Implémentation Pythonique d'un Chiffrement Post-Quantique : ML-KEM comme Mécanisme d'Encapsulation de Clés

Motivation :

En découvrant la menace quantique il y a plusieurs années, j'ai été marqué par ses implications. Lorsque le thème de l'année a été annoncé, mes recherches m'ont conduit au concours post-quantique du NIST et à ses résultats, rendant l'implémentation de ML-KEM un choix évident pour mon étude.

Ancrage :

Ce travail illustre la transition vers l'ère post-quantique, nécessitant la transformation de ML-KEM d'un algorithme théorique à une implémentation fonctionnelle. Il met aussi en évidence la conversion de l'information, passant de sa forme claire à sa forme chiffrée, garantissant ainsi la sécurité face aux nouvelles menaces computationnelles.

Positionnement thématique :

INFORMATIQUE (Informatique pratique), INFORMATIQUE (Informatique théorique), MATHÉMATIQUES (Algèbre).

Mots clés :

Cryptographie post-quantique	Post-Quantum Cryptography
ML-KEM	ML-KEM
Implémentation	Implementation
Vecteurs de test	Test Vectors
Mécanisme d'encapsulation de clé	Key Encapsulation Mechanism

Bibliographie commentée :

L'objectif fondamental de la cryptographie est d'assurer la confidentialité des communications entre deux entités à travers un canal potentiellement compromis. Pour y parvenir, elle mobilise deux opérations essentielles : le chiffrement, qui convertit un message en clair en une forme chiffrée illisible pour un tiers, et le déchiffrement, qui permet de retrouver le message initial à l'aide d'une clé appropriée. L'ensemble de ce système repose sur des concepts mathématiques complexes, rendant toute tentative de décryptage sans la clé appropriée pratiquement irréalisable. [3][4]

Les systèmes de chiffrement asymétrique, en particulier, éliminent la nécessité d'un échange préalable d'une clé secrète, en s'appuyant sur une paire de clés distinctes mais algébriquement

liées : une clé publique, librement accessible et utilisée pour le chiffrement, et une clé privée, conservée exclusivement par le destinataire. La sécurité de ce schéma réside dans l'intractabilité de certains problèmes mathématiques, c'est-à-dire leur résistance à une résolution en temps polynomial. [3][4] Les mécanismes d'encapsulation de clé (KEM) exploitent cette propriété en utilisant le chiffrement asymétrique pour transmettre en toute sécurité une clé dédiée à la cryptographie symétrique, fusionnant ainsi les avantages des deux paradigmes dans une architecture cohérente et efficiente [1].

Aujourd'hui encore, les algorithmes RSA, Diffie–Hellman, ainsi que la cryptographie basée sur les courbes elliptiques (ECC), occupent une place prédominante dans le domaine de la cryptographie asymétrique. Néanmoins, l'algorithme de Shor représente une menace sérieuse pour la pérennité de ces systèmes dès lors qu'un ordinateur quantique suffisamment puissant — un ordinateur quantique dit "cryptographiquement pertinent" (CRQC) — verrait le jour. [3][5] Bien que la temporalité d'une telle percée technologique demeure incertaine, la menace des attaques qualifiées de harvest now, decrypt later — consistant à intercepter aujourd'hui des données chiffrées afin de les déchiffrer ultérieurement grâce à des capacités quantiques futures — rend indispensable la mise en œuvre de contre-mesures dès à présent [5].

C'est dans cette optique que le National Institute of Standards and Technology (NIST) a lancé, en 2016, un processus public de standardisation de la cryptographie post-quantique. Cette initiative visait à sélectionner des algorithmes de chiffrement à clé publique capables de résister aux menaces posées par l'informatique quantique. [1] Parmi les nombreuses propositions, la soumission de Kyber s'est démarquée, aboutissant à la standardisation en 2024 du ML-KEM (Module-Lattice-Based Key-Encapsulation Mechanism), aujourd'hui reconnu comme l'unique standard pour le chiffrement à usage général [5].

« An algorithm may be completely post-quantum in theory but may still be broken once implemented » [3]. Cette mise en garde prend tout son sens à la lumière des subtilités que révèle le schéma ML-KEM. En tant que mécanisme d'encapsulation de clé, il repose principalement sur le chiffrement asymétrique K-PKE [1], lequel fait l'objet de multiples optimisations, notamment en ce qui concerne la taille des textes chiffrés et les performances en temps d'exécution. Ces ajustements nécessitent le recours à divers algorithmes auxiliaires. [2][5] Une fois optimisé, le code est converti en un mécanisme d'encapsulation de clé via la transformation de Fujisaki-Okamoto, permettant ainsi d'atteindre le niveau de sécurité IND-CCA [1][2][5]. Autrement dit, même un adversaire capable d'encrypter et de décrypter des messages ne serait pas en mesure de distinguer le texte chiffré de deux messages différents [3].

