JOURNAL AFRICAIN DES SCIENCES
VOL. 3, **2** (2026) 145-158

https://jafrisci.org

ISSN (online) : 3107-0949 ; ISSN (print) : 3107-0930

OPEN ACCESS

ARTICLE DE RECHERCHE

Article Info.:
Reçu : le 02/03/2026
Accepté: le 23/03/2026
Publié : le 31/03/2026

ETUDE COMPARATIVE DE LA COMPLEXITÉ ALGORITHMIQUE DES PROBLÈMES DIFFICILES DES SYSTÈMES CRYPTOGRAPHIQUES CLASSIQUES, QUANTIQUES ET POST-QUANTIQUES

Josias Josue Mukebu [1,*], Eugene Mbuyi Mukendi [2], Rostin Matendo Mabela [2], Badibanga Ntumba [2]

https://doi.org/10.70237/jafrisci.2026.v3.i2.14

Résumé

Cet article, propose une analyse comparative de la complexité des problèmes difficiles qui sont à la base de la sécurité des systèmes cryptographiques classiques, quantiques, et post-quantiques. L'objectif est d'identifier les schémas cryptographiques les plus résistants aux attaques de l'ordinateur quantique, et de dégager les perspectives pour la cryptographie post-quantique.

Mots-clés: Complexité, problèmes difficiles, algorithmes cryptographiques, attaques quantiques

Abstract

This article proposes a comparative analysis of the complexity of hard problems underlying the security of classical, quantum, and post-quantum cryptographia systems. The gaol is to identify the most quantum computer-resistant cryptographic schemes and to highlight perspectives for post-quantum cryptography.

Keywords: Complexity, hard problems, cryptographic algorithms, quantum attacks

1. Introduction

La cryptographie est un domaine de mathématiques ou de l'informatique théorique qui étudie des méthodes permettant de transférer les informations d'un point A vers un point B, de manière confidentielle. Elle a commencé par la conception de primitives cryptographiques anciennes qui avaient servi essentiellement dans le domaine militaire et diplomatique. Puis très vite, l'évolution de l'informatique a révolutionné la cryptographie en mettant en œuvre des systèmes cryptographiques modernes pour la protection de données sensibles. Dans cette période, était apparu un nouveau concept dit "clé de chiffrement", qui était un secret que seuls les deux interlocuteurs préservaient la confidentialité. Un système cryptographique où la même clé est utilisée pour le chiffrement et le déchiffrement est appelé système cryptographique ou crypto-système symétrique. Plusieurs systèmes cryptographiques symétriques étaient mise en place pour résister aux attaques de l'ordinateur classique. Certains avaient des faiblesses, mais d'autres étaient résistants et restent jusqu'à ce jour incassables face aux attaques de l'ordinateur classique. Si les conditions de Securities sont respectées, alors, l'AES par exemple est très sécurité contre les attaques de l'ordinateur classique.

Un peu plus tard, un nouveau paradigme a vu le jour. Il s'agissait de la cryptographie à clé publique. Un crypto-système est dit asymétrique, si la clé de chiffrement est différente de celle de déchiffrement. Dans ce cas, les deux interlocuteurs utilisent deux clés différentes pour communiquer. Deux problèmes difficiles sont à la base de la sécurité des systèmes cryptographiques à clés publiques. Il s'agit de problème de factorisation d'un grand nombre premier et le problème de logarithme discret. R.S.A est par exemple un crypto-système asymétrique dont la sécurité repose sur le problème de factorisation; EL GAMAL fonde sa sécurité sur le problème de logarithme discret. Si les paramètres de sécurité sont correctement utilisés, alors l'ensemble des systèmes cryptographiques dont la sécurité est basée sur le problème de factorisation d'un grand nombre et de logarithme discret, restent jusqu'à ce jour invulnérables aux attaques de l'ordinateur classique. Les problèmes difficiles de factorisation et logarithme discret sont définies sur un corps fini ou sur un groupe défini sur les courbes elliptiques. Plusieurs algorithmes cryptographiques à clé publique ont été développés pour résister aux attaques de l'ordinateur classique, et certains les sont encore jusqu'à ce jour.

Correspondance : *josuemukebu959@gmail.com* ***(J. J. Mukebu)***

Copyright : © The Author(s) Published under a Creative Commons Attribution 4.0 International Licence (CC BY 4.0)

[1] *Institut Supérieur de Techniques Appliquées de Kinshasa, , Kinshasa, République démocratique du Congo*
[2] *Mention Mathématique, informatique et Statistique, Université de Kinshasa, Kinshasa, République démocratique du Congo*

En 1970, Monsieur STEPHEN WEISNER a proposé le concept de codage conjugué et de Qubit. C'est alors que l'informatique quantique a vu le jour et c'est ainsi qu'on envisagera la mise au point de l'ordinateur quantique.

Un ordinateur quantique est un calculateur qui utilise les principes de la mécanique quantique pour le traitement des informations. Au lieu d'utiliser le bit, comme l'ordinateur classique, l'ordinateur quantique utilise le Qubit. Un Qubit est définie simplement comme la superposition de deux bits. Un Qubit est l'équivalent d'un bit quantique.

La mise au point d'un ordinateur quantique amènera des nouveaux problèmes en cryptographie: les quasi-totalités des systèmes cryptographies modernes symétriques et asymétriques deviendront vulnérables. Grâce à l'algorithme de Shor, l'ordinateur quantique serait à mesure de résoudre le problème difficile de factorisation et de l'logarithme discret en un temps polynomial. Par conséquent, tous les systèmes cryptographiques dont la sécurité repose sur ces problèmes difficiles, deviendront vulnérables face aux attaques de l'ordinateur quantique. L'algorithme de Shor est un algorithme quantique capable de résoudre des problèmes difficiles qui autre fois étaient impossible à résoudre par des méthodes classiques.

Face aux attaques de l'algorithme de Shor, certains systèmes cryptographiques classiques restent invulnérables. Les questions fondamentales qui sont à la base de cette publication sont les suivantes: Pourquoi ces crypto-systèmes résistent-ils aux attaques de l'ordinateur quantique? Quel est le fondement sécuritaire des systèmes cryptographiques post-quantiques?

Si nous arrivons à répondre aux questions posées, c'est alors que nous serons à mesure d'identifier, à partir de l'étude de la complexité des problèmes difficiles, les schémas cryptographiques les plus résistants aux attaques de l'ordinateur Quantique. Dans ce travail trois objectifs seront poursuivis:

- Le premier est celui de comprendre la construction des problèmes difficiles,
- Le second sera d'analyser la complexité des problèmes difficiles des systèmes cryptographiques classiques et post-quantiques, pour comprendre le degré de leurs résistances contre les attaques de l'ordinateur quantique,
- Le troisième consistera à comparer la complexité des problèmes difficiles pour arriver à identifier les schémas cryptographiques les plus résistants aux attaques de l'ordinateur quantique.

Organisation de l'article

Afin d'atteindre les objectifs fixés, hormis l'introduction et la conclusion générale, le travail sera organisé en trois points repartis de la manière suivante:

- Au premier point, nous aborderons la notion sur la complexité algorithmique. Les notions suivantes seront définies: complexité, algorithme déterministe, non déterministe et probabiliste. Puis nous parlerons des classes de complexité;
- Le deuxième point sera consacré à l'étude de la complexité des systèmes cryptographiques classiques symétriques et asymétriques. Un tableau sera dressé pour établir les liens entre systèmes cryptographiques classiques et leurs complexités;
- Le troisième point commencera par la définition de la cryptographie post-quantique. Ensuite, nous étudierons les variantes des algorithmes cryptographiques post-quantiques. Leurs complexités seront analysées et celles des problèmes difficiles sous-jacents.

2. Notions de base sur la complexité algorithmique [1, 2, 6, 10, 17, 19, 10]

La théorie de complexité est un domaine de mathématiques, et plus précisément de l'informatique théorique, qui étudie de manière formelle la quantité de ressources (en temps et en espace) nécessaires pour résoudre de problèmes au moyen de l'exécution d'un algorithme. Il s'agit d'étudier la difficulté intrinsèque de problèmes mathématiques posés.

Un algorithme répond à un problème précis. Il est composé d'ensembles d'étapes élémentaires nécessaires à la résolution d'un problème. Le nombre d'étapes varie en fonction des données d'entrées à traiter.

Plusieurs algorithmes peuvent répondre à un même problème. Pour comparer leurs efficacités, on utilise une mesure appelée ''Complexité''.

Un problème est dit algorithmique s'il est posé de façon mathématique, c'est-à-dire il est énoncé de manière rigoureuse en utilisant un langage mathématique. On distingue deux types de problème algorithmiques :

- Les problèmes de décision : On pose une question dont la réponse est oui ou non ;
- Les problèmes d'existence ou de recherche d'une solution.

Un problème est dit décidable s'il s'agit d'un problème de décision, c'est-à-dire, un problème dont la réponse est oui ou non et si sa réponse peut être fournie par un algorithme.
Un problème est calculable s'il s'agit d'un problème d'existence de solution et si sa solution peut être trouvée par un algorithme. En d'autres termes, s'il existe une fonction calculable qui associe à chaque entrée la réponse au problème.
Un Algorithme est dit déterministe si étant donné une entré particulière $x_i \in X$, on a la même sortie avec la même fonction f, c'est-à-dire, $f : X \to S$, $x_i \longmapsto f(x_i) = y$ où $y\ est\ unique$.

Un Algorithme est dit non déterministe si pour une même entrée $x \in X$, on a des résultats différents (qui présentent de comportements différents).

Un algorithme est dit probabiliste ou randomisé si étant donné $y_i\ (entrée\ aléatoire), i \in I$
où $Y = \{y_1, y_2, y_3, \dots, y_n\}$ est l'ensemble des variables aléatoires, il peut prendre à plusieurs moments des décisions. C'est-à-dire, la probabilité pour que y_k soit choisi parmi le $y_i, i \in I$ est égale à 1 ; $[P(y_k = y_{i,i\in I} = 1]$ et la probabilité pour que l'algorithme donne une réponse $oui\ ou\ non$ en fonction des entrées aléatoires y_i, est toujours égale à 1 ; $[P(f(y_i) = \{oui, non\} = 1]$.

