sécurité)
  - Implémentations efficaces possibles
  - Fondements mathématiques robustes
- 

## Concepts mathématiques à maîtriser

### Algèbre

Structures algébriques :

- Anneaux de polynômes
- Modules
- Opérations modulo

Polynômes :

- Addition, multiplication de polynômes
- Réduction modulo  $X^n + 1$
- Coefficients modulo  $q$

### Théorie des réseaux

Concepts fondamentaux :

- Base d'un réseau
- Combinaisons linéaires entières
- SVP et CVP
- Difficulté NP et average-case

## **Cryptographie**

### **Primitives :**

- KEM (Key Encapsulation Mechanism)
- PKE (Public Key Encryption)
- Sécurité IND-CCA2

### **Fonctions de hachage :**

- SHA3-256, SHA3-512
- SHAKE256 (XOF)

### **Problèmes de difficulté**

#### **LWE et variantes :**

- Learning With Errors
- Module-LWE
- Decision-MLWE
- Lien avec les réseaux

---

## **Questions d'examen types**

### **Questions conceptuelles**

#### **Ordinateurs quantiques :**

- Différence entre bit et qubit
- Qu'est-ce que la superposition ?
- Qu'est-ce que l'intrication ?
- Quels sont les défis de l'informatique quantique ?

#### **Menaces :**

- Pourquoi les ordinateurs quantiques menacent-ils la cryptographie actuelle ?
- Qu'est-ce que l'algorithme de Shor ?
- Quels systèmes sont affectés ?

#### **Solutions post-quantiques :**

- Qu'est-ce que la cryptographie post-quantique ?
- Pourquoi les réseaux sont-ils intéressants ?
- Qu'est-ce que CRYSTALS-Kyber ?

## Questions techniques

### LWE et MLWE :

- Différence entre LWE et Learning Without Errors
- Qu'est-ce que MLWE ?
- Différence entre MLWE (search) et D-MLWE (decision)

### Réseaux :

- Définir un réseau euclidien
- Qu'est-ce que SVP et CVP ?
- Pourquoi sont-ils NP-difficiles ?

### Kyber :

- Structure de Kyber-KEM
- Rôle de Kyber-PKE
- Fonctions de hachage utilisées

## Questions pratiques

### Sécurité :

- Sur quoi repose la sécurité de Kyber ?
- Qu'est-ce que IND-CCA2 ?
- Quelles sont les attaques d'implémentation possibles ?

### Crypto-agilité :

- Pourquoi est-elle importante ?
- Recommandations du NIST
- Approche hybride classique/post-quantique

### Exemple jouet :

- Expliquer les étapes de KeyGen, Encapsulation, Decapsulation
- Pourquoi le déchiffrement peut-il échouer ?
- Comment récupère-t-on le message ?

## Recommandations pratiques

### Pour la transition post-quantique

#### 1. Inventaire :

Identifier tous les systèmes utilisant la cryptographie asymétrique

#### 2. Priorisation :

Prioriser les systèmes critiques pour la migration

#### 3. Crypto-agilité :

Concevoir les systèmes pour permettre le changement d'algorithmes

#### 4. Approche hybride :

Combiner algorithmes classiques et post-quantiques pendant la transition

#### 5. Tests :

Tester les implémentations contre les attaques d'implémentation

### Pour l'implémentation de Kyber

#### Sécurité :

- Implémenter des **contre-mesures** contre les canaux auxiliaires
- Utiliser des **générateurs aléatoires cryptographiques** de qualité
- Valider tous les **paramètres** en entrée

#### Performance :

- Optimiser les opérations sur polynômes
- Utiliser des implémentations vectorisées si possible
- Considérer l'utilisation de matériel dédié

#### Conformité :

- Suivre les **spécifications NIST**
- Utiliser des implémentations **certifiées** si possible
- Rester à jour avec les recommandations

## Conclusion générale

### Résumé des points clés

#### Menace quantique :

Les ordinateurs quantiques menacent la cryptographie actuelle basée sur factorisation et logarithmes discrets

#### Solution : Kyber-KEM :

- Mécanisme d'encapsulation de clés **post-quantique**
- Basé sur la difficulté de **MLWE** et **D-MLWE**
- Fondé sur les **problèmes de réseaux** (NP-difficiles et average-case hard)
- Sécurité **IND-CCA2** prouvée

#### Transition :

- Nécessite une **crypto-agilité**
- Approche **hybride** recommandée
- Plus grande **migration cryptographique** de l'histoire

### Perspectives futures

#### Défis à venir :

- Optimisation des implémentations
- Analyse cryptanalytique continue
- Standardisation complète
- Déploiement à grande échelle

#### Opportunités :

- Nouvelles primitives cryptographiques
  - Meilleure compréhension des mathématiques sous-jacentes
  - Infrastructure plus robuste et agile
-

## **Glossaire des termes importants**

**AES** : Advanced Encryption Standard - chiffrement symétrique

**Average-case hard** : Difficile en moyenne, pas seulement dans le pire cas

**Ciphertext** : Texte chiffré

**CVP** : Closest Vector Problem

**D-MLWE** : Decision Module Learning With Errors

**Decoherence** : Effondrement de la superposition quantique

**Entanglement** : Intrication quantique

**IND-CCA2** : Indistinguishability under Adaptive Chosen Ciphertext Attack

**KEM** : Key Encapsulation Mechanism

**Lattice** : Réseau euclidien

**LWE** : Learning With Errors

**MLWE** : Module Learning With Errors

**NP-hard** : Non-deterministic Polynomial-time hard

**PKE** : Public Key Encryption

**Plaintext awareness** : Conscience du texte clair

**Qubit** : Quantum bit - unité quantique d'information

**Superposition** : État quantique combinant plusieurs états de base

**SVP** : Shortest Vector Problem