Recherche de collisions et cryptanalyse symétrique quantique

André Schrottenloher

Recherche de collisions et cryptanalyse symétrique quantique

André Schrottenloher
Sous la direction de María Naya-Plasencia
Inria, équipe-projet SECRET

13 mars 2017 – 25 août 2017

Le contexte général

Depuis la découverte décisive de l'algorithme de Shor ([Sho94]), le monde de la cryptographie s'est intéressé de près aux capacités d'un éventuel ordinateur quantique, dont l'émergence mettrait à bas la plupart des primitives asymétriques utilisées aujourd'hui. La situation en cryptographie *symétrique* est plus ambiguë : la croyance générale veut qu'un doublement de la taille des clés suffise à protéger les systèmes actuels. En effet, l'algorithme de Grover ([Gro96]) promet une accélération quadratique de tout type de recherche exhaustive. Cependant, de récents travaux ont appelé à discuter de cette affirmation péremptoire ([Kap+16a]). Mon stage s'inscrit dans la continuité de ces travaux.

Le problème étudié

Améliorer des attaques sur des primitives de cryptographie symétrique au moyen d'algorithmes quantiques a été l'un des objectifs premiers de mon stage, en plus de se familiariser avec des attaques basées sur des décalages cachés, qui continuent d'être étudiées.

Le principal problème développé pendant mon stage a été la recherche de collisions et de préimages utilisant un ordinateur quantique fortement contraint en *mémoire* (dans notre modèle, le nombre de qubits utilisés). En effet, du point de vue de l'informatique quantique, des algorithmes optimaux en nombre de requêtes sont déjà connus, mais leur coût effectif d'implémentation se révèle prohibitif : du point de vue du traditionnel produit *temps*(complexité en temps)-*mémoire*(complexité en mémoire), ils échouent à battre les techniques classiques.

Or, donner des bornes de sécurité post-quantiques précises pour la cryptographie symétrique exige de modéliser des adversaires disposant de capacités de calcul quantiques, mais limités en terme de coût (tout comme le sont les adversaires classiques).

La contribution proposée

Pour la recherche de collisions, il n'existait à ce jour aucun algorithme polynomial en nombre de qubits permettant de battre les algorithmes classiques : très souvent cité, l'algorithme BHT ([BHT98]) faisait défaut dans ce sens très strict de coût.

Ma première contribution a donc été un algorithme de recherche de collisions et de préimages améliorant le *coût* des procédures existantes dans ce sens, dans un modèle qui nous a semblé raisonnable du point de vue de la cryptanalyse symétrique. De nombreuses applications sont concernées et l'application de ces techniques dans des cas particuliers se poursuit à ce jour.

Les arguments en faveur de sa validité

En ne considérant qu'un nombre très limité de qubits et faisant abstraction de tout autre type de mémoire quantique (RAM quantique...), les algorithmes construits peuvent permettre de donner des bornes à la fois plus raisonnables et plus vraisemblables en terme de sécurité post-quantique de primitives symétriques. Ils font l'objet d'un article ([CNS17]) qui paraîtra à ASIACRYPT 2017.

Ce problème a déjà été abordé sous d'autres aspects, notamment en terme de communication ([Ber09]) ou de parallélisation efficace ([Bea+13]). Gustavo Banegas et Daniel J. Bernstein ont présenté à SAC 2017 un algorithme quantique de recherche de préimages avec une parallélisation efficace ([BB17]). Cette zone d'ombre tend donc à être comblée par la communauté.

Le bilan et les perspectives

Ce stage s'inscrit dans la tâche plus générale de rapprocher la théorie du calcul quantique et la pratique de la cryptanalyse symétrique, en cherchant comment les outils de calcul quantique à disposition peuvent être utilisés de façon raisonnable afin d'augmenter les capacités d'un adversaire. La recherche efficace de collisions

est un premier pas ; elle rend possible la description d'attaques quantiques sur de nombreuses constructions symétriques jusque-là jugées hors de portée. Par ailleurs, cette question est intéressante du point de vue algorithmique : sous une notion raisonnable de « coût », c'est la première fois que la recherche quantique de collisions se révèle plus « efficace » que son pendant classique.

L'élargissement de l'éventail des outils disponibles augure de possibles améliorations quantiques des techniques actuelles en cryptanalyse symétrique. En effet, les primitives symétriques ne pourront être jugées sûres que si elles résistent aux *attaques* post-quantiques, et pas seulement à la recherche exhaustive de Grover. Nous avons récemment commencé ce travail pour le chiffrement AES.

Ainsi, joindre la pratique de la cryptanalyse « classique » de primitives symétriques et les techniques algorithmiques quantiques sera le sujet de fond de la thèse que je commencerai l'an prochain à l'Inria, dans le cadre de l'ERC Quasymodo.

Remerciements

Mes remerciements les plus sincères vont d'abord à María Naya-Plasencia pour s'être impliquée sans limitation durant mon stage et m'avoir communiqué sa passion pour son domaine de recherche. Avec le soutien d'André Chailloux pour l'algorithmique quantique, je termine ce stage avec un précieux bagage de connaissances et, je l'espère, de rigueur ; acquises par l'entremise de nos nombreuses discussions techniques, mais aussi lors de la rédaction de notre article. Merci aussi à Xavier Bonnetain pour nos nombreux échanges, qui ont grandement contribué à me mettre à niveau et permis de profiter le plus possible de ces quelques mois de travail.

Sans pouvoir les citer tous, je remercie également les membres de l'équipe SECRET qui contribuent, par leur dynamisme et leur passion, à en faire un lieu de travail exceptionnel, que j'aurai grand plaisir à retrouver d'ici quelques mois.

Table des matières

1 Introduction

Science du secret et de la protection des données, la cryptologie se subdivise en deux grands axes que sont la **cryptographie**, consistant à construire des primitives, et la **cryptanalyse**, visant à les attaquer.

Aujourd'hui, la cryptographie se sépare en deux branches. En cryptographie symétrique où à clé privée, l'information secrète (clé de chiffrement) est connue des deux parties communiquantes (couramment nommées *Alice* et *Bob*). En cryptographie asymétrique, ou à clé publique, invention récente aujourd'hui primordiale pour tous les systèmes de communication, clés de chiffrement et de déchiffrement sont différentes : lorsqu'Alice chiffre un message pour Bob, seul celui-ci peut le déchiffrer.

Les principales familles de primitives symétriques sont les fonctions de **chiffrement** et les fonctions de **hachage**. La notion de **sécurité** de ses primitives est modélisée par des problèmes algorithmiques. Pour s'assurer de cette sécurité, il existe deux approches complémentaires : ou bien **démontrer** la difficulté du problème considéré pour un modèle raisonnable d'**adversaire**, ou bien, à l'inverse, chercher soi-même les attaques possibles.

Souvent attribuée à David Deutsch dans les années 1980, l'idée du calcul quantique est avant tout un concept de physique théorique. Nous utilisons aujourd'hui des ordinateurs reposant sur la physique de l'électromagnétisme : de notre connaissance de ces lois découle l'exactitude des calculs effectués. Mais à une conférence donnée au MIT en 1981, Feynman ([Fey82]) insiste déjà sur l'impossibilité de simuler efficacement des systèmes quantiques en utilisant des ordinateurs classiques : il pose alors les jalons d'un ordinateur quantique universel capable d'effectuer ces calculs. Deutsch a montré son existence : il est possible de simuler la mécanique quantique en construisant une telle machine (une introduction plus développée peut être trouvée dans [DEL00]).

Intéressantes pour la physique, ces questions ont rencontré un écho dans le monde de la cryptographie lorsque Peter Shor, dans les années 1990 ([Sho94]), a inventé un algorithme quantique capable de factoriser des entiers en temps *polynomial*, ainsi que de calculer des logarithmes discrets. Or, c'est sur la difficulté de ces problèmes pour des ordinateurs classiques que reposent la plupart des primitives asymétriques utilisées aujourd'hui (RSA et ses alternatives à base de courbes elliptiques).

Bien que les réalisations pratiques d'ordinateurs quantiques ne soient pas encore abouties, une hégémonie quantique dans les prochaines décennies est à craindre, sinon à envisager. Il est donc souhaitable, pour protéger les données contre de potentiels futurs attaquants quantiques, de recourir dès à présent à des primitives plus sûres, reposant sur des problèmes qui demeurent difficiles en calcul quantique (par exemple le décodage de codes linéaires quelconques). Autrement, des informations sensibles chiffrées aujourd'hui pourraient être stockées et compromises dès l'arrivée d'un ordinateur quantique. C'est pourquoi le champ de la cryptographie post-quantique est devenu très actif ([BBD09]).