Les Classes de complexité sont des ensembles de problèmes

de décision dont la résolution nécessite la même quantité de ressource (temps et en espace). Ensemble de problèmes de décision ayant la même complexité selon certains critères. Parmi les classes les plus utilisées, nous avons celles qui ont une même complexité en temps et celles ayant la même complexité en espace.

Les Classes de complexité en temps sont celles qui regroupent les problèmes algorithmiques selon le temps nécessaire pour les résoudre. Ces classes sont définies en fonction d'un algorithme déterministe ou non déterministe et de la fonction qui borne le temps en fonction de la taille des entrées. Les classes le plus courantes sont : Classe de Complexité en temps constant $O(1)$; Classe de Complexité linéaire $O(n)$; Classe déterministe Polynomiale P, $O(n^k), k \in \mathbb{N}$, Classe non déterministe polynomiale NP : elle contient les problèmes de décision pouvant être décidés par un algorithme non déterministe en un temps polynomial; Classe Co-NP (Complémentaire de NP) : c'est l'ensemble des problèmes qui admettent quand ils sont faux des contre-exemples calculables en temps polynomial, la résolution avec un ordinateur classique des problèmes dans cette classe est un défi majeur ; Classe NP Difficile : Contenue dans la classe NP, elle regroupe les problèmes qui sont aussi difficiles que les problèmes difficiles de la classe NP. Un problème d_i appartenant à la classe NP est NP difficile si $\forall\, d_{j\in I\subset N} \in NP$, d_j peut être réduit-en d_i;les problèmes qui sont dans cette classe ne peuvent généralement pas être résolus efficacement avec un ordinateur classique. Les ordinateurs peuvent cependant être utilisés pour les résoudre soit en utilisant des algorithmes approximatifs qui offre rapidement des solutions non précises, soit en utilisant des algorithmes exacts qui peuvent prendre beaucoup de temps de calcul; Classe EXPTIME (temps exponentiel), elle contient des problèmes de décision résolus en temps exponentiel par un algorithme déterministe, la résolution de problèmes qui sont dans cette classe avec l'ordinateur classique nécessite souvent des algorithmes efficaces pour des tâches complexes, car ces problèmes peuvent devenir rapidement ingérables en raison de leur complexité exponentielle; Classe NP-Complet , un problème d_i est dit NP-Complet s'il est à la fois dans la classe NP et NP-Difficile. Cela signifie que tout problème $d_{j\in I\subset N} \in NP$ peut être réduit à temps polynomial (rapidement) au problème $d_i \in NP -$Complet. $d_{j\in I\subset N}$ devient le problème le plus difficile de NP. Les problèmes NP-complets sont réputés difficiles à être résolus de manière efficace avec un ordinateur classique avec des entrées de grandes tailles. Les algorithmes exacts prennent généralement un temps exponentiel, ce qui les rend impraticable pour de nombreux problèmes concrets. Toutefois, il existe des stratégies et des algorithmes (exacts, approximatifs, heuristiques ou de réduction) qui permettent de s'attaquer à ces problèmes, même si une solution optimale n'est toujours garantie.
Réduire un problème difficile d_i appartenant à la classe NP, NP-Difficile ou NP-Complets consiste à le transformer en un temps polynomial en un problème $d_{j\in I\subset N} \in P$ facile à résoudre ou déjà résolu. Si on peut résoudre facilement le problème $d_{j\in I\subset N} \in P$ alors on peut aussi résoudre d_i.
Le lien entre ces différentes classes peut être établi de la manière suivante:

$P \subset NP,\ P \subseteq EXPTIME,\ NP \subseteq NEXPTIME, P \subsetneq EXPTIME\ et\ P \subsetneq NEXPTIME$.

Les classes de complexité en espace sont celles qui mesurent la quantité de mémoire nécessaire pour résoudre un problème de décision, en fonction de la taille des entrées. Les classes de Complexité en espace les plus courantes sont : Les classes de Complexité L : contiennent l'ensemble des problèmes pouvant être résolus en un espace logarithmique sur une machine ou un algorithme déterministe, elles ont pour complexité $Log(n)$; Les Classes de Complexité NL : regroupe l'ensemble des problèmes résolvables en un espace logarithmique sur une machine ou un algorithme non déterministe, c'est un problème équivalent à L, mais avec de capacité non déterministe; Les Classes de Complexité PSPACE : contiennent les problèmes qui peuvent être résolus en un espace polynomial sur une machine ou un algorithme déterministe ; elles ont pour complexité $\mathbf{O}(k^n)$; Les Classes de Complexité NPSPACE : contiennent les problèmes qui peuvent être résolus en un espace polynomial sur une machine ou un algorithme non déterministe ; Les Classes de Complexité EXSPACE : regroupent les problèmes solubles en un espace exponentiel sur une machine ou un algorithme déterministe; Les Classes de Complexité NEXSPACE : regroupent les problèmes solubles en un espace exponentiel sur une machine ou un algorithme non déterministe.

Pour résoudre les problèmes algorithmiques dans les classes de Complexité en Espace avec l'ordinateur classique, il faut de manière générale, l'entretien et l'optimisation de matériels, formaliser les problèmes, développer un algorithme, programmer et l'exécuter.

Les classes de complexité en espace sont reliées de la manière suivante: $L \subseteq NL,\ NL \subseteq PSPACE, PSPACE = NPSPACE\ et\ EXPSPACE = NEXPSPACE$.
Les liens entre les classes de complexité en temps et en espace sont définies par:
$P \subset NP \subset PSPACE \subset EXPSPACE\ et\ PSPACE \subseteq EXPTIME$.

Nous pouvons établir les liens entre les Classes de complexité et l'ordinateur Quantique.
L'informatique quantique, introduit une nouvelle classe dite BQP (Bounded-Error Quantum polynomial Time), qui regroupe les problèmes pouvant être résolus par l'ordinateur quantique en temps polynomial (rapidement) avec une probabilité d'erreur bornée. La classe polynomiale P est incluse dans la classe BQP. $(P \subset BQP) \Leftrightarrow$ (Tout problème de décision noté $d_{i\in I\subset \mathbb{N}}$ soluble dans P l'est également dans BQP). Un ordinateur quantique peut résoudre en temps polynomial, tout problème de décision $d_{i\in I\subset \mathbb{N}} \in P$.

L'ordinateur quantique serait capable de résoudre certains problèmes de décision classés comme NP, NP-complets, NP-difficile et EXPTIME. La question de savoir si l'ordinateur quantique peut résoudre tous les problèmes de décision dans ces classes de complexité reste encore non résolue (Problème ouvert). En informatique quantique, la conjoncture P=NP reste aussi un problème ouvert.

On notera que le niveau de recherche actuelle dans le domaine de l'informatique quantique, n'a pas encore réussi à établir complètement le lien entre les classes de complexité en espace classique et l'ordinateur quantique. Ce domaine de recherche reste encore actif.
Les algorithmes probabilistes utilisent des entrées aléatoires

pour trouver des solutions. Les ordinateurs quantiques pourraient résoudre certains problèmes de décision plus rapidement que l'ordinateur classique. Les algorithmes quantiques, qui exploitent les principes de la mécanique quantique, pourraient résoudre des problèmes inaccessibles aux ordinateurs classiques, y compris certains problèmes probabilistes.
Nous pouvons conclure le premier point en affirmant ce qui suit : Seuls les problèmes de décision classés dans P $\subset BQP$ et NP sont solubles exactement ou approximativement en un temps polynomial avec l'ordinateur classique et quantique, en fonction de la taille des entrées. Les liens des autres classes de complexité temporelles et spatiales avec l'ordinateur classique et quantique restent encore des problèmes ouverts.

3. Etude de la complexité des systèmes cryptographiques classiques symétriques et asymétriques [8, 13, 15, 18, 21, 25, 26, 28]

3.1. Systèmes Cryptographiques Symétriques

Le quadruplet d'ensembles $(\beta, \wp, K, \mathcal{F}(\beta, \wp))$ est dit système cryptographique si les conditions suivantes sont vérifiées :

- β et $\wp$ sont des sous-espaces vectoriels de $\mathbb{R}^n, n \in \mathbb{N}$ appelé ensemble de messages clairs et ensemble de messages chiffrés. $\forall \vec{x} \in \beta, \vec{x} = (x_i)_{i\in I\subset\mathbb{N}}$ et $\forall \vec{y} \in \wp, \vec{y} = (y_i)_{i\in I\subset\mathbb{N}}$ de vecteurs de $\mathbb{R}^n$. La suite $(x_i)_{i\in I\subset\mathbb{N}}$ est appelée bloc de message clair et la suite $(y_i)_{i\in I\subset\mathbb{N}}$ est appelée bloc de message chiffré.
- K sous-espace vectoriel de $\mathbb{R}$ d'algorithme déterministe qui prend en entrée un paramètre de sécurité 1^k est génère la clé sécrète$k$. $\forall k \in K$, k est la clé de chiffrement et de déchiffrement.
- $\mathcal{F}(\beta, \wp))$ sous-espace vectoriel d'applications linéaires bijectives de β dans $\wp$ ou d'algorithmes non déterministes. $\forall f_k \in \mathcal{F}(\beta, \wp)$, f_k est appelée fonction de chiffrement. C'est un algorithme qui prend en entrée la clé sécrète k et le bloc de message clair $(x_i)_{i\in I\subset\mathbb{N}}$ et donne comme résultat le chiffré $(y_i)_{i\in I\subset\mathbb{N}} = f_k(x_i)$. $\forall \mathcal{E}_k \in \mathcal{F}(\beta, \wp)$, $\mathcal{E}_k$ est appelée fonction de déchiffrement. C'est un algorithme qui prend en entrée la clé k et le chiffré $(y_i)_{i\in I\subset\mathbb{N}}$ et donne comme résultat la suite $(x_i)_{i\in I\subset\mathbb{N}} = \mathcal{E}_k((y_i)_{i\in I\subset\mathbb{N}})$.