Compte tenu de la longueur déjà significative du processus menant à K-PKE, l'introduction d'un test intermédiaire consistant à chiffrer et déchiffrer une image a été privilégiée. Ce test avait pour objectif de conjecturer la justesse de l'implémentation à ce stade, tout en fournissant une évaluation empirique de sa robustesse.

La problématique de la validation de la conformité de l'implémentation complète demeure cependant cruciale. Heureusement, le NIST met à disposition des vecteurs de test pertinents, fournis au format JSON, permettant de vérifier la cohérence des trois algorithmes fondamentaux — KeyGen, Encaps et Decaps — avec les sorties de référence prédéfinies. [6][7]

Problématique :

Avec la normalisation du ML-KEM, se pose la question cruciale de son implémentation sécurisée. En effet, un algorithme peut être sûr en théorie, mais son implémentation ne l'est pas forcément. Il s'agit donc de garantir une mise en œuvre correcte, conforme aux vecteurs de test fournis par le NIST.

Objectifs :

Ce projet a pour ambition de concevoir, tester et valider une implémentation cryptographique post-quantique conforme aux standards les plus récents, selon la démarche suivante :

1. Implémenter le chiffrement asymétrique K-PKE, fondement du schéma ML-KEM.

2. Programmer un test visuel permettant de conjecturer la correction fonctionnelle de l'algorithme.

3. Adapter l'implémentation en appliquant la transformation de Fujisaki-Okamoto, afin d'obtenir une version complète de ML-KEM.

4. Vérifier rigoureusement la validité finale de l'algorithme à l'aide des vecteurs de test officiels publiés par le NIST.

Références bibliographiques :

[1] National Institute of Standards and Technology (NIST). Module-Lattice-Based Key-Encapsulation Mechanism Standard. Federal Information Processing Standards Publication (fips 203), 2024, https://doi.org/10.6028/NIST.FIPS.203.

[2] Roberto Avanzi et al. CRYSTALS-Kyber (version 3.02). Submission to round 3 of the NIST post-quantum project, 2021.

[3] Jean-Philippe Aumasson. Serious Cryptography : A Practical Introduction To Modern Encryption. No Starch Press, 2024, chapitres 1, 9 et 14.

[4] Douglas Stinson. Cryptographie : Théorie et pratique. Vuibert, 2003, chapitres 1 et 4.

[5] https://cryptography101.ca/kyber-dilithium/ : Un cours complet sur ML-KEM et ML-DSA par Alfred Menezes, professeur de mathématiques à l'Université de Waterloo au Canada. Consulté depuis décembre 2024.

[6] https://github.com/mjosaarinen/py-acvp-pqc/tree/main : Une implémentation en Python de ML-KEM par Markku-Juhani O. Saarinen, professeur à l'Université de Tampere en Finlande. Consulté depuis mars 2025.

[7] https://github.com/usnistgov/ACVP-Server : Le dépôt GitHub officiel du NIST contenant les vecteurs de test. Consulté depuis mars 2025.

DOT :

Mars 2024 : Initiation à la cryptographie à travers l'étude théorique des bases fondamentales, avec un approfondissement particulier du chiffrement RSA.

Juillet 2024 : Découverte des concepts de cryptographie post-quantique, notamment à travers une première prise de connaissance de Kyber et du concours NIST PQC, menant au choix de cette thématique comme domaine général de recherche.

Août 2024 : Poursuite de l'exploration du contexte de la cryptographie post-quantique et collecte de références bibliographiques. Tentatives progressives de cerner une problématique plus précise.

Septembre 2024 : Les recherches sur les nouveaux standards de cryptographie post-quantique permettent de recentrer l'étude sur l'implémentation du schéma ML-KEM.

Octobre - Novembre 2024 : Lecture des documents fondamentaux [1] et [2] afin de comprendre la construction générale de l'algorithme. La vision d'ensemble demeure néanmoins complexe à appréhender à ce stade.

Décembre 2024 : La découverte de [5] apporte un éclairage décisif, offrant ainsi une base solide pour entamer l'implémentation pratique.

Janvier - Février 2025 : Phase d'implémentation du schéma ML-KEM en respectant scrupuleusement les spécifications définies dans [1]. Les algorithmes auxiliaires, en particulier, suscitent plusieurs interrogations techniques.

Mars 2025 : La lecture de [6] contribue à lever les zones d'ombre persistantes. Parallèlement, émergence de l'idée originale d'utiliser le chiffrement K-PKE sur une image comme test intermédiaire visuel. Vérification de la robustesse de l'implémentation par comparaison avec les vecteurs de test publiés dans [7].