Paradoxalement, la cryptographie symétrique a été jusqu'à récemment très peu affectée par la menace quantique : l'algorithme de Grover offre une accélération seulement *quadratique* à la recherche dans une base de données non triée. Un doublement de la taille des clés privées suffirait alors à assurer le même niveau de sécurité face à une recherche exhaustive accélérée quantiquement.

Les premiers résultats contre-intuitifs sur la sécurité des primitives symétriques ont été découverts par Kuwanado et Morii : ils ont montré qu'un schéma de Feistel à 3 tours ([KM10]) et que la construction d'Even-Mansour ([KM12]), pour lesquels des preuves de sécurité existent dans le domaine classique, présentaient des failles très graves face à un attaquant quantique. Plus tard, Kaplan *et al.* ([Kap+16a]) ont montré que certaines *slide attacks* classiques, transférées à un attaquant quantique, pouvaient être accélérées de façon exponentielle. Ils ont également fourni des attaques sur certains modes très courants de chiffrement authentifié comme CBC-MAC.

Dans le même temps, les notions de sécurité post-quantique se développent de manière similaire au domaine classique. Ainsi, un modèle d'oracle aléatoire autorisant les requêtes en superposition quantique a été développé ([Bon+11]). Zhandry ([Zha12]) a donné une construction de fonctions pseudo-aléatoires *post-quantiques*.

Des notions de sécurité relatives aux attaquants quantiques ont déjà été définies ([BZ13], [GHS16]) : IND-qCPA, EUF-qCMA...[1]

1. La sécurité IND-CPA est définie à l'aide d'un jeu entre l'attaquant et le *challenger*. L'adversaire peut envoyer des requêtes à texte clair choisi avant et après la phase de *challenge*. Dans celle-ci, il donne au challenger deux textes clairs x_0, x_1 et reçoit

La cryptanalyse tient un rôle fondamental en cryptographie symétrique. Les primitives de chiffrement utilisées aujourd'hui présentent des constructions complexes et souvent *ad hoc*, visant à maximiser la résistance aux attaques déjà connues. La découverte de nouvelles attaques et de techniques pour s'en prémunir a de réels impacts pratiques.

Puisque nous souhaitons attester de la sécurité face à un adversaire quantique, une véritable *cryptanalyse symétrique post-quantique* s'avère nécessaire : c'est dans le cadre de ces travaux que s'inscrit ce stage. De la même façon qu'il existe aujourd'hui un certain nombre d'outils et de techniques de cryptanalyse symétrique classique, notre objectif est de compléter le paysage des outils de cryptanalyse quantique. En particulier, nous nous sommes concentrés sur les attaques par collisions et la cryptanalyse différentielle.

La section 2 suivante présente les notions qui sous-tendent le contexte de ce rapport, en cryptographie et en calcul quantique. La section 3 détaille ensuite l'état de l'art en recherche de collisions et de préimages quantiques ; la section 4 donne le fonctionnement et la complexité de nos algorithmes tandis que la section 5 finale développe un exemple de cryptanalyse symétrique quantique.

L'algorithme présenté dans la section 4 a fait l'objet d'un article accepté à ASIACRYPT 2017 ([CNS17]), travail réalisé conjointement avec María Naya-Plasencia et André Chailloux. Ce rapport s'en inspire très fortement, sans rentrer dans les détails les plus techniques, et en ajoutant de nouvelles remarques.

2 Préliminaires

2.1 Notions de cryptographie classique

Dans tout ce qui suit, deux types de primitives symétriques sont considérés : des fonctions de **chiffrement par bloc** et des fonctions de **hachage**. Un chiffrement par bloc est une famille de permutations $\mathrm{E_K} : \{0,1\}^n \to \{0,1\}^n$ indexée par une clé K de longueur k, de description publique. En disposant de la clé, il est facile d'inverser cette permutation (déchiffrement). Au contraire, pour qui ne dispose pas de K, un chiffrement idéal oblige à considérer toutes les valeurs possibles : la recherche exhaustive nécessite alors 2^k opérations. La **sécurité** d'un chiffrement, souvent définie comme le nombre d'opérations nécessaires pour trouver K, se rapporte donc à cette attaque générique : si moins de 2^k opérations suffisent, il s'agit d'une **attaque** ; le chiffrement est cassé.

Modèle. Un modèle d'attaque très courant est l'attaque à **texte clair connu** (known-plaintext, KPA) où l'adversaire dispose de paires texte clair-chiffré aléatoires. Un peu plus puissante, l'attaque à **texte clair choisi** (CPA) donne accès à un oracle de chiffrement qui répond aux requêtes de l'adversaire. Ce nombre de requêtes est la complexité en données. De nombreux autres modèles existent, que nous ne considérerons pas, par exemple les attaques à clés liées (related-key, RKA) pour lesquelles l'adversaire effectue des requêtes à deux oracles, dont les clés sont en relation linéaire $K_1 \oplus K_2 = \delta$ avec δ une valeur connue.

Modes opératoires. Pour chiffrer en pratique un message de longueur arbitraire, on découpe celui-ci en blocs qui sont chiffrés par $\mathrm{E_K}$ et combinés au moyen d'un **mode opératoire**. Il en existe de nombreux (*Cipher Block Chaining*, CBC ; *Counter Mode*, CTR ; *Offset Codebook Mode*, OCB ...), dont les sécurités ont été étudiées d'un point de vue classique ([Bel+97]) comme quantique ([Ana+16]). Nous allons considérer notamment le mode **CBC** qui procède comme suit : le message étant découpé en l blocs de taille n $m_0, \ldots, m_{l-1}$, les chiffrés sont $c_0 = \mathrm{E_K}(m_0 \oplus IV)$ et pour tout $i \leq l-1$:

$$c_i = \mathrm{E_K}(m_i \oplus c_{i-1})$$

où IV est un vecteur d'initialisation public et aléatoire.

La taille de bloc étant n, le paradoxe des anniversaires implique que parmi $2^{n/2}$ blocs chiffrés, il y a une probabilité constante et élevée que deux d'entre eux soient égaux :

$$\mathrm{E_K}(m_i \oplus c_{i-1}) = \mathrm{E_K}(m_j \oplus c_{j-1})$$

en retour le chiffré de l'un des deux, il doit alors indiquer auquel des deux ce chiffré correspond. Dans le modèle IND-qCPA, strictement plus puissant que IND-CPA ([Bon+11]), l'oracle de chiffrement auquel l'adversaire a accès est un oracle quantique, les requêtes sont posées en superposition.

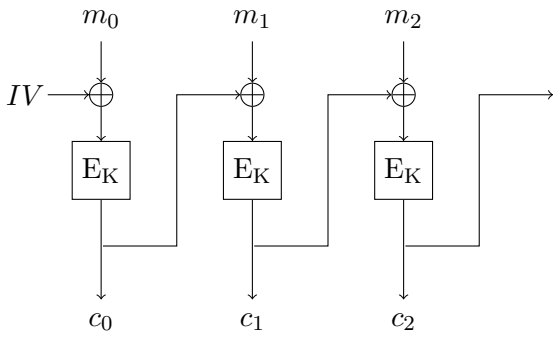

FIGURE 1 – Mode CBC

On obtient alors $m_i \oplus c_{i-1} = m_j \oplus c_{j-1}$ donc $m_i \oplus m_j$, le XOR de deux textes clairs. Il suffit que l'un des deux soit connu et l'autre recherché. Ceci implique qu'il faut toujours chiffrer (beaucoup) moins que $2^{n/2}$ blocs avec la même clé.

Fonctions de hachage. Dans toute la suite de ce document, nous nommerons H une **fonction de hachage**. Étant donné un message m de longueur arbitraire, elle calcule un haché $H(m) = h$ de longueur fixe n. Les problèmes suivants doivent alors n'admettre aucune solution plus rapide que les méthodes génériques :

- Trouver une **collision** : x, y tels que $H(x) = H(y)$. La technique générique consiste à effectuer $\Omega(2^{n/2})$ requêtes et chercher la collision promise par le paradoxe des anniversaires ;
- Trouver une **préimage** : étant donné h, m tel que $H(m) = h$. La technique générique consiste en une recherche exhaustive en $\Theta(2^n)$ opérations ;
- Trouver une **seconde préimage** : étant donné m et $H(m)$, m' tel que $H(m') = H(m)$. La technique générique effectue une recherche exhaustive en $\Theta(2^n)$ opérations.