Puisque $\mathcal{E}_k$ est l'inverse de la fonction f_k, on a : $\mathcal{E}_k((y_i)_{i\in I\subset\mathbb{N}}) = {f_k}^{-1}((y_i)_{i\in I\subset\mathbb{N}}) = {f_k}^{-1}(f_k(x_i)) = (x_i)_{i\in I\subset\mathbb{N}}$ qui le bloc de message clair qu'on retrouve.
Nous rappelons une de propriété des applications bijectives : $\mathcal{E}_k o\, f_k = id_\beta$ et $f_k o\, \mathcal{E}_k == id_\wp$.

Actuellement, la cryptographie peut être divisée en quatre grandes parties : La cryptographie ancienne, La cryptographie moderne symétrique et asymétrique, La cryptographie quantique et La cryptographie Post-quantique. Si le but de la cryptographie ancienne était de protéger les informations secrètes en les rendant inintelligible pour les non-autorisées, celui de la cryptographie moderne est de fournir un certain nombre des services de sécurité ainsi que des mécanismes de mis en œuvre de ces services, mécanismes construits au moyen des outils cryptographiques. Ces outils sont notamment les fonctions, les algorithmes cryptographiques, le générateur aléatoire, les protocoles, etc.

Les principaux services fournis par la cryptographie moderne sont : Service de confidentialité de données, Service d'intégrité de données, Service d'authentification de l'origine de données et Service de non-répudiation. Les mécanismes de mise en œuvre de ces services sont : le chiffrement, signature numérique et le protocole d'authentification.

Le tableau (1) donne la relation entre les services cryptographiques et leurs mécanismes de mise en œuvre.

Tableau 1. Lien entre services et mécanismes cryptographiques. [4]

N°	Service Cryptographique	Mécanisme de mise en Œuvre
1	Confidentialité de données	Chiffrement
2	Intégrité de données	Signature numérique
3	Authentification de données	Protocole d'authentification
4	Non répudiation de données	Signature numérique

Soit le système cryptographique $(\beta, \wp, K, \mathcal{F}(\beta, \wp))$.On sait que K est l'ensemble de clés de chiffrement et déchiffrement. On suppose qu'il existe dans K un algorithme f probabiliste qui s'exécute en temps polynomial prenant entrée le paramètre de sécurité 1^k et produit la suite de clés $(k_{i,i\in I})$ $avec\ I \subset \mathbb{N}$.
Si l'application f est un Endomorphisme c'est-à-dire, $f: K \longmapsto K$, $f(1^k) = k_1 = k, i = 1$, alors le Système cryptographique est dit Symétrique. La clé de chiffrement = la clé de déchiffrement = k. Les systèmes de chiffrement symétriques sont répartis en deux groupes: les systèmes de chiffrement symétriques anciens et les systèmes de chiffrement symétriques modernes.
Les Systèmes de chiffrement anciens incluent les méthodes telles que la substitution simple, décalage, affine, VIGINERE, masque jetable, Hill, permutation. Ces systèmes étaient souvent utilisés dans les contextes militaires ou diplomatiques. Ces méthodes, bien que rudimentaires par rapport aux normes actuelles, ont marqué l'histoire cryptographique et ont évolué au fils du temps.

3.1.1. Complexité des systèmes de Chiffrement symétriques anciens

Le tableau **(2)** donne la complexité de chiffrement, déchiffrement et des attaques des algorithmes cryptographiques symétriques anciens.

Tableau 2. Complexité des Systèmes Cryptographiques anciens et des attaques.

Algorithme de Chiffrement	Complexité Chiffrement et Déchiffrement	Complexité des Attaques	Classe de Complexité des Attaques
Décalage	$O(n)$	$O(n)$, A.F	Polynomiale
Affine	$O(n)$	$O(n^2)$, A.F ou F.B, si n est petit	Polynomiale
Substitution	$O(n)$	$O(n) + O(1)$, A.F	Polynomiale
VIGENERE	$O(n)$	$O(n)$, A.F, si k est courte ou $O(n^2)$, T.K	Polynomiale
HILL	$O(n^3)$	$O(n^3)$, Textes Clairs choisis	Polynomiale
Permutation	$O(nlog(n$	$O(n!)$, si n est petit.	EXPTIME, NP-difficile.

Dans ce tableau, les abréviations : A.F (Attaque Fréquentielle), F.B (Force Brute=Attaque par Force Brute) et T.K (Test de KASISKI) indiquent les noms d'attaques utilisées pour casser les systèmes cryptographiques.

Quel que soit la taille des entrées, définies par l'entier n, les algorithmes cryptographiques par Décalage, Affine, par Substitution, de VIGENERE et de HILL sont cassables par l'ordinateur classique via les algorithmes d'attaques indiquées. La complexité temporelle de ces attaques dépend de la taille d'entrées, de la fonction $f(n)$, de la nature des attaques, ainsi que de la performance de la machine. Seuls le chiffrement par permutation et à masque jetable peuvent résister aux attaques de l'ordinateur classique si n est très grand ou si la clé k n'est utilisée qu'une et une seule fois.

3.1.2. Systèmes de chiffrement symétriques modernes

Les systèmes de chiffrement symétriques modernes les plus utilisés sont : le D.E.S (Data Encryption Standard), le triple D.E.S (3D.E.S) et l'A.E.S (Advanced Encryption Standard). Le D.E.S est un algorithme de chiffrement en bloc. Il était autre fois une norme de chiffrement largement utilisée. Il fonctionne en partageant les données en blocs de 64 bits, utilise principalement les opérations mathématiques de substitutions, de permutation ainsi que l'arithmétique modulaire et applique une même clé sécrète pour chiffrer et déchiffrer. Cependant, il est désormais considéré comme obsolète en raison de sa clé de 56 bits, qui est jugée trop courte face aux attaques les plus avancées.

Le 3-D.E.S n'est que trois fois la composée de D.E.S. Le triple D.E.S est un chiffrement par bloc à clé symétrique qui applique trois fois l'algorithme D.E.S à chaque bloc des données. Il a été introduit comme une alternative plus forte à l'algorithme D.E.S original, qui était devenu vulnérable en raison de la longueur relativement courte de sa clé. Il est considéré comme moins sûr que l'algorithme A.E.S, bien qu'il utilise une clé de 128 bits. Sa sécurité effective est souvent estimée à 128 bits en raison des attaques de type MEET-IN-THE-MIDDLE.
TripleD.E.S=$E_{DES}{}^{k_3}(E_{DES}{}^{k_2}(E_{DES}{}^{k_1}(x)))$.
x *est message non chiffré.*

L'A.E.S est considéré comme le standard actuel et est largement employé, notamment dans les applications de sécurité les plus modernes, dans la protection des informations confidentielles, et il est connu pour sa robustesse, sa flexibilité et sa résistance à l'ordinateur quantique.

L'A.E.S chiffre un message clair de 128 bits grâce à une clé de taille 128, 192 ou 256 bits. Il est un algorithme symétrique itératif. Le nombre de tours est de 10, 12 ou 14, si la clé est de taille 128, 192 ou 256 bits respectivement. De la même façon, l'algorithme de cadencement de clé diffère en fonction de la taille de clé. Le processus de chiffrement implique plusieurs tours d'opérations : BubBytes, ShftRows, MixColumns et AddRoundeKey, qui transforment les données claires en Chiffrées. L'A.E.S est conçu pour être difficilement cassable dans un temps raisonnable, même avec des ressources de calculs importantes, ce qui le situe dans la classe de complexité des problèmes NP, même si il n'y a pas de preuve qu'il est un problème NP-complet.

Le fondement mathématique du problème difficile lié à l'algorithme A.E.S réside dans l'utilisation de transformations complexes, notamment la substitution, la permutation et l'arithmétique modulaire dans un espace mathématique spécifique appelé : le corps de Galois. Ces opérations appliquées aux blocs de données rendent extrêmement difficile le déchiffrement du texte chiffré sans avoir la clé correcte.

L'A.E.S-256 est considéré comme résistant aux attaques de l'ordinateur quantique, seulement parce qu'il n'est pas vulnérable à l'algorithme de Shor. Il n'est pas non plus considéré comme quantiquement sûr dans le sens où il n'offre pas de protection garantie contre des attaques future des ordinateurs quantiques de grande puissance.

3.1.3. Complexité des systèmes de chiffrement symétriques modernes

Le tableau (3) donne la synthèse de la complexité des algorithmes cryptographiques symétriques modernes et de leurs attaques.

Tableau 3. Complexié des Crypto-système symétriques modernes.

Algorithme de Chiffrement	Complexité Chiffrement et Déchiffrement	Complexité des Attaques	Classe de Complexité des Attaques
D.E.S	$\boldsymbol{O}$(n), où n=taille des blocs=56 bits	Attaque Force brute : (2^{45}) Cryptanalyse linéaire : (2^{43})	EXPTIME (Complexité exponentielle)
IDEA	$\boldsymbol{O}$(n), où n=taille des blocs=64 bits	Attaque Force brute : (2^{128}) ,très élevés Cryptanalyse différentielle : peu pratique.	EXPTIME (Complexité exponentielle)
BLOWFISH	$\boldsymbol{O}$(n), où n=taille des blocs=64 bits Génération de clés : $\boldsymbol{O}$(k), où k est la taille de la clé.	Attaque par Force brut : (2^k), avec k la taille de la clé, Cryptanalyse différentielle : Connue sur une version réduite,$<$ 16 tours.	EXPTIME (Complexité exponentielle)
RC, $\boldsymbol{RC4}, \boldsymbol{RC5}, \boldsymbol{RC6}$	(n), où n=taille de blocs ou de flux.	Tous, Attaque par Force brute : (2^k), où la clé k a une taille variable,$k =$ 8-128-192-256 bits.	EXPTIME (Complexité exponentielle)
3DES	(n), où n=taille de blocs.	Attaque par Force brut : $\boldsymbol{O}(2^{112})$, Meet-in-the-middle complexité $< \boldsymbol{O}(2^{112})$.	EXPTIME (Complexité exponentielle)
A.E.S	(n), où n=taille de blocs 128, 192, ou 256 bits.	Attaque par Force Brute : (2^k), où k est la taille de la clé, Attaque par canal auxiliaire.	EXPTIME (Complexité exponentielle)

Les crypto-systèmes D.E.S, IDEA, Blow Fish, RCx et 3DES sont tous vulnérables aux attaques de l'ordinateur classique. L'IDEA est plus sécurisé, mais son exécution est trop lente par rapport aux autres. Si la clé k est trop courte, l'attaque par Force brute est applicable (commune) à l'ensemble des systèmes cryptographiques repris dans le tableau (4). Les attaques différentielles, linéaires, par canaux axillaires, à clés faibles et à textes clairs connus, sont applicables efficacement ou non à l'un ou l'autre crypto-système.