2.2 Recherche de collisions et de préimages multi-cibles

Dans ce document, deux problèmes vont particulièrement nous préoccuper.

Problème 2.1 (Recherche de collisions)**.** *Étant donné accès à une fonction aléatoire* $H : \{0,1\}^n \to \{0,1\}^n$*, trouver* $x, y \in \{0,1\}^n$ *avec* $x \neq y$ *tels que* $H(x) = H(y)$*.*

Nous considérons une fonction aléatoire pour nous placer au mieux dans un contexte cryptographique ; H doit être envisagée comme une fonction de hachage.

Problème 2.2 (Recherche de préimages multi-cibles)**.** *Étant donné accès à une permutation aléatoire* $H : \{0,1\}^n \to \{0,1\}^n$ *et un ensemble* $T = \{y_1, \ldots, y_{2^t}\}$*, trouver la préimage de l'un des* y_i *par* H *i.e. trouver* $i \in \{1, \ldots, \ell\}$ *et* $x \in \{0,1\}^n$ *tel que* $H(x) = y_i$*.*

Contrairement à certains travaux dans la littérature ([Bra+02]), nous ne considérons pas le cas des fonctions r-vers-1. Les algorithmes que nous présenterons plus loin peuvent s'y adapter sans difficulté et cela ne ferait qu'en simplifier l'étude.

2.3 Attaques par collisions en cryptographie symétrique.

En dehors des fonctions de hachage que nous venons de voir, la recherche de collisions et de préimages apparaît très souvent en cryptographie symétrique, ce qui motive les algorithmes quantiques que nous nous apprêtons à développer.

Utilisateurs multiples. Un scénario d'attaque déjà étudié dans la littérature ([Bih02 ; BMS06 ; CMS12 ; Men12]) considère un adversaire essayant de retrouver la clé d'un utilisateur parmi un grand nombre, qui chiffrent le même message m. En considérant la fonction $K \to \mathrm{E}_K(m)$, il s'agit de retrouver une des préimages K.

En général, si K est de longueur k, une quantité de 2^t données disponibles (et, par conséquent, 2^t requêtes de chiffrement) permet de descendre le coût d'une recherche à 2^{k-t} calculs.

Collisions sur des modes opératoires. Nous reprenons ici l'exemple du mode CBC. Au bout de $2^{n/2}$ blocs chiffrés avec la même clé, des collisions apparaissent et divulguent de l'information. La communauté recommande donc de limiter le nombre de blocs à $\ell \ll 2^{n/2}$, ce que les applications ne respectent pas toujours. Les auteurs de [BL16] ont mis en évidence une telle attaque, contre des chiffrements à 64 bits utilisant CBC (il est également possible, au prix d'un peu plus de travail, d'attaquer le mode CTR).

Briques de base. Un algorithme pour la recherche de collisions ou de préimage est susceptible d'entrer dans la composition de techniques d'attaques telles que les différentielles tronquées [Knu94] ou impossibles [Knu98 ; BBS99] où l'adversaire a besoin de trouver des collisions partielles.

2.4 Algorithmes classiques de recherche de collisions

Collisions. Le paradoxe des anniversaires permet de trouver une collision en temps et en mémoire $O(2^{n/2})$ (il suffit de faire $2^{n/2}$ requêtes et de trier les réponses). L'algorithme rho de Pollard [Pol75] rend la complexité mémoire négligeable et résout le problème 2.1 en temps $O(2^{n/2})$. Avec un seul processeur, aucun algorithme classique ne permet de descendre plus bas.

Collisions en parallèle. Une méthode de réduction de la complexité en temps par parallélisation a été proposée dans [OW94]. Le produit temps-mémoire ne descend pas en dessous de $O(2^{n/2})$. L'idée est de calculer de nombreuses chaînes en parallèle et de considérer un ensemble arbitraire de *points distingués* sur lesquels s'arrêtent les chaînes. La liste des points distingués découverts est partagée et permet de découvrir d'éventuelles collisions. Sur m processeurs, avec une proportion θ de points distingués, la complexité en temps devient $O(2^{n/2}/m + 2.5\theta)$.

Attaques préimage multi-cibles. Le meilleur algorithme classique pour résoudre le problème 2.2 trouve une des $\ell = 2^t$ préimages par recherche exhaustive, en temps $\Omega(2^{(n-t)})$ (voir par exemple [And+08]), la probabilité de trouver une des 2^t préimages étant $\frac{2^t}{2^n}$. Il est également possible de paralléliser efficacement ce calcul.

2.5 Principes de calcul quantique

Nous rappelons ici quelques principes fondamentaux du calcul quantique. Une présentation plus détaillée peut être trouvée dans [DW13] ou [NC02].

Qubits. Dans un ordinateur classique, les unités de calcul sont des **bits** valant 0 ou 1. Dans un ordinateur quantique, des objets nommés **qubits** les remplacent, qui peuvent être dans une superposition de deux états $|0\rangle$ ou $|1\rangle$. Ainsi, un qubit est représenté par un vecteur dans un espace de Hilbert

$$\alpha\,|0\rangle + \beta\,|1\rangle$$

où α et β sont des amplitudes complexes telles que $|\alpha|^2$ et $|\beta|^2$ sont les probabilités d'obtenir respectivement $|0\rangle$ et $|1\rangle$ lorsque le qubit est **mesuré**. Par conséquent, $|\alpha|^2 + |\beta|^2 = 1$.

Alors que le qubit semble contenir une quantité invraisemblable d'information, la nécessité d'une mesure finale impose qu'on ne peut lire, après un calcul impliquant n qubits, qu'une chaîne de caractères à n bits.

Intrication. En combinant deux qubits, qui sont des vecteurs dans l'espace de Hilbert $\mathcal{H}$, le système résultant n'est pas dans $\mathcal{H}^2$ (contrairement à des chaînes de caractère), mais dans $\mathcal{H}^{\otimes 2}$, le produit tensoriel de ces espaces. Concrètement, un système de deux qubits peut être dans une superposition quelconque des états $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, et ne peut donc pas être *a priori* décrit comme une simple *paire* de qubits. L'espace $H^{\otimes n}$ obtenu avec n qubits **intriqués** a dimension 2^n.

Modèle de circuit quantique. Dans ce modèle standard, que nous adoptons, un algorithme quantique est une série d'opérations élémentaires ou **portes quantiques** appliquées à un système de qubits, de façon analogue aux portes de logique classique. Ces portes sont des opérateurs unitaires de $\mathcal{H}^{\otimes r}$. Par exemple,

la porte de Hadamard à un qubit H effectue l'opération suivante : $H\ : |x\rangle \to \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x |1\rangle)$ et se généralise en :

$$H^{\otimes n}\ : |x\rangle \to \frac{1}{2^n} \sum_{y\in\{0,1\}} (-1)^{x\cdot y} |y\rangle$$

.

Le calcul s'achève par une mesure, qui projette le vecteur sur l'un des éléments de la base canonique avec les probabilités correspondantes. Le plus souvent, un algorithme quantique est probabiliste. Tant que la probabilité de réussite est constante, répéter plusieurs fois le calcul permet de la rendre arbitrairement proche de 1.

Unitarité et réversibilité. Un algorithme quantique étant une composition d'opérateurs unitaires, il peut être facilement inversé : tout calcul quantique est **réversible**, l'inverse de $\mathcal{A}$ est noté $\mathcal{A}^\dagger$. La porte de Toffoli T est universelle pour la logique réversible, il suffit de l'associer avec la porte de Hadamard H pour obtenir un ensemble universel pour le calcul quantique.

Tout calcul quantique fait usage d'un certain nombre de qubits ancillaires initialisés à l'état $|0\rangle$ et retournés à l'état $|0\rangle$ à la fin du calcul. Nous les omettrons le plus souvent, bien que leur compte précis intervienne dans le calcul des ressources nécessaires aux algorithmes quantiques.

Complexité en temps et mémoire. Il existe différentes notions de complexité relatives au modèle des circuits quantiques. Généralement, le **temps** est représenté par la profondeur du circuit (le nombre d'opérations élémentaires appliquées successivement sur le système), tandis que la **mémoire** est le nombre de qubits utilisés (ancillaires compris, mais souvent omis par simplicité). Il nous faut remarquer que la mémoire quantique, sous cette définition, n'est pas analogue à la mémoire classique que nous voyons comme une simple base de données indexée. Qui dit plus de qubits, dit plus d'opérations pouvant être appliquées en parallèle. Par ailleurs, la mémoire quantique est susceptible d'être beaucoup plus chère que son analogue classique : c'est un modèle de *coût* que nous garderons à l'esprit dans ce qui suit.

Modèle d'oracle quantique. Toute fonction $f\ : \{0,1\}^n \to \{0,1\}^m$ avec une description connue peut être implémentée sous la forme d'un opérateur unitaire O_f sur $n+m$ qubits :

$$O_f\ : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle$$

Un autre oracle est :

$$O'_f\ : |x\rangle \mapsto (-1)^{f(x)} |x\rangle$$

Une requête à O_f peut être simulée avec deux requêtes à O'_f.

Le temps d'exécution de O'_f est deux fois celui de f. En effet, on effectue d'abord le calcul de cette fonction, puis on remet les qubits ancillaires à leur état initial en appliquant l'inverse de ce calcul : l'oracle n'a ainsi aucun effet de bord.

2.6 Amplification d'amplitude

Notre outil principal est la procédure d'**amplification d'amplitude**.[2] Il s'agit d'une généralisation du principe de l'algorithme de Grover.

Théorème 2.1 ([Bra+02], Amplification d'amplitude)**.** *Soit $f\ : \{0,1\}^n \to \{0,1\}$ une fonction de test et $\mathcal{A}$ un algorithme quantique qui produit, sur l'entrée $|0\rangle^n$, un état $|\phi\rangle = \alpha |\phi_G\rangle + \beta |\phi_B\rangle$ où $|\phi_G\rangle$ est une superposition de « bons » états vérifiant $f(x) = 1$ et $|\phi_B\rangle$ une superposition de « mauvais » états avec $f(x) = 0$. On note $|\alpha| = \sin(\theta)$ pour un certain $\theta \in [0, \pi/2]$. Alors, il existe un algorithme quantique qui :*

- *Contient exactement $N = \left\lfloor \frac{\pi}{4\theta} - \frac{1}{2} \right\rfloor$ appels à $O_f, O_f^\dagger = O_f\mathcal{A}, \mathcal{A}^\dagger$, où O_f est l'oracle :*

$$O_f(|x\rangle |b\rangle) = |x\rangle |b \oplus f(x)\rangle$$

- *Produit un état proche de $|\phi_G\rangle$. Des calculs d'erreurs détaillés sont effectués dans [CNS17].*

2. Dans [CNS17], cette procédure est présentée en utilisant un projecteur. Pour simplifier, nous remplaçons ce dernier par une plus simple fonction de test, sans conséquence pour la justesse de cette présentation.

Nous appelons $\mathcal{A}$ la préparation *et f le* test*. Cette procédure générique s'écrit donc :*

$$\texttt{QAA}(\texttt{prep}, \texttt{test}) = \texttt{QAA}(\mathcal{A}, f)$$

et son temps d'exécution est

$$N\left(|\mathcal{A}|_{RT} + |O_f|_{RT}\right).$$

où $|\cdot|_{RT}$ représente le temps d'exécution des algorithmes. Si $|\mathcal{A}|_{RT}$ et $|O_f|_{RT}$ sont polynomiaux en n, le résultat est bien sûr $\widetilde{O}(N)$.

L'**algorithme de Grover** est un cas particulier d'amplification d'amplitude, dans lequel on recherche un élément tel que $f(x) = 1$. L'algorithme $\mathcal{A}$ produit la superposition de toutes les possibilités $|\phi\rangle = \frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} |x\rangle$ en appliquant une porte de Hadamard sur chaque qubit de l'entrée $|0\rangle^n$.

On trouve x tel que $f(x) = 1$ avec grande probabilité en temps $\widetilde{O}\left(\sqrt{\frac{2^n}{|\{x:f(x)=1\}|}}\right)$. Lorsqu'il y a une seule possibilité parmi 2^n, la complexité atteint $O\left(\sqrt{2^n}\right)$, produisant ainsi un gain quadratique par rapport à la recherche exhaustive classique (à condition de pouvoir implémenter O_f). Le gain systématique de l'algorithme de Grover est optimal (une preuve assez simple de ce fait peut être trouvée dans [Gro98]). Il s'agit d'un ingrédient incontournable de cryptanalyse symétrique quantique.

Parallélisation. Une question pertinente dans le cadre de l'algorithme de Grover est l'impact que peut avoir l'ajout de plus de mémoire quantique, donc de qubits. En effet, la recherche exhaustive parmi 2^n éléments se parallélise de manière évidente sur 2^s processeurs classiques avec une complexité en temps $\frac{2^n}{2^s}$. Dans le cas de Grover, ou de toute autre amplification d'amplitude, supposons que le nombre de qubits soit multiplié par 2^s. Dans ce cas, nous effectuons en parallèle 2^s copies du calcul et arrêtons les itérations lorsque la probabilité de réussite approche $\frac{1}{2^s}$, car cela nous donnera une bonne solution avec probabilité constante. Dans ce cas, le nombre d'itérations nécessaire est de l'ordre de $\sqrt{\frac{2^n}{2^s}}$: le gain de la parallélisation est très inférieur au cas classique. Le produit temps-mémoire, d'ailleurs, devient $\sqrt{2^{n+s}}$.

Zalka ([Zal99]) a montré qu'il était impossible, en utilisant 2^s processeurs quantiques en parallèle, de faire moins de $\widetilde{O}\left(\sqrt{\frac{2^n}{2^s}}\right)$ itérations : ainsi, la parallélisation naïve est optimale.

2.7 Adversaire quantique

Deux modèles peuvent être considérés pour un adversaire quantique, et les scénarios d'attaque diffèrent selon le modèle. Nous reprenons ici une terminologie déjà présente dans [Kap+16b].

Modèle Q_1. L'adversaire a accès à un ordinateur quantique. Il est capable d'implémenter ses propres oracles O_f pour les fonctions f qui lui sont connues et effectuer des requêtes en superposition, mais il ne *peut pas* effectuer une requête en superposition à l'oracle de chiffrement. C'est le modèle *IND-CPA quantique* vu dans [Bon+11 ; Bra+11 ; Zha15b ; Unr15].

Modèle Q_2. L'adversaire peut effectuer des requêtes en superposition à un oracle de chiffrement (IND-qCPA), voire de déchiffrement (IND-qCCA). Ce modèle a déjà été considéré dans [Dam+13 ; BZ13 ; Zha12 ; Kap+16a ; GHS16 ; Bon+11].

Il s'agit d'un modèle strictement plus fort que Q1 (c'est montré dans [Bon+11]), qui présente de nombreux avantages d'un point de vue théorique :

- Il s'agit d'un modèle simple et inclusif ;
- Ce modèle permet de prévenir les faiblesses de toute primitive symétrique et des constructions sous-jacentes : ainsi, dans le cas des fonctions de hachage ou dans un modèle *known-key*, l'adversaire peut implémenter lui-même l'oracle ;
- Plutôt que de tenter de se prémunir des mauvais cas d'utilisation, mieux vaut certifier la sécurité dans le modèle le plus fort disponible ;
- Des preuves de sécurité peuvent être construites dans ce modèle, tout à fait non-trivial.

2.8 Autres questions en cryptanalyse quantique symétrique

Les travaux récents menés en cryptanalyse quantique symétrique couvrent un champ plus large. Ces autres applications, qui ne seront pas développées dans ce rapport, concernent par exemple :

- L'étude des *slide attacks* quantiques utilisant les algorithmes de Simon, de Kuperberg et leur combinaison ([BNP17]) ;
- La combinaison de l'algorithme de Grover avec des sous-procédures probabilistes et quantiques (par exemple l'algorithme de Simon, voir à ce sujet [LM17]) ;
- L'étude d'algorithmes quantiques tels que l'algorithme de Kuperberg, en calculant le plus précisément possible la complexité de ces procédures et leurs coûts effectifs ([BNP17]) ;
- De nouvelles attaques utilisant l'algorithme de Simon, comme avec Keyak dans [LL17].

3 État de l'art

Nous développons maintenant les algorithmes quantiques déjà connus pour les problèmes 2.1 et 2.2, leur complexité et leurs limites à un usage cryptographique.