Il n'y a à ce jour, aucun algorithme efficace permettant de casser le système cryptographique A.E.S. Sauf peut-être certaines attaques théoriques, qui généralement sont trop complexe pour être réelles. Si la taille de clé est respectée $(k \geq 256)$ et si l'implémentation est bien faite, alors l'AES est considéré en pratique, comme un algorithme cryptographique très sûr contre les attaques de l'ordinateur classique et même quantique.

3.2. Systèmes Cryptographiques Asymétriques

Si l'application linéaire f est définie par $f: K \mapsto K^2, f(1^k) = k_i \ pour \ i = 1,2$ alors le Système Cryptographique est dit Asymétrique. La clé de chiffrement = k_1 est différente de la clé de déchiffrement = k_2. Un système cryptographique Asymétrique est aussi appelé Système cryptographique à clé publique.

3.2.1. Problèmes Difficiles en Cryptographie à Clé publique et leurs liens avec les classes de complexité

Certains problèmes mathématiques difficiles sont utilisés pour assurer la sécurité de systèmes cryptographiques asymétriques. Ces problèmes, dont la résolution est considérée comme complexe, voire impossible avec les méthodes de calcul actuelles, sont le fondement de la sécurité de la plupart des algorithmes cryptographiques asymétriques. Parmi ces problèmes on retrouve la factorisation de grands entiers et le logarithme discret.

Problème de Factorisation : Etant donné un entier n. Le problème est de trouver deux grands nombres premiers $p \ et \ q$ tel que $n = p \times q$. Ce problème est le fondement de la sécurité de l'algorithme R.S.A.

La complexité temporelle de ce problème dépend de la taille du nombre à factoriser. Elle s'exprime en $O(n^k)$. Le problème de factorisation est dans la classe NP puisqu'il y a des algorithmes efficace pour le résoudre. On ne sait pas s'il est dans la classe P ou vérifier qu'il est dans la classe NP-difficile. Le problème reste ouvert.

L'ordinateur quantique, grâce à l'algorithme de Shor, pourrait rendre ce problème beaucoup plus facile, menaçant ainsi la sécurité de tous les systèmes cryptographiques qui ont pour fondement sécuritaire le problème de factorisation.

Le problème de R.S.A : Soit le triplet (n, e, y) $o\grave{u}$ $y \in \mathbb{Z}_n{}^*$ et $n = p \times q$.
Le problème consiste à trouver $x\ tel\ que\ y = x^e modulo\ n$.
La complexité du problème de R.S.A réside principalement dans deux aspects : la factorisation et le problème de déchiffrement qui est très couteux surtout si l'exposant secret e n'est pas connu. Lié au problème de la factorisation, le problème R.S.A est également classé NP. Il peut être cassé grâce à l'algorithme de crible quadratique, avec une clé 512 bits. Malgré le progrès de l'informatique quantique, R.S.A utilisé avec une clé $\geq$ 2048 bits, reste à l'abri des attaques classiques et quantiques.

Le problème de logarithme discret : Etant donnée le groupe $(G, +)$, un élément $\alpha \in G\ et$ un β appartenant à un sous groupe engendré par α. Le problème consiste à trouver un entier x tel que $\alpha^x = \beta$.
Si l'ordre du groupe fini (G, +) est n, alors la complexité temporelle des algorithmes est d'ordre$O(\sqrt{n})$. Il n'existe aucune preuve formelle montrant que le problème du logarithme discret appartient à la classe P, NP ou même Co-NP. Il n'existe aucune preuve non plus qui montre que ce problème n'appartient pas à ces classes. On considère simplement qu'il est entre les classes NP et Co-NP. Ce problème est considéré comme difficile à résoudre en un temps polynomial avec l'ordinateur classique. Cependant, les ordinateurs quantiques, grâce à l'algorithme de Shor, pourraient potentiellement résoudre ce problème efficacement en un temps polynomial, cela signifie qu'ils pourraient casser la sécurité des protocoles basés sur le problème de logarithme discret.
La sécurité du crypto-système R.S.A est base sur le problème de factorisation, celle d'ELGAMAL est base sur le problème de logarithme discret.

Le tableau (4), donne la synthèse de la complexité de chiffrement/déchiffrement et attaques des algorithmes cryptographiques dont la sécurité est basée sur le problème de factorisation et de logarithme discret.

Tableau 4. Complexité des systèmes cryptographiques Asymétriques.

Algorithme de Chiffrement	Complexité Chiffrement et Déchiffrement	Classe de Complexité des Attaques
R.S.A	$\boldsymbol{O}(log^{3n})$ Génération clé (log^{4n}) : tests de primalité	C: EXPTIME Q: P (Polynomiale)
ELGAMAL	(log^{3p}),où p est le modulo d'ELGAMAL Génération de clés : (log^{4p}).	C: EXPTIME Entre P et EXPTIME. Q: P(Polynomiale)

Le crypto-système RSA est vulnérable aux attaques de l'ordinateur classique si la clé $k \leq 2048$ bits. En plus des attaques citées dans le tableau, il faudra faire attention aux attaques ciblant le secret $d, si\ d\ est\ petit$ et à l'implémentation de RSA (Attaque par canaux auxiliaires). Pour une meilleur sécurité, il est recommandé de prendre aléatoirement $p\ et\ q$ (facteurs premiers) et $k \geq 2048$ bits.

RSA est vulnérable aux attaques de l'ordinateur quantique, car l'algorithme de Shor peut résoudre le problème difficile de factorisation en un temps polynomial.
ELGAMAL-2048 et ELGAMAL-3072 sont vulnérables aux attaques de l'ordinateur quantique, car l'algorithme de Shor peut résoudre le problème difficile de logarithme discret (DLP) en un temps polynomial. Si p est grand et les autres paramètres sont bien choisis, alors les deux versions d'ELGAMAL citées ci-dessus sont résistantes aux attaques de l'ordinateur classique, mais leur exécution est très lente.

La signature avec RSA et ELGAMAL s'exécute en un temps polynomial avec l'ordinateur classique, en $\boldsymbol{O}(log^{3p}), o\grave{u}\ p\ est\ l'entier\ modulo$. Leur vulnérabilité à l'ordinateur classique est observée lorsque la taille de la clé $k \leq 2048\ bits$ et lorsque les autres paramètres sont mal choisis. Les attaques les plus courantes qu'on leurs applique sont notamment celles qui dépendent de leurs implémentations, et celles qui sont conçues pour factoriser un grand entier premier et résoudre le problème de logarithme discret. Même avec la taille de clé$k \geq 3072\ bits$, les protocoles de signature cryptographique dont la sécurité est basée sur les problèmes difficiles de factorisation et de logarithme discret sont vulnérables aux attaques de l'ordinateur quantique.

3.2.2. Problème Difficile des Courbes Elliptiques [9, 10,]

1. Le Logarithme Discret sur les courbes Elliptiques (ECDLP)

On appelle ordre d'un élément g d'un groupe, le plus petit entier n tel que $g^n = e$, où e est l'élément neutre du groupe. Dans le cas du groupe formé par les points d'une courbe elliptique, c'est le plus petit entier n tel que $[n]P = \infty$.
Notons $\langle P \rangle$ l'ensemble $\{[i]P; 0 \leq i \leq n-1\}$ où n est l'ordre de P.
L'ensemble $\langle P \rangle$ forme un sous-groupe cyclique de $E(\mathbb{F}_q)$ engendré par P.
Le problème de logarithme discret sur les courbes elliptiques est l'un de problème difficile des courbes elliptiques. Il est énoncé de la manière suivante : Etant donnés une courbe elliptique E définie sur un corps fini $\mathbb{F}_q$, un point $P \in E(\mathbb{F}_q)$ d'ordre n et un point $Q \in \langle P \rangle$, trouver l'entier $k \in [0, n-1]$ tel que $Q = [k]P$.
La complexité de l'algorithme est de $O(\sqrt{n})$, où n est l'ordre du groupe de la courbe elliptique.
La difficulté de résoudre le problème de logarithme discret sur les courbes elliptiques, constitue le fondement de la sécurité des

systèmes cryptographiques basés sur les courbes elliptiques (ECC).