3.1 Recherche quantique de collisions

La recherche quantique de collisions a été étudiée en premier par Brassard, Høyer et Tapp ([BHT98]). Leur algorithme utilise Grover comme sous-routine et trouve une collision pour une fonction f 2-vers-1 en $\widetilde{O}(2^{n/3})$ requêtes à O_f.

Des bornes inférieures en nombre de requêtes ont ensuite été démontrées ([AS04 ; Amb05 ; Kut05]) jusqu'à atteindre $\Omega(2^{n/3})$. Zhandry a étendu le problème aux fonctions aléatoires ([Zha15a]) et montré que cette borne tenait toujours.

Rappelons l'algorithme de [BHT98]. Avec les outils que nous introduisons par la suite, il deviendra évident que cet algorithme nécessite soit une complexité en mémoire quantique $\widetilde{O}(2^{n/3})$ pour un temps $\widetilde{O}(2^{n/3})$, soit un temps total $\widetilde{O}(2^{2n/3})$ pour une quantité de mémoire polynomiale en n.

Algorithme 1: Algorithme BHT de recherche de collisions ([BHT98])

Entrée. Accès en superposition à la fonction 2-vers-1 $H : \{0,1\}^n \to \{0,1\}^n$ (oracle O_H).
Oracle d'appartenance. Calculer H sur un ensemble arbitraire T de taille $2^{n/3}$, pour obtenir $H(T)$. L'oracle d'appartenance O détermine si $H(x) \in H(T)$ pour une requête x en superposition. Ou bien chaque requête à O nécessite un temps $\widetilde{O}(|T|) = \widetilde{O}(2^{n/3})$, ou bien toutes sont effectuées en temps polynomial, mais l'implémentation requiert $\widetilde{O}(2^{n/3})$ qubits.
Algorithme de Grover. Utiliser l'algorithme de Grover pour trouver x tel que $x \notin T \wedge H(x) \in H(T)$. L'espace de recherche est $\{0,1\}^n$, de taille 2^n, la fonction de test est $g(x) = 1 \iff x \notin T \wedge H(x) \in T$ calculée à l'aide d'une requête à O_H et à O.
Le nombre de bons états est $|\{x, H(x) \in H(T) \wedge x \notin T\}|$, soit $|T| = 2^{n/3}$ puisque H est 2-vers-1. Cette instance de Grover nécessite $\widetilde{O}(\sqrt{2^{n-n/3}}) = \widetilde{O}(2^{n/3})$ itérations.

Limites des travaux existants. Comme nous l'avons dit, l'algorithme BHT n'est pas satisfaisant, car il consomme un total temps-mémoire $\widetilde{O}(2^{2n/3})$, là où la recherche de collisions classique arrive aisément à $O(2^{n/2})$. Ambainis ([Amb07]) a construit un algorithme pour le problème des éléments distincts (*element distinctness*), à savoir déterminer si 2^n éléments accessibles en superposition sont tous distincts ou non, en temps $O\left(2^{2n/3}\right)$, qui implique une solution pour la recherche de collision en temps $O\left(2^{n/3}\right)\ldots$ mais toujours au moyen de $O\left(2^{n/3}\right)$ mémoire quantique disponible. [3]

La question de savoir s'il était possible, au moyen d'un ordinateur quantique, d'améliorer sensiblement le *coût* véritable des algorithmes de recherche de collision, a été posée par Grover et Rudolph ([GR04]). En particulier, réaliser des algorithmes avec peu de mémoire quantique est d'un grand intérêt pour la cryptographie, car ils sont susceptibles de modéliser des adversaires plus réalistes. En 2009, Berstein ([Ber09]) a remarqué qu'aucun algorithme quantique disponible ne représentait de menace pour les fonctions de hachage, du fait

3. Cette réduction est très simple : on ramène une recherche de collision parmi 2^n éléments au problème des éléments distincts parmi $2^{n/2}$, en sélectionnant un sous-ensemble arbitraire.

de leur coût temps-mémoire *supérieur ou égal* au classique $2^{n/2}$, même si un ordinateur quantique à grande échelle venait à voir le jour.

Parallélisation des algorithmes quantiques. Une réponse au problème posé par Grover et Rudolph, à savoir démontrer un produit temps-mémoire meilleur qu'en classique pour le problème de collision et des éléments distincts, apparaît dans [Bea+13]. Sans entrer dans les détails ici, il est possible d'arriver, avec S qubits, à $T \times S = O(2^n)$ pour les éléments distincts sur 2^n éléments et $T \times S = O(\sqrt{2^n})$ pour la collision. Avec un nombre de qubits polynomial en n, il n'existait donc pas à ce jour d'algorithme quantique plus efficace en temps que les procédures classiques.

Modèle de mémoire quantique. Dans [GR04], les auteurs font remarquer que *mémoire quantique* et *processeurs quantiques* sont des notions qui se rejoignent en parlant de qubits. Il existe d'autres modèles, par exemple celui de la RAM quantique, contenant des données fixées mais accessible en superposition. Par ailleurs, les algorithmes quantiques avec beaucoup de mémoire peuvent poser des problèmes en terme de *communication* entre leurs différents processeurs, qui doivent être organisés en fonction. C'est aussi ce qui motive la remarque de Bernstein ([Ber09]) selon laquelle le calcul quantique ne pouvait pas prétendre à améliorer la cryptanalyse des fonctions de hachage, et sous-tend le récent travail présenté à SAC 2017 ([BB17]) concernant la recherche de préimages multi-cibles.

Intérêt de la communauté. Différents travaux indépendants sont apparus au cours de ce travail, comme [BB17] ou [TU17], où Ebrahimi et Unruh étudient la résistance post-quantique aux collisions de fonctions aléatoires distribuées non uniformément, montrant un intérêt encore renouvelé pour ces questions.

3.2 Recherche quantique de préimages multi-cibles

La recherche de préimages multi-cibles dans le domaine quantique a été nettement moins étudiée. De façon simultanée et indépendamment de nos travaux, Banegas et Bernstein ([BB17]) ont présenté à SAC 2017 un algorithme qui résout le problème de recherche de préimages multi-cibles en temps $\sqrt{N/pt}$ pour p processeurs et t cibles, ou $\sqrt{N/pt^{1/2}}$ lorsque les processeurs, disposés en deux dimensions, ne peuvent communiquer qu'avec leurs voisins. Mais cet algorithme nécessite toujours d'avoir $p > t$ et ne s'applique donc pas lorsque le nombre de qubits est très faible.

Pour donner un exemple simple et représentatif de notre point de départ, nous adaptons plutôt les idées de [BHT98].

Algorithme 2: Préimages multi-cibles avec Grover ([BHT98])

Entrée. L'ensemble $T = \{y_1, \dots y_{2^t}\}$ des cibles, un accès en superposition à $H : \{0,1\}^n \to \{0,1\}^n$ (oracle O_H).
Oracle d'appartenance. L'algorithme utilise un oracle O_T d'appartenance à T. Une requête coûte un temps $\widetilde{O}(|T|) = \widetilde{O}(2^t)$, qui peut être transféré sous forme de mémoire.
Algorithme de Grover. Chercher x tel que $H(x) \in T$ avec Grover. L'espace de recherche est l'ensemble des textes clairs possibles $\{0,1\}^n$ de taille 2^n, la fonction de test est $g(x) = 1 \iff H(x) \in T$. Calculer g requiert une requête à O_H et une requête à O_T.
Le nombre de bons états est $|\{x, H(x) \in T\}| = |T|$ puisque H est une permutation. Le nombre d'itérations est donc de l'ordre de $\sqrt{2^{n-t}}$.

On remarque alors que le produit temps-mémoire est toujours $\widetilde{O}(2^t 2^{\frac{n-t}{2}}) = \widetilde{O}(2^{\frac{n+t}{2}})$. Le coût serait donc minimisé pour $t = 0$, soit une seule préimage : dans ce modèle de coût, l'algorithme dont nous disposons est incapable de tirer un réel avantage de la présence de cibles multiples, et revient à utiliser l'algorithme de Grover sur une seule cible (comme cela est fait, par exemple, dans [Amy+16]).

3.3 Nouvel algorithme et comparaison avec les précédents

J'ai proposé pendant ce stage une nouvelle technique permettant de réduire sensiblement les produits temps-mémoire des problèmes 2.1 et 2.2, notamment lorsque l'on ne dispose que de $O(n)$ qubits, en passant pour la première fois en dessous du produit temps-mémoire classique $O(2^{n/2})$. Cet algorithme a donné lieu

à un travail commun avec André Chailloux et María Naya-Plasencia ([CNS17]), qui paraîtra à ASIACRYPT 2017.

Avant de le présenter en détail dans ce qui suit, nous comparons les produits temps-mémoire dans différents cas : classique, quantique, en utilisant les algorithmes 1 et 2, puis en utilisant nos nouveaux algorithmes ; enfin, en parallélisant notre procédure. Nous utilisons la variable s pour signifier que 2^s processeurs sont disponibles et t pour 2^t cibles.

Nos algorithmes remplacent la consommation excessive de mémoire quantique précédente par une utilisation de mémoire *purement classique*. Nous l'excluons des produits temps-mémoire, les quantités demandées présentant un coût modique.

Là où tous les algorithmes classiques sont cantonnés à un produit temps-mémoire $O(2^{n/2})$, notre recherche de collisions est en-dessous pour $s \leq \frac{n}{4}$. Si $s \leq \frac{n}{6}$, la recherche de préimages multi-cibles pour $t \geq \frac{3n+3s}{7}$ également. Lorsque s tends vers $\frac{n}{4}$, la parallélisation de notre algorithme de recherche de collisions devient impossible et son coût rejoint celui de la parallélisation quantique efficace présentée dans [Bea+13]. Ces résultats sont regroupés dans les tables 1, 2 et, pour les collisions, dans la figure 3.3.

TABLE 1 – Algorithmes pour la recherche de collision. La dernière ligne est valide pour $s \leq n/4$.

	Temps	Mémoire Q.	Mémoire C.	# Processeurs
Grover ([BHT98])	$2^{n/3}$	$2^{n/3}$	-	$2^{n/3}$
Algorithme d'Ambainis ([Amb07])	$2^{n/3}$	$2^{n/3}$	-	1
Parallélisation classique ([OW94])	$2^{n/2-s}$	-	2^s	2^s
Parallélisation quantique ([Bea+13])	$2^{n/2-s}$	-	2^s	2^s
Notre travail – un processeur	$2^{2n/5}$	$O(n)$	$2^{n/5}$	1
Notre travail – parallélisation	$2^{2n/5-3s/5}$	$O(2^s n)$	$2^{n/5+s/5}$	2^s

TABLE 2 – Algorithmes pour la recherche de préimages multi-cibles. Nous considérons 2^s processeurs pour les trois algorithmes parallèles et un seul pour les autres. L'algorithme de [BB17] ne peut pas considérer plus de cibles qu'il n'a de processeurs disponibles.

	Temps	Mémoire Q.	Mémoire C.	# Proc.
Classique	2^{n-t}	-	2^t	1
Parallélisation classique	2^{n-t-s}	-	$2^t + 2^s$	2^s
Algorithme quantique naïf	$2^{n/2}$	$O(n)$	-	1
Algorithme de [BB17] à faible communication	$2^{n/2-3s/4}$	2^s	-	2^s
Algorithme de [BB17] à communication libre	$2^{n/2-s}$	2^s	-	2^s
Notre travail – un processeur	$2^{n/2-t/6} + \min\{2^t, 2^{3n/7}\}$	$O(n)$	$\min\{2^{t/3}, 2^{n/7}\}$	1
Notre travail – parallélisation	$2^{n/2-t/6-s/2} + \min\{2^t, 2^{\frac{3n-4s}{7}}\}$	$O(2^s n)$	$\min\{2^{t/3}, 2^{n/7+s/7}\}$	2^s

4 Un nouvel algorithme quantique pour les collisions

L'algorithme que j'ai proposé durant mon stage et qui va être détaillé ici part d'une idée simple. Nous avons retrouvé une idée similaire dans [Gro02] après l'avoir appliquée à notre contexte : il s'agit de gagner en temps global en équilibrant le coût entre les requêtes d'oracle et les autres calculs.

En effet, l'algorithme de Brassard *et al.* contient $\widetilde{O}\left(2^{n/3}\right)$ appels à l'oracle d'appartenance et chacun d'eux, si peu de mémoire quantique est disponible, requiert $O(2^{n/3})$ calculs : nous avons donc essayé de réduire ce coût, quitte à augmenter le nombre de requêtes à l'oracle O_H.

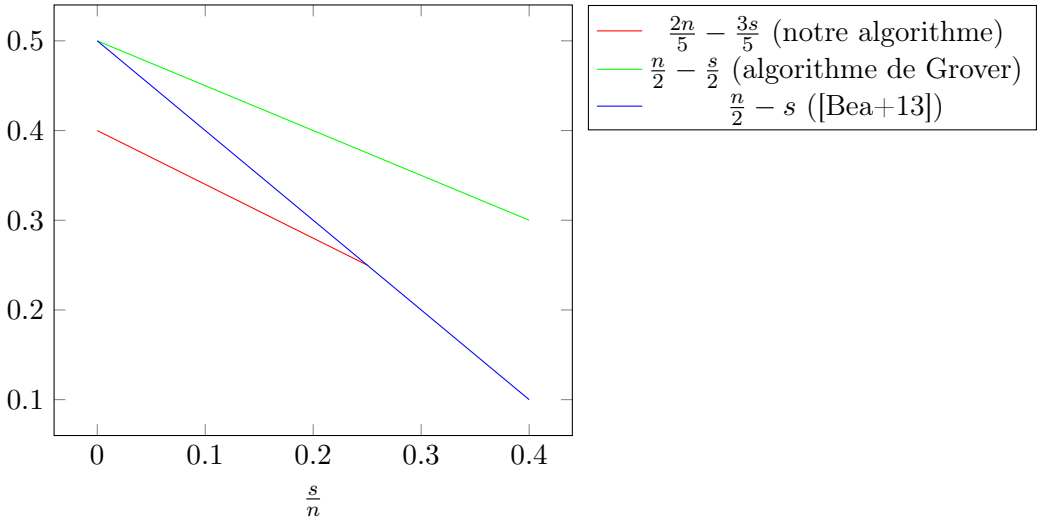

FIGURE 2 – Logarithme en base 2 de la complexité en temps (fraction de n), en fonction de la quantité de mémoire quantique

D'une consommation supplémentaire de mémoire *classique*, et non quantique, va découler cet algorithme employant un nombre restreint de qubits.

4.1 Oracle d'appartenance

Pour commencer, nous devons formaliser et décrire un oracle d'appartenance à un ensemble T, *strictement séquentiel*, puisque nous souhaitons l'utiliser avec peu de mémoire quantique. Sans surprise, le coût en temps de cet oracle est linéaire en le nombre d'éléments de T.

Définition 4.1. *Soit T un ensemble (non structuré) de 2^t chaînes de n caractères. Un oracle d'appartenance classique à T calcule : $f_T(x) = 1$ si $x \in T$ et 0 sinon. Un oracle d'appartenance quantique est un opérateur O_T qui calcule f_T :*

$$O_T(|x\rangle\,|b\rangle) = |x\rangle\,|b \oplus f_T(x)\rangle$$

Modèle de calcul et de mémoire considéré. L'ensemble $T = \{x_1, \ldots, x_{2^t}\}$ considéré est stocké dans une mémoire **classique** qui est parcourue de façon séquentielle. Les éléments sont obtenus les uns après les autres en temps 1. On considère, au prix d'un éventuel précalcul, que T ne contient pas de doublons. L'interface quantique se situe ensuite au niveau des opérations suivantes :

- Un algorithme de **création** qui, sur une entrée classique x de n bits et un registre de n qubits initialisés à $|0\rangle$, renvoie $|x\rangle$. Un temps n suffit (considérer séparément chaque qubit).
- Un opérateur de **comparaison** unitaire :

$$\forall x, y \in \{0,1\}^n, \forall b \in \{0,1\},\ COMP(|x\rangle\,|y\rangle\,|b\rangle) := |x\rangle\,|y\rangle\,|b \oplus \delta_{xy}\rangle\,.$$

- Un algorithme de **destruction** qui sur l'entrée $x, |x\rangle$, renvoie $|0\rangle$. Il suffit d'inverser l'opérateur de création correspondant.

Un appel à $O_T\,|x\rangle\,|b\rangle = |x\rangle\,|b \oplus f_T(x)\rangle$ consiste alors à tester chaque élément $x_i \in T$ **séquentiellement** en utilisant $COMP$. Les états successifs sont notés $\phi_1, \ldots \phi_i$.