2. Relation entre le Problème Difficile du Logarithme Discret sur les Courbes Elliptiques (ECDLP) et l'Ordinateur Quantique [12, 23]

Les ordinateurs quantiques s'ils voyaient le jour, représenteraient une menace potentielle pour les systèmes cryptographiques basés sur les courbes elliptiques, notamment le problème difficile du logarithme discret.
L'algorithme de Shor en cryptographie quantique, conçu pour l'ordinateur quantique, permettrait de résoudre le problème difficile du logarithme discret sur les courbes elliptiques en un temps considérablement réduit (temps polynomial), compromettant la sécurité des crypto-systèmes dont la sécurité est basée le problème difficile de logarithme discret sur les courbes elliptiques. Les ordinateurs quantiques exploitent les propriétés de la physique quantique pour effectuer des calculs massives en parallèles. L'algorithme de Shor utilise ces capacités pour factoriser des grands entiers et résoudre le problème de logarithme discret classique, y compris le problème de logarithme discret sur les courbes elliptiques (ECDLP).
Plusieurs systèmes cryptographiques basés sur l'ECDLP ont été proposés. Nous pouvons citer les protocoles de signature: ECDSA; les protocoles de chiffrement de données: ECEG, ECIES, PSEC ; et les protocoles d'échange de clés : ECMQV, ECDH, etc.
Ces systèmes cryptographiques sont vulnérables aux attaques de l'ordinateur quantique, car nous savons que leur sécurité repose sur le problème difficile de logarithme discret sur les courbes elliptiques (ECDLP). Pour remédier à ces problèmes, nous allons recourir aux algorithmes cryptographiques post-quantiques dont la sécurité reposera sur d'autres problèmes mathématiques difficiles.

3. Analyse de la Complexité des systèmes cryptographiques quantiques, post-quantiques et des problèmes difficiles sous-jacents [5, 7, 14, 16, 22, 24]

La cryptographie post-quantique est une branche de la cryptographie où on conçoit des systèmes cryptographiques résistants aux attaques de l'ordinateur quantique.

1. Les Variantes des Algorithmes Cryptographiques Post-quantiques

Les variantes des algorithmes cryptographiques post-quantiques sont des outils mathématiques permettant de concevoir de systèmes cryptographiques résistants aux attaques de l'ordinateur quantique.

Actuellement, la recherche se concentre principalement sur six variantes d'algorithmes cryptographiques post-quantiques. Il s'agit d'AES, un algorithme de chiffrement à clés secrète connu pour être résistants aux attaques quantique ; de Réseaux euclidiens très utilisés pour construire des schémas d'échange de clé, de signature numérique et de chiffrement ; de la cryptographie à base de codes qui fait référence à la notion de codes correcteurs d'erreurs ; de la cryptographie multi variée basée sur le problème de polynômes quadratiques multi variés ; de la cryptographie basée sur le hachage qui repose sur les propriétés des fonctions de hachage cryptographique ; et de la Cryptographie à bases d'Isogénies basée sur les Isogénies qui dépendent des courbes elliptiques.
Dans cette partie, nous nous proposons d'étudier la complexité de trois variantes cryptographiques post-quantiques : L'AES, la cryptographie à base de réseaux euclidiens, et la Cryptographie à base de codes correcteurs d'erreurs. Les autres feront l'objet des études approfondies en thèse. **Les problèmes difficiles** qui sont à la base de la résistance de ces algorithmes face à l'ordinateur quantique seront décrits et leurs complexités analysées.

a. L'AES Résistant à l'ordinateur Quantique

L'A.E.S est considéré comme le standard actuel et est largement employé, notamment dans les applications de sécurité les plus modernes, dans la protection des informations confidentielles, et il est connu pour sa robustesse, sa flexibilité et sa résistance à l'ordinateur quantique.
Le problème difficile de L'A.E.S repose sur la difficulté de retrouver la clé k. Pour une recherche exhaustive, la complexité est exponentielle. L'AES est résistant aux attaques par force brute et par canal axillaire. Il faudra aussi faire attention à la gestion de clé et à la mise en œuvre correcte de l'algorithme. L'A.E.S est conçu pour être difficilement cassable dans un temps raisonnable, même avec des ressources de calculs importantes, ce qui le situe dans la classe de complexité des problèmes NP.

Le fondement mathématique du problème difficile lié à l'algorithme A.E.S réside dans l'utilisation de transformations complexes, notamment la substitution, la permutation et l'arithmétique modulaire dans un espace mathématique spécifique appelé : le corps de Galois. Ces opérations appliquées aux blocs de données rendent extrêmement difficile le déchiffrement du texte chiffré sans avoir la clé correcte.

L'A.E.S-256 est considéré comme résistant aux attaques de l'ordinateur quantique, seulement parce qu'il n'est pas vulnérable à l'algorithme de Shor. Il n'est pas non plus considéré comme quantiquement sûr dans le sens où il n'offre pas de protection garantie contre des attaques future des ordinateurs quantiques de grande puissance.
Bien qu'il existe des menaces potentielles, l'A.E.S-256 reste un algorithme de chiffrement robuste, particulièrement dans sa version 256 bits. Il est peu probable qu'il soit compromis par des ordinateurs quantiques dans avenir proche.

b. Systèmes Cryptographique à Base de Codes Correcteurs d'Erreurs

Lorsqu'Alice et Bob communiquent, il est préférable que l'information transmise par Bob coïncide avec celle reçue par Alice. Cependant, le canal utilisé pour cette communication peut être un canal-bruité, et chaque bit d'information a une probabilité p de se retrouver altéré. L'idée principale des codes correcteurs d'erreurs est de restreindre l'ensemble des mots que Bob peut utiliser de sorte que si Alice reçoit un mot hors de cet ensemble, mais relativement proche d'un mot valide, alors elle pourra tout de même comprendre quel mot Bob a voulu émettre. Cette restriction se fait en rajoutant de la redondance, d'une manière bien spécifique.

1. Problèmes Difficiles sur les codes

Distribution SD (Syndrome Decoding)

Pour $n, k\ et\ w$ des entiers positifs, la distribution $SD\ (n,k,w)$ consiste à choisir $H \overset{\$}{\leftarrow} \mathbb{F}^{(n-k)\times n}$ et $x \overset{\$}{\leftarrow} \mathbb{F}^n$ tel que $w(x) = w, et$ retourner (H, Hx^T).

Pour un ensemble S, $x \overset{\$}{\leftarrow} S$ signifie que x est échantillonné uniformément au hasard parmi les éléments de S.

Le problème SD est NP-complet et pour des poids relativement petits, la complexité d'attaques classique et quantique est exponentielle, soit $O(2^{n-k})$. IL reste un problème difficile pour un ordinateur Quantique et est à la base de la sécurité du crypto système de McElice.

Problème (SEARCH) S-QCSD

Soient $n, k, w\ et\ s$ des entiers positifs, H une matrice de parité aléatoire d'un code QCC et $y \overset{\$}{\leftarrow} \mathbb{F}^{(n-k)}$. Le problème SEARCH S-QCSD $(n\ , k, w)$ consiste à trouver $x \in \mathbb{F}^n$ de poids $w\ (x) \leq\ \delta$ tel que $y = x\ H^T$.

Il existe une réduction du Problème (SEARCH) S-QCSD au Problème SD. Il est considéré comme un problème difficile pour l'ordinateur classique et Quantique. Il a complexité d'attaques exponentielles, soit $O(2^{n-k})$ et est dans la classe NP-complet.

Problème Décisionnel S-DQCSD

Soient $n, k, w\ et\ s$ des entiers positifs, H une matrice de parité aléatoire d'un code QC et $y \overset{\$}{\leftarrow} \mathbb{F}^{(n-k)}$. Le problème $S - DQCSD(n, k, w)$ consiste à décider avec un avantage non négligeable si (H, y^T) provient de la distribution $s - DQCSD(n, k, w)$ ou de la distribution uniforme sur $\mathbb{F}^{(n-k)\times n} \times \mathbb{F}^{(n-k)}$.

Il est de complexité exponentielle, soit $O(2^{n-k})$, est dans la classe NP-complet et est résistant aux attaques de l'ordinateur Quantique.

En résumé, les problèmes difficiles Distribution SD, le problème (SEARCH) S-QCSD et Problème Décisionnel S-DQCSD sont de complexité exponentielle et sont classes NP-Complet. Les systèmes cryptographiques dont la sécurité repose sur ces problèmes difficiles, ne sont pas vulnérables aux attaques de l'ordinateur quantique.

2. Principaux Schémas de Chiffrement

Plusieurs Schémas dont la sécurité repose sur les problèmes difficiles de codes correcteurs d'erreurs ont été développés. Les crypto-systèmes de McEliece, MDPC pour la métrique de Hamming, ainsi que le crypto système LRPC pour la métrique rang sont les plus utilisés.

On notera que la sécurité du système cryptographique McEliece repose sur le problème de distributions SD.

LRPC et MDPC sont des variantes de McEliece. Leur sécurité repose également sur le problème de distribution SD. LRPC est un algorithme cryptographique utilisant le code LRPC pour le chiffrement quantique. Il est nécessaire de faire des analyses continues en fonction des paramètres utilisés pour assurer sa résistance contre les attaques de l'ordinateur Quantique. L'LRPC est en surveillance par la communauté cryptographique.

Avec l'évolution de recherche sur la mise au point de l'ordinateur Quantique, des analyses sont régulièrement faites pour s'assurer que les crypto-systèmes MDPC et LRPC continuerons à résister contre les attaques de l'ordinateur quantique. En attendant, nous pouvons affirmer que McEliece et ses variantes, sont incassables et constituent donc des systèmes cryptographiques post-quantiques fiables.

c. La Cryptographie Basée Sur les Réseaux Euclidiens

Un Réseau Euclidien est un sous-groupe discret de l'espace vectoriel $\mathbb{R}^p$ qui est stable par l'addition et possède un élément neutre pour cette opération, et il y a toujours « un blanc » entre deux points d'un réseaux aussi proches soient- ils.