Algorithme 3: Oracle d'appartenance

Entrée : $|\phi_1\rangle := |x\rangle\,|b\rangle$ à $n+1$ qubits. Pour $i = 1 \ldots 2^t$:

- Récupérer $x_i \in T$ et construire le registre $|x_i\rangle$ auquel on concatène l'état $|\phi_i\rangle$.
- Appliquer $COMP$ à l'état $|x_i\rangle\,|\phi_i\rangle$.
- Détruire le premier registre, $|\phi_{i+1}\rangle$ est l'état restant.

L'état final $|\phi_{2^t+1}\rangle$ est exactement égal à $|x\rangle\,|b \oplus f_T(x)\rangle$.

Proposition 4.1. *L'algorithme 3 implémente O_T parfaitement, en temps $n2^t$ avec $2n+1$ qubits.*

La preuve est par une induction immédiate. Un argument de comptage permet de remarquer que mis à part des facteurs logarithmiques, il est impossible d'améliorer le temps d'exécution de O_T si l'ensemble T est pris au hasard (sans structure) parmi tous les ensembles à 2^t éléments parmi 2^n possibles. Lorsque 2^s registres à $2n+1$ qubits sont disponibles, il est possible de partitionner T en 2^s sous-ensembles et d'effectuer les comparaisons en parallèle : le temps est alors réduit à $n2^{t-s}$.

4.2 Recherche quantique de collisions

Munis d'une implémentation pour O_T, nous allons maintenant détailler notre algorithme de recherche de collisions. Il va réaliser l'équilibre promis entre requêtes à l'oracle O_T et à la fonction H, en utilisant l'amplification d'amplitude (théorème 2.1) à plusieurs reprises.

Notons que contrairement à la description initiale de l'algorithme BHT ([BHT98]), nous considérons déjà le cas de fonctions aléatoires, plus inclusif et utile pour la cryptographie que celui des fonctions 2-vers-1 qui en constitue une bonne approximation.

Algorithme 4: Notre algorithme de recherche de collisions

Accès est donné à l'oracle O_H pour une fonction aléatoire $H : \{0,1\}^n \to \{0,1\}^n$. L'algorithme renvoie une collision, (x, x') tels que $x \neq x'$ et $H(x) = H(x')$). Les paramètres r et t sont fixés au cours du calcul et seront optimisés plus tard.

Pour $r \in [1, \ldots, n]$, soit $S_r^H := \{(x, H(x)) : \exists z \in \{0,1\}^{n-r}, H(x) = \underbrace{0 \ldots 0}_{r \text{ times}} || z\}$. S_r^H consiste des paires entrée / sortie $(x, H(x))$ telles que $H(x)$ commence avec r zéros. Les étapes de l'algorithme sont :

1. Construire une liste L consistant en 2^{t-r} éléments de S_r^H. Soit $f_L^H(x) := 1$ si $\exists (x', H(x')) \in L,\ H(x) = H(x')$ et $f_L^H(x) := 0$ sinon.
2. Appliquer une amplification d'amplitude où :
   - La préparation `prep` consiste en la construction de $|\phi_r\rangle := \frac{1}{\sqrt{|S_r^H|}} \sum_{x \in S_r^H} |x, H(x)\rangle$, par amplification d'amplitude sur ce « bon » sous-espace ;
   - La fonction de test `test` est f_L^H :

$$O_{f_l^H}(|x, H(x)\rangle\,|b\rangle) = |x, H(x)\rangle\,\left|b \oplus f_L^H(x)\right\rangle.$$

Cette procédure effectue une recherche de type Grover sur un espace de départ modifié, S_r^H. L'algorithme renvoie un élément $(x, H(x))$ tel que $f_L^H(x) = 1$, soit $\exists (x', H(x')) \in L,\ H(x) = H(x')$: avec probabilité constante, on a $x \neq x'$. Ce test peut également être intégré à f_L^H : il suffit de s'assurer que les entrées x de la liste L appartiennent à un sous-ensemble clairement défini (par exemple avec un préfixe).

Nous avons dans l'algorithme 4 un nouveau degré de liberté : en fonction du choix des paramètres r et t, il est possible de faire baisser le temps de l'oracle $O_{f_l^H}$ tout en augmentant le temps de préparation `prep`. Rien de tout cela n'impacte la quantité de mémoire, toujours polynomiale en n.

Hypothèses simplificatrices. Nous supposons que :

- Les procédures d'amplification `QAA` renvoient exactement une superposition d'états du « bon » sous-espace [4] ;
- $|S_r^H| \approx 2^{n-r}$: la fonction H étant aléatoire, la déviation par rapport à cette moyenne est très faible [5] ;
- Soit $Sol_f := \{x : f_L^H(x) = 1\}$, alors $|Sol_f| \approx 2 \times 2^{t-r}$ (2^{t-r} dans le cas où l'on impose $x \neq x'$) ;
- Nous omettons les facteurs polynomiaux en n et considérons que le temps d'exécution de O_H est 1.

Si toutes ces suppositions sont omises, le temps d'exécution final obtenu sera $2^{\frac{2n}{5}}\,(|O_H|_{RT} + O(n))$ pour une probabilité de succès supérieure à $\frac{99}{100}$.

Analyse. Quatre sous-routines constituent notre algorithme :

4. Nous omettons ainsi l'amplitude résiduelle du « mauvais » sous-espace. Il est possible de gagner en précision au moyen de corrections (cf. [Bra+02]), dont le facteur (polynomial en temps) n'intervient pas multiplicativement dans la complexité finale.

5. Par exemple, les inégalités de Chernoff impliquent qu'avec probabilité $\frac{99}{100}$ on a $\left||S_r^H| - 2^{n-r}\right| \leq 4 \times 2^{\frac{n-r}{2}}$

- La construction de L : un élément de L peut être construit en temps $\sqrt{2^n/2^{n-r}} = 2^{r/2}$ avec une recherche de Grover. La fonction de test est $f(x) := 1$ si $x \in S_r^H$ et $f(x) := 0$ sinon. Cette première étape s'effectue en temps $2^{t-r} \times 2^{r/2} = 2^{t-\frac{r}{2}}$.
- La construction de $|\phi_r\rangle$: nous utilisons une amplification d'amplitude $\mathcal{A} = \texttt{QAA}(\texttt{setup}_{\mathcal{A}}, f_{\mathcal{A}})$ où $\texttt{setup}_{\mathcal{A}}$ construit la superposition $|\phi_0\rangle = \frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} |x\rangle$ avec une requête à O_H et $f_{\mathcal{A}}(x) := 1$ si et seulement si x commence avec r zéros. Au total, $2^{r/2}$ itérations sont nécessaires, chacune effectuée en temps constant.
- La construction de $O_{f_L^H}$: il s'agit d'un oracle d'appartenance tel que décrit en section 4.1.
- La procédure globale : $\mathcal{B} = \texttt{QAA}(\texttt{setup}_{\mathcal{B}} = \mathcal{A}, f_L^H)$ où $\mathcal{A}$ est la sous-routine de préparation construisant l'état $|\phi_r\rangle$ et f_L^H la fonction de test. La probabilité que $x \in S_r^H$ satisfasse $f_L^H(x) = 1$ est $\frac{2\times 2^{t-r}}{2^{n-r}}$, donc l'algorithme $\mathcal{B}$ effectue $2^{\frac{n-t-1}{2}}$ itérations et appels subséquents à $\mathcal{A}$ et $O_{f_L^H}$.

Le temps d'exécution global de $\mathcal{B}$ est donc :

$$2^{\frac{n-t-1}{2}} \left(2^{\frac{r}{2}} + 2^{t-r}\right)$$

La procédure exige par ailleurs de créer la liste L, ce qui donne un total

$$2^{\frac{n-t-1}{2}} \left(2^{\frac{r}{2}} + 2^{t-r}\right) + 2^{t-\frac{r}{2}}.$$

L'optimisation de cette expression impose $t = \frac{3n}{5}$ et $r = \frac{2t}{3} = \frac{2n}{5}$, qui réalise l'équilibre entre le coût d'une préparation interne et le coût d'un appel à $O_{f_L^H}$. Le temps d'exécution est $\widetilde{O}\left(2^{2n/5}\right)$.

Mémoire. Aucun des circuits considérés, ni l'amplification d'amplitude, ni le circuit pour $O_{f_L^H}$, ne requièrent plus de $O(n)$ qubits. La mémoire classique nécessaire au stockage de L représente $2^{t-r} = 2^{\frac{n}{5}}$ éléments de taille n.