Soient la matrice notée $A = (a_{ij})_{\substack{1\leq i\leq n \\ 1\leq j\leq p}}$ où $n, p \in \mathbb{N}$. On note

$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \dots a_{1p} \\ a_{21} & a_{22} & a_{23} \dots . a_{2p} \\ a_{31} & a_{32} & a_{33} \dots a_{3p} \\ \vdots & \vdots & \vdots \ddots \vdots \\ a_{n1} & a_{n2} & a_{n3 \dots\dots a_{np}} \end{pmatrix}$ Matrice à

$n\ lignes\ et\ p\ colonnes$ et $\mathcal{M}_{n,p}(\mathbb{R})$ l'ensemble des matrices à n lignes et p colonnes à coefficients réels. Clairement la matrice $A \in \mathcal{M}_{n,p}(\mathbb{R})$. Si on pose :

$b_1 = (a_{11}, a_{21}, a_{31}, \dots, a_{n1})$, $b_2 = (a_{12}, a_{22}, a_{32}, \dots, a_{n2})$,

$b_3 = (a_{13}, a_{23}, a_{33}, \dots, a_{n3})$, ...,

$b_p = (a_{1p}, a_{2p}, a_{3p}, \dots, a_{np})$,

alors la famille notée $M = (b_1, b_2, b_3, \dots, b_p) \in \mathbb{R}^{p\times n}$ et est dit à $n\ vecteurs\ et\ p\ coordonnées$.

Le réseau $\Gamma(M)$ engendré par la famille M est défini par une combinaison linéaire à coefficients entiers, notée : $\Gamma(b_i,\ 1 \leq i \leq p) = \{\sum_{i=1}^{n} \lambda_i b_i :\ \lambda_i \in \mathbb{Z}\}$. La famille M forme une base du réseau, p est sa dimension et n son rang. Lorsque $n = p\ (matrice\ carrée)$, alors ont dit le réseau est de rang plein.

Un des problèmes les plus connu en réseaux euclidien est celui, de la recherche du ou des plus court(s) vecteurs. Dans un réseau $\Gamma(M)$, tout vecteur $y \neq 0$ à longueur strictement positive, $\|y\| = L(x) > 0$. La question qu'on se pose souvent est de savoir si cette longueur est relativement petite par rapport aux autres vecteurs.

Soit $\Gamma(M)$ un réseau de base $M \in \mathbb{R}^{p\times n}$. Pour $1 \leq i \leq n$, on définit la $ième\ minimun\ successif$, noté par : $\lambda_i = inf\{r: dim(Vect\ (\Gamma(M) \cap B_p(0,r)) \geq i\}$ ou $B_p(0,r) = \{x \in \mathbb{R}^p : \|x\| \leq r\}$ désigne la boule fermée en dimension p, centrée en $O \in \Gamma(M)\ et\ de\ rayon\ r$.

Pour $i = 1$, $\lambda_1(\Gamma(M))$ n'est rien d'autre que la longueur du plus petit vecteur du réseau $\Gamma(M)$. Comme nous le verrons dans la suite, cette valeur n'est pas évidente. Le théorème suivant donne la borne inférieure :

$\lambda_1(\Gamma(M)) \geq min_{1\leq i\leq n}\|\tilde{b_i}\|$. Où le vecteur $\tilde{b}_i$ correspond à l'orthogonalisation de Gram-Schmidt des vecteur $\tilde{b}_i$ dans l'ordre lexicographie.

En effet, Soit $x \in \mathbb{Z}^n$, un vecteur non nul et quelconque, $j \in \{1,2,3, \dots, n\}$ le plus grand entier tel que $X_j \neq O$, alors $|\langle Mx, \tilde{b}_j\rangle| = |\langle \sum_{i=1}^{n} x_i b_i\ , \tilde{b}_j\rangle| = |x_j|\langle \tilde{b}_j, \tilde{b}_j\rangle = |x_j|\|\tilde{b}_j\|^2$;

d'après Cauchy-Schwartz, $|\langle Mx, \tilde{b}_j \rangle| \leq \|Mx\| \|\tilde{b}_j\|$. Lorsqu'on combine les deux inégalités, on obtient :
$\|Mx\| \geq |x_j| \|\tilde{b}_j\| \geq \|\tilde{b}_j\| \geq \min_{1 \leq J \leq n} \|\tilde{b}_j\|$. De plus, les minima successifs sont atteints dans les sens où $pour\ tout\ i \{1,2,3, \dots, n\}$, il existe un vecteur $v_i \in \Gamma(M)$ tel que $\|V_i\|=\lambda_i$.

2. Problèmes Difficiles de la Cryptographie à base des réseaux euclidiens

En cryptographie sur les réseaux euclidiens, les problèmes difficiles se présentent sous deux formes : Problèmes difficiles dans le cas-moyen et le problème difficile dans le pire-cas.
Les travaux d'ATTAI ont établi une connexion entre ces deux problèmes. Dire par exemple qu'un problème mathématique est NP-difficile, signifie qu'on fait référence aux instances les plus difficiles de ce problème. Générer une telle instance en cryptographie, peut être compliqué, ou calculatoirement impossible ; mais il existe une autre approche qui consiste à générer la clé cryptographique en réduisant la solution d'un problème dans le pire-cas à celle d'un problème dans le cas moyen.

La cryptographe à base des réseaux euclidiens, repose notamment sur la conjecture de Micciancio et Regev qui atteste qu'il n'existe aucun algorithme s'exécutant en temps polynomial, qui puisse approximer jusqu'à un facteur polynomial les problèmes basés sur les réseaux.

Dans ce paragraphe, nous nous proposons de présenter, les problèmes difficiles dans le pire-cas, ainsi que quelques problèmes difficiles dans le cas-moyen auxquels cas ces premiers peuvent se réduire.

1. Problème dans le Pire-Cas
Shortes Vector Problem (SVP)

Le problème du plus court vecteur dans un réseau euclidien est probablement le plus connu et le plus étudier de tous les problèmes sur les réseaux euclidiens.
La sécurité des crypto systèmes basée sur les problèmes difficiles de réseaux euclidiens possédant une réduction s'y ramène de manière systématique.
Etant donnée une base B d'un réseau $\Gamma(M)$ et un facteur d'approximation $\gamma \in \mathbb{R}, \gamma \geq 1$, le problème Shearch-SVP consiste à trouver un vecteur non-nul $v \in \Gamma(M)$, tel que $\|v\| \leq \gamma \lambda_1(\Gamma(M))$..

Shortes Indépendant Vector Problem (SIVP)

Le problème SIVP est équivalent au problème SVP, mains ici au lieu de trouver l'unique vecteur plus court, non-nul du réseau $\Gamma(M)$, on s'emploiera à trouver plusieurs vecteurs courts, et linéairement indépendants. La difficulté augmente, car chercher plusieurs vecteurs plus courts est plus difficile que d'en chercher un seul.

Le problème **SIVP** est énoncé de la manière suivante : Soient une base M du réseau $\Gamma(M)$ et un facteur d'approximation $\gamma \in \mathbb{R}$. Le problème SIVP consiste à trouver n vecteurs $v_i (1 \leq i \leq n)$ non-nuls et libres tels que $v_i \in \Gamma(M)\ et$ leur module, $\|v\|_i \leq \gamma \lambda_1(\Gamma(M))$; la longueur des vecteurs v_i est plus courtes que tous les autres vecteurs du réseau.

Le Problème du plus proche vecteur dans le réseau euclidien. Il est noté CVP qui signifie en anglais (Closest Vector Problèm)

E Etant données une base du réseau $\Gamma(M)$, le sous-espace vectoriel $Vect\ (M)$ engendré par les composantes de la base M et un vecteur $t \in Vect\ (M)$. Le problème CVP consiste à trouver un vecteur $v \in \Gamma(M)$ tel que la distance entre $t\ et\ v$ notée $\|v - t\| \leq \gamma\ dist(t, \Gamma(M))$, est plus courte que la distance entre t et tous les autres vecteurs du réseau $\Gamma(M)$. Où γ un facteur d'approximation.
Il à noter que tous les problèmes difficiles énoncés ci-dessus, sont strictement indépendants du facteur d'approximation γ. Leurs versions exactes sont obtenues en prenant $\gamma = 1$. Le facteur d'approximation γ un est indicateur de la plus petite valeur (distance ou longueur) parmi les autres valeurs dans le réseau $\Gamma(M)$.
Si le facteur d'approximation γ est un majorant, c'est-à-dire pour tout $\mu \in \Gamma(M), \gamma \geq \mu$ (le plus grand parmi les plus petits), alors le problème CVP devient ADD (Absolue Distance Deconding).
Si le facteur d'approximation $\gamma \in \left[1, \frac{\mu}{2}\right]$, alors le problème CVP devient BDD (Bounded Distance Decoding).
Dans BDD, si la solution existe, alors elle unique. Alors que dans ADD, l'existence de la solution est assurée mais pas l'unicité.
Il a été démontré qu'approcher SVP n'était pas plus dur que d'approcher CVP. La complexité du problème CVP est plus forte que celle du problème SVP.
Complexité du problème SVP en fonction de l'ordre de grandeur de $\boldsymbol{\gamma}$.
Les problèmes SVP, SIVP et CVP sont classés NP-difficiles. Nous pouvons affirmer que tous systèmes cryptographiques dont la sécurité est basée sur ces problèmes sont quantiquement sûrs.

2. Problèmes Difficiles dans le Cas-moyen
Problème SIS (Small Integer Solution)
Soient la matrice $A \in \mathbb{Z}_q^{n \times m}$ **et** les $k -$ $vecteurs\ e_1, e_1, e_3, \dots, e_k \in \Gamma_q^{\perp}(M)$ vérifiant $\|e_i\| \leq \gamma\ avec\ \gamma \in \mathbb{R}$.

Le problème noté $k - SIS_{\gamma,q}$(dépendant des paramètres γ, q) consiste à trouver le vecteur $v\ de\ \mathbb{Z}^m$ tel que :

- $\|v\| \leq \gamma$,
- $Mv = 0\ modulo\ q$, c'est-à-dire que $v \in \Gamma_q^{\perp}(M)$ et
- $v \notin Q - vect\ (e_1, e_1, e_3, \dots, e_k)$.

Si $k = 0$, alors le problème le problème $k - SIS_{\gamma,q}$ devient le problème SIS.
Il a été démontré que si un attaquant possédant un ordinateur quantique peut résoudre le problème $k - SIS_{\gamma,q}$ en dimension m, alors il peut résoudre le problème SIS en dimension $m - k$.
Le problème SIS peut être vu comme une instance approchée du problème SVP. Sa complexité est équivalente à celle du problème difficile SVP. Il est dans la classe de complexité NP-dur.

Problème LWE (Learning Errors)

Si nous fixons un paramètre de sécurité λ et si on pose $n = n(\lambda)$ une dimension entière, alors le polynôme $P(x) = x^{2^k} + 1$ est dit Cyclo-tonique, pour $k \in \mathbb{N}$. Posons $R = \mathbb{Z}[x]/f(x)$, $R_q = \frac{R}{qR}$ et $x\ qui\ dépend\ de\ \lambda,\ x = y(x)$ une distribution $R\ d'écart\ type\ \sigma$.
Le problème GLW consiste à distinguer deux distributions $(a_i, \langle a_i, s\rangle + e)\ et\ (a_i, r) \in \mathbb{R}_q^{n+1}$, pour $a_i, s \in \mathbb{R}_q^n\ et\ b \in \mathbb{R}_q$ distribué uniformément, sans perte de sécurité $s \leftarrow x^n$ et $e \leftarrow x$.
On notera que si $d = 2^k = 1$, c'est –à-dire que $P(x) = x + 1$ alors le problème $LWE \equiv GLWE$ et si $n = 1$, le problème LWE devient $Ring - LWE$.
REGEV a montré récemment que si un attaquant qui possède un calculateur quantique peut résoudre un problème LWE dans le cas-moyen, alors il peut résoudre le problème SIVP et SVP dans le pire-cas. Dans ce cas, le problème SIVP et SVP dans le pire-cas, peuvent être réduit au problème LWE dans les cas moyens.

LYBASHEVSKY, PEIKAT et REGEV ont montré également, qu'il est possible de réduire un problème SVP dans le pire-cas en un problème Ring-LWE dans le cas-moyen.
La version binaire, c'est-à-dire $(q = 2)$ du problème LWE est appelée LPN (Learning Party With Noise).
La complexité du problème LWE est exponentielle, elle dépend de la dimension du réseau. Elle augmente très rapidement en fonction de la dimension du réseau et le niveau de bruit. LWE appartient à la classe NP-dur.
Les relations graphiques entre les différents problèmes difficiles dans un réseau euclidiens se présentent de la manière suivante :

4. Systèmes Cryptographiques à bases de réseaux euclidiens

Nous nous proposons d'étudier dans cette partie deux crypto-systèmes basés sur les réseaux euclidiens :

Le Crypto système NTRU
Algorithme (Principe général).

Si Alice veut envoyer un message m à Bob, elle doit tout d'abord générer une clé Publique e_b et deux clés privées $e_{k_1}\ et\ e_{k_2}$. Si Bob veut chiffrer le message m, il calcule $c = f_{e_b}(m)$ et l'envoie à Alice. A la réception, Alice calcule $g_{(e_{k_1}e_{k_2})}(c) = g_{(e_{k_1}e_{k_2})}\left(f_{e_b}(m)\right)$ pour retrouver le message m.

Procédé de génération des clés

Pour générer les clés, Alice doit d'abord choisir trois entiers $n, p\ et\ q$, et ensuite exécuter l'algorithme suivant :

- Elle choisit aléatoirement deux polynômes $f\ et\ g\ dans\ E_q\ et\ E_p$ respectivement.
- Elle calcule $f^{-1}\ modulo\ q = e_{k_1}\ et\ f^{-1}\ modulo\ p = e_{k_2}$.
- La clé publique $e_b = p \times g \times e_{k_1}\ modulo\ q$ (On obtient e_b par le produit de convolution de polynômes).
- Elle obtient finalement la clé publique e_b et les deux clés privées $e_{k_1}\ et\ e_{k_2}$.

Procédé de Chiffrement

Pour Chiffrer le message m, Bob doit d'abord transformer le message m en un polynôme de M_n et appliquer l'algorithme suivant :

- Il choisit aléatoirement un polynôme $r\ de\ M_r$,
- Il calcule $c = r \times e_b + m\ modulo\ q$,
- Il obtient le chiffré c et l'envoie à Alice.

Procédé de Déchiffrement

A la réception, avec les clés secrètes $e_{k_1}\ et\ e_{k_2}$, Alice calcule :

- $\boldsymbol{a} = e_{k_1} \times c\ modulo\ q$,
- Il faut recentrer les coefficients dans l'intervalle de longueur q),
- $m = e_{k_2} \times c\ modulo\ p$ et obtient le message clair m.

Vérification : En effet : $\boldsymbol{a} = e_{k_1} \times c\ modulo\ q = e_{k_1} \times \big(r \times (p \times g \times e_{k_1}\ modulo\ q\ ) + m\ modulo\ q\ \big) modulo\ q = \big(e_{k_1} \times \boldsymbol{r}\big) \times \big(\boldsymbol{p} \times \boldsymbol{g} \times e_{k_1}\ modulo\ q\ \big) + m\ moduloq = \Big(e_{k_1} \times \big(\boldsymbol{r} \times \boldsymbol{p} \times \boldsymbol{g} \times e_{k_1}\ modulo\ q\big)\Big) + m\ modulo\ q = e_{k_2} \times c\ modulo\ p = m.$ ■

Complexité **:** La sécurité du crypto-système NTRU est basée sur le problème difficile SVP.
Sa complexité est exponentielle et dépend largement de la taille de la dimension du réseau. NTRU est dans la classe de complexité NP-dur. Dans cette classe, aucun ordinateur, même pas quantique, ne résoudre efficacement les problèmes difficiles appartenant à cette classe.

Le Crypto-système de REGEV

La sécurité du système cryptographique de REGEV repose sur le problème difficile D'LWE. Ce schéma cryptographique a été proposé par OLDED REGEV en 2005.
Le paramètre de sécurité de ce schéma est la dimension n du réseau. Les autres paramètres sont des entiers notés : $m, q\ et\ a$.
Toutes les opérations dans ce schéma se font modulo q.

- **Génération des clés**
  On considère les paramètres $n, m\ et\ q$ où $q\ est\ premier\ et\ m \geq 4(n+1)\ log_2 p$ et $a \in \left]0, \frac{1}{4m}\right[\subset \mathbb{R}$. La clé secrète est donnée $k_1 \in \Gamma\left(\mathbb{Z}_q^m\right)$. La clé publique est le couple $k_2 = (A, b) = (A, As + e)$, où $A \in \Gamma\left(\mathbb{Z}_q^{m \times n}\right)\ et\ e \in D_{\mathbb{Z}^m, aq}$.
- **Procédé du Chiffrement**
  Soit le message m à $chiffrer$. On suppose que $m \in \{0,1\}$. $r \in \Gamma(\{0,1\}^n)$ est échantillonné pour renvoyer le chiffré $\left(r^T A,\ r^T b + \frac{q}{2} m\right) \in \mathbb{Z}_p^n \times \mathbb{Z}_p$.
- **Procédé de Déchiffrement**
  Soient un chiffré $(v^T,\ V) \in \mathbb{Z}_p^n \times \mathbb{Z}_p$. Calculer $v - V^T s$. Le message clair est nul si le résultat tend vers zéro que $\frac{q}{2}$, et 1 si non.

Complexité. Le schéma du chiffrement de REGEV est dans la classe NP-dur. Sa complexité est exponentielle, grossie très vite en fonction de la dimension n. Même complexité avec le problème LWE.

5. Résultats:

Les résultats de notre analyse sont présentés dans le tableau ci-dessous de la manière suivante :

Algorithme	Problème Difficile	Complexité des Attaques Classiques	Complexité des Attaques Quantiques
AES	Recherche de Collision	$O(2^n)$	$O(2^{\frac{n}{2}})$ (Attaque Grover)
RSA	Factorisation	$O\left((e^{\,n})^{\frac{1}{3}}\right)$	$O\ (n^3)$ (Attaque de Shor)
DIFFIE-HELMAN et ELGAMAL	Logarithme Discret	$O\left((e^{\,n})^{\frac{1}{3}}\right)$	$O\ (n^3)$ (Attaque de Shor)
ECDSA, ECEG, ECIES, PSEC et ECMQV	ECDLP (Logarithme discret sur la courbe Elliptique)	$O\left((2)^{\frac{n}{2}}\right)$	$O\ (n^3)$ (Attaque de Shor).
BB84	Problème de distinction entre les états quantiques entremêlés et séparés.	$O((2)^n)$	$O\ (n)$ (Attaque de Shor).
EPR	Problème de non-localité quantique	$O\left((2)^{\frac{n}{2}}\right)$	$O\ (n)$ (Attaque de Shor).
MCELIECE	Problème de Syndrome Decoding (SD)	$O\left((2)^{\frac{n}{2}}\right)$	$O\left((2)^{\frac{n}{4}}\right)$ (Attaque de Grover).
MDCPC	Difficulté de décoder un code linéaire	$O\left((2)^{\frac{n}{2}}\right)$	$O\left((2)^{\frac{n}{2}}\right)$ (Attaque de Grover).
LRPC	Problème de Décodage de LRPC	$O\left((2)^{\frac{n}{2}}\right)$	$O\left((2)^{\frac{n}{2}}\right)$ (Attaque de Grover).
NTRU	Problème SVP	$O\left((2)^{\frac{n}{2}}\right)$	$O\left((2)^{\frac{n}{2}}\right)$ (Attaque de Grover).
REGEV	Problème d'ELWE	$O((2)^n)$	$O\ (n^3)$ (Attaque de Shor).

6. Discussion

L'AES est résistant aux attaques de l'ordinateur classique et quantique. Il est à noter qu'on dit qu'il résiste aux attaques quantiques seulement parce qu'il résiste aux attaques de l'algorithme de Shor ; il n'est pas quantiquement sûr car il n'existe aucune preuve démontrant qu'il résistera aux attaques futures de l'ordinateur quantique. Lors de la sélection par le NIST (National Institute of Standards and Technology) des algorithmes cryptographiques post-quantiques, l'AES a joué un rôle très important dans la définition des exigences de sécurité ; le niveau de sécurité des crypto-systèmes post-quantiques sélectionnés était comparé à celui de l'AES : soit il était égalé, soit il était supérieur. La sécurité par exemple de ML-KEM et ML-DSA est comparable à celui de l'AES. Les versions AES-128 et AES-256 sont celles qui résistent aux attaques de l'ordinateur quantique.

Le RSA est vulnérable aux attaques de l'ordinateur quantique, mais résiste aux attaques de l'ordinateur classique (Si les paramètres de sécurité sont respectés). Lors du développement des algorithmes cryptographiques post-quantiques, il a été utilisé comme « un exemple à n'est pas suivre ». Les faiblesses de RSA face aux attaques quantiques, ont inspiré la création des algorithmes cryptographiques sécurisés contre les attaques quantiques. Des alternatives post-quantiques ont été développées aussi pour contrer ces faiblesses.
Le niveau de recherche actuel envisage la création de crypto-système hybride où on combine RSA et des systèmes cryptographiques post-quantiques.

Les systèmes cryptographiques dont la sécurité est basé sur le problème de logarithme discret sur un corps fini ou sur la courbe elliptique sont vulnérables aux attaques de l'ordinateur quantique, mais résistants aux attaques de l'ordinateur classique (si les paramètres de sécurité sont respectés) ; leurs vulnérabilités aux attaques quantiques sont à la base de la mise en place rapide des alternatives post-quantiques. Il s'agit clairement d'une migration vers les systèmes cryptographiques résistants aux attaques de l'ordinateur quantique.

Les systèmes cryptographiques Quantiques (QKD) sont vulnérables aux attaques quantiques, mais résistants aux attaques de l'ordinateur classique. Contrairement aux crypto-systèmes post-quantiques qui peuvent être exécutés sur un ordinateur classique, les QKD sont conçus pour tourner exclusivement sur un ordinateur quantique. Le problème qui se posait était de concevoir des algorithmes cryptographiques post-quantiques pour de systèmes non-quantiques. Le besoin était de prouver la sé-

curité des algorithmes post-quantiques sur des systèmes classiques ; les faiblesses des QKD face aux attaques quantiques a boosté la conception des crypto-systèmes post-quantiques pouvant tourner sur un ordinateur classique et quantique et dont la sécurité peut être prouvée sur un ordinateur classique.
Tous les systèmes cryptographiques post-quantiques sont créés pour résister aux attaques de l'ordinateur classique et quantique.

Seuls les algorithmes cryptographiques post-quantiques dont la sécurité repose sur les problèmes difficiles de réseaux euclidiens sont les plus résistants aux attaques de l'ordinateur quantique. Cette affirmation repose sur la conjecture de MICCIANCIO ET REGEV[**14**].

7. Implications Pratiques

Migration vers les algorithmes cryptographiques post-quantiques : Il est recommandé aux administrateurs de données bancaires, gouvernementales, médicales ; les télécoms, les objets connectés et les cyptomonaies, de migrer vers les crypto-systèmes post-quantiques pour une meilleure sécurité. **Le NIST** avait lancé un appel pour développer des systèmes cryptographiques capables de résister aux attaques de l'ordinateur quantique. En 2016, il lenca le processus de sélection pour identifier et standardiser les normes post-quantiques. En 2024, il publia les trois premiers normes post-quantiques : ML-KEM (Module-Lattice-Based Key-Encapsulation Menchanism) est un algorithme cryptographique symétrique de chiffrement et échange des clés, dont la sécurité est basée sur les problèmes difficiles des réseaux euclidiens et, en pratique peut être utilisé pour sécuriser les transactions bancaires, les objets connectés, les cryptomonaies et les télécoms ; SLH-DSA (Stateless Hach-Based Digital Signature Algorithm) et ML-DSA (Module-Lattice-Based Digital Signature Algorithme) sont des algorithmes cryptographiques de signature numérique dont la sécurité est fondée sur le problème difficile de fonctions de hachage, ils peuvent être utilisé en pratique pour les signatures électroniques ultra sécurisées , ils permettent d' authentifier l'utilisateur pour lui donner accès à un système numérique sécurisé.
Concrètement les systèmes cryptographiques post-quantiques sont déjà en train d'être déployés dans plusieurs secteurs pour protéger les données contre les attaques quantiques. Des grandes entreprises intègrent déjà les crypto-systèmes post-quantiques, par exemple le CDN (Cloudflra aux états unis) utilise l'algorithme ML-KEM pour sécuriser les communications de ses clients et échanger les clés, Thales et Samsung ont reçu des certifications pour leurs produits intégrant des algorithmes post-quantiques à bases des réseaux euclidiens, les banques et transactions financières ont commencé à intégrer les crypto-systèmes post-quantiques pour sécuriser les données contre les attaques quantiques.

8. Conclusion

L'analyse comparative de la complexité des problèmes difficiles des algorithmes cryptographiques classiques, quantiques et post-quantiques montre que les crypto-systèmes quantiques représentent d'une part une menace pour la sécurité des systèmes cryptographiques classiques et d'autre part, une alternative pour protéger les communications contre les attaques de l'ordinateur quantique. Plusieurs algorithmes cryptographiques post-quantiques standardisés sont déjà utilisés pour sécuriser les systèmes cibles contre les attaques quantiques. Le choix d'utilisation de ces algorithmes varie selon leurs caractéristiques. Ces algorithmes sont conçus pour être résistant aux attaques de l'ordinateur quantique, et peuvent tourner sur les ordinateurs classiques et quantiques. Il est donc possible de prouver leurs sécurités dans un domaine classique. Il est également possible de diminuer le temps de leurs exécutions en optimisant la gestion des clés et leurs implémentations.
Plusieurs directions futures sont envisagées : il y a par exemple la standardisation des nouvelles normes post-quantiques de chiffrement asymétrique, de signatures numériques et protocoles d'authentification ; la réduction de la complexité temporelle et même spatiale (optimisation du fonctionnement des algorithmes cryptographiques post-quantiques) ; la conception des crypto-systèmes post-quantiques hybrides en combinant les systèmes cryptographiques classiques et post-quantiques. Dans ma prochaine publication (en thèse) je prévois aller dans le sens d'optimisation des algorithmes cryptographiques post-quantiques en utilisant le calcul formel.

Référence :

[1] ARORA et BARAK, Complexity Theory : A Modern Approach, (2009) ;
[2] DONALD KNUTH, The art of Computer Programming, (1968-2011) ;
[3] DUQUESNE S., Cryptographie sur les Courbes Elliptiques, Ecole Jeunes Chercheurs, (5 avril 2005) ;
[4] ARNAULT François, Mémoire D.E.A en Maths Appliquées: Théorie des Nombres et Cryptographie, Université de Limoges, (7 mai 2002) ;
[5] BADA, H. Mémoire de maitrise. Protocoles Quantiques et relativistes de mise en Gage. Université du Québec à Montréal, (2009) ;
[6] CHRISTOS Papadimitriou, Computational Complexity, (1994) ;
[7] DENEUVILLE, J, C, Thèse de Doctorat. Contributions à la cryptographie post-quantique. Université de Limoge (2016) ;
[8] DOUGLAS STINSON, Cryptography : Theory and Practice, (1995, 4è édition, 2019) ;
[9] DUQUESNE Sylvain, thèse de Doctorat en cryptographie, Courbes Algébriques : de l'Inutile à l'Indispensable, (2007) ;
[10] CHRISTOS PAPADIMITRIOU, Computational complexity, (1994) ;
[11] HEDABOU Mustapha, Contribution à l'Implémentation des Crypto-systèmes Basés sur les Courbes Elliptiques (ECC), thèse de Doctorant en cryptographie, l'INSA Toulouse (2006) ;
[12] IZARD Thomas, Optimisation de la Multiplication Scalaire sur les Courbes Elliptiques pour de Petits Scalaires, Mémoire de master/Université Montpellier, (2008) ;
[13] A. FAUQUE, Le Partage De Clés Cryptographiques Théorie Et Pratique, Thèse de Doctorat présentée et défendue à l'Université de Limoges, Faculté des Sciences, Département de Spécialité Mathématiques, (2004-2005) ;
[14] FELDERHOFF,J . Thèse de Doctorat, Difficultés de problèmes de réseaux structurés pour la cryptographie post-quantique. Ecole normale Supérieure de Lyon, (2024) ;
[15] KATZ et LINDELL, Introduction to Modern Cryptography, (2007, 3è édition, 2020) ;

[16] KOUSSA, E. Thèse de Doctorat. Analysis and design of post-quantum cryptographic algorithms : Pkp-based signature schéma and ultra-short multivariate signatures. Université paris-Saclay. (2020) ;
[17] LIESERSON, RIVEST et STEIN, Introduction to Algorithms, (2001,3è édition 2009) ;
[18] MENEZES, Handbook of Applied Cryptography, (1996) ;
[19] MICHAEL SIPSER, Introduction to the Theory of Computation, (1997.3è édition, 2013) ;
[20] SEDGEWICK et WAYNE, Algorithms, (2011) ;
[21] SCHNEIER, Applied Cryptography, (1996) ;
[22] SCHROTTENLOHAR, A. Thèse de Doctorat Quantum Agorithms for Cryptanalysis and Quantum-Safe Symmetric cryptography. Université de Sorbone (2021) ;
[23] VANSTONE, Elliptic curve Cryptography, (2024);
[24] VASSEUR, U Thèse de Doctorat. Post-quantum Cryptography : a study of the decoding of QC-MDCPC codes. Université paris (2017) ;
[25] WILLIAM STARLLINGS, Cryptography and Network (édition 2017 et 2023);
[26] WILLIAM STARLINGS, Network security essentials (édition 2011 et 2017);
[27] WILLIM STARLINGS, Menezes et Van Oorschot et Vanstone, Handbooks of Applied cryptograghy (1996);
[28] GILLES BRASSARD, Cryptographie contemporaine (1988).