Théorème 4.1. *Soit $H : \{0,1\}^n \to \{0,1\}^n$ une fonction aléatoire calculable facilement. Il existe un algorithme quantique en temps $\widetilde{O}\left(2^{\frac{2n}{5}}\right)$, utilisant $\widetilde{O}\left(2^{\frac{n}{5}}\right)$ en mémoire classique et $O(n)$ qubits, qui renvoie une collision de H.*

4.3 Recherche de préimage multi-cibles

Avec une idée similaire, il est possible de résoudre le problème 2.2 en moins de $2^{n/2}$ opérations, quoique l'amélioration soit inférieure. Nous considérons cette fois une permutation aléatoire H, afin d'assurer l'existence de préimages. [6] Nous disposons d'une liste $L' = \{y_1, \ldots, y_{2^t}\}$ et recherchons la préimage de l'un de ces éléments : x tel que $\exists y_i, H(x) = y_i$.

La lecture de la liste complète L' « en ligne » , pour pouvoir former la sous-liste intéressante L, devient un temps 2^t incompressible, ce qui affecte la complexité. Le temps devient :

$$2^{\frac{n-t}{2}} \left(2^{\frac{r}{2}} + 2^{t-r}\right) + 2^t$$

minimisé pour $r = \frac{2t}{3}$ et $t = \frac{3n}{7}$. Deux cas sont à séparer : si $t \leq \frac{3n}{7}$, on prend $r = \frac{2t}{3}$ et on obtient un temps

$$2^{n/2-t/6} + 2^t \leq 2^{n/2-t/6+1}.$$

Si $t \geq \frac{3n}{7}$, nous n'avons pas besoin de tous les éléments de L', $2^{3n/7}$ suffisent et notre algorithme nécessite de l'ordre de $2^{3n/7}$ opérations.

Mémoire. Seuls $O(n)$ qubits sont nécessaires ; en revanche, la mémoire classique est toujours $2^{t-r} = 2^{\frac{t}{3}}$ pour stocker L. Les éléments de L' qui ne nous intéressent pas sont lus, mais pas stockés.

Théorème 4.2. *Soit $H : \{0,1\}^n \to \{0,1\}^n$ une permutation aléatoire. Soit L une liste de 2^t éléments.*

- *Si $t \leq \frac{3n}{7}$, il existe un algorithme quantique en temps $\widetilde{O}\left(2^{n/2-t/6}\right)$, utilisant $O(n)$ qubtis et $\widetilde{O}\left(2^{\frac{t}{3}}\right)$ mémoire classique, qui trouve la préimage d'un élément de L ;*
- *Si $t \geq 2^{\frac{3n}{7}}$, ce temps est $\widetilde{O}\left(2^{\frac{3n}{7}}\right)$ et la mémoire classique nécessaire est $\widetilde{O}\left(2^{\frac{n}{7}}\right)$.*

6. Si H est remplacée par une fonction aléatoire, cela nous ramène au problème de seconde préimage. Les complexités sont inchangées.

Algorithme 5: Algorithme quantique pour la recherche de préimages multi-cibles

Accès est donné à l'oracle O_H pour une permutation aléatoire $H : \{0,1\}^n \to \{0,1\}^n$. La liste $L' = \{y_1, \ldots, y_{2^t}\}$, contenant des valeurs aléatoires, est lue en entrée. L'algorithme renvoie la préimage de l'un des y_i.

Comme précédemment, nous fixons un paramètre r et notons $S_r^H := \{(x, H(x)) : \exists z \in \{0,1\}^{n-r}, H(x) = \underbrace{0 \ldots 0}_{r \text{ times}} || z\}$.

Les étapes de l'algorithme sont :

1. Construire une liste L contenant tous les éléments de L' qui commencent avec r zéros. Nous supposons qu'elle contient bien 2^{t-r} éléments, avec une déviation très faible. Soit $f_L^H(x) := 1$ si $H(x) \in L$ et $f_L^H(x) := 0$ sinon.
2. Appliquer une amplification d'amplitude avec :
   - La préparation `prep` consiste à construire $|\phi_r\rangle := \frac{1}{\sqrt{|S_r^H|}} \sum_{x \in S_r^H} |x\rangle$, par amplification d'amplitude de ce « bon » sous-espace ;
   - La fonction de test `test` est f_L^H :

$$O_{f_l^H}(|x\rangle |b\rangle) = |x\rangle \left|b \oplus f_L^H(x)\right\rangle .$$

Comme précédemment, la recherche de type Grover est effectuée sur le sous-espace S_r^H. L'algorithme renvoie x tel que $f_L^H(x) = 1$ et donc $H(x) \in L$. Les modifications par rapport à l'algorithme 4 concernent en particulier la construction de L et l'information dont on dispose.

4.4 Parallélisation quantique

Comme mentionné en section 2.6, l'algorithme de Grover ne gagne qu'un facteur $\sqrt{2^s}$ lorsque parallélisé sur 2^s registres de $O(n)$ qubits, ce qui limite son applicabilité. Les algorithmes 4 et 5 disposent d'une petite marge de manœuvre dans ce domaine.

Une manière simple semble d'effectuer 2^s instances en parallèle en réduisant le nombre d'itérations de la procédure d'amplification d'amplitude externe. Initialement, $2^{\frac{n-t}{2}}$ itérations sont nécessaires. Pour avoir un bon résultat avec probabilité $O(2^{-s})$ (et donc, au total, $O(1)$), on réduit ce nombre d'itérations à $2^{\frac{n-t-s}{2}}$.

Collisions. Calculer la liste L est également effectué en parallèle et le temps pour cette opération décroît à $2^{t-r/2-s}$. Le temps total devient :

$$2^{\frac{n-t-s}{2}} \left(2^{r/2} + 2^{t-r}\right) + 2^{t-r/2-s}$$

ce qui donne $t = \frac{3n}{5} + \frac{3s}{5}$, $r = \frac{2n}{5} + \frac{2s}{5}$, un exposant $t - r = \frac{n}{5} + \frac{s}{5}$ pour la mémoire classique utilisée et $t - \frac{r}{2} - s = \frac{2n}{5} - \frac{3s}{5}$ pour le temps. Cet exposant $\frac{3s}{5}$, meilleur que $\frac{s}{2}$, provient du partage de L entre les instances parallèles (contrairement aux instances parallèles de l'algorithme de Grover, qui ne partagent aucune information).

Notons que cette parallélisation n'a un sens que tant que $n - t - s \geq 0$ soit $s \leq \frac{n}{4}$. Lorsque $s = \frac{n}{4}$, nous croisons la complexité en temps de l'algorithme de recherche de collisions de [Bea+13].

Préimages multi-cibles. Le traitement de L étant lui aussi effectué en parallèle sur 2^s processeurs (classiques cette fois), le temps de calcul devient :

$$2^{\frac{n-t-s}{2}} \left(2^{r/2} + 2^{t-r}\right) + 2^{t-s}$$

ce qui donne $r = \frac{2}{3}t$ toujours ; la valeur optimale de t est obtenue lorsque $\frac{n}{2} - \frac{t}{6} - \frac{s}{2} = t - s$ i.e $t = \frac{3n}{7} + \frac{3s}{7}$. L'exposant en temps devient $\frac{3n}{7} - \frac{4s}{7}$ et en mémoire classique $\frac{n+s}{7}$.

Théorème 4.3. *Avec 2^s processeurs quantiques :*

- *Si $s \leq \frac{n}{4}$, il est possible de trouver une collision en temps $\widetilde{O}\left(2^{\frac{2n}{5} - \frac{3s}{5}}\right)$ et en mémoire classique $2^{\frac{n}{5} + \frac{s}{5}}$.*
- *Si $t \leq \frac{3n+3s}{7}$, il est possible de trouver une préimage parmi t en temps $\widetilde{O}\left(2^{n/2 - t/6 - s/2}\right)$ et en mémoire classique $\widetilde{O}\left(2^{\frac{t}{3}}\right)$.*

- *Si* $t = \frac{3n+3s}{7}$, *il est possible de trouver une préimage parmi* t *en temps* $\widetilde{O}\left(2^{3n/7-4s/7}\right)$ *et en mémoire classique* $\widetilde{O}\left(2^{\frac{n+s}{7}}\right)$.

4.5 De la théorie à la pratique

Quatre influences potentiellement néfastes doivent être prises en compte pour faire des calculs plus précis :

1. Les constantes cachées : un facteur $\pi/4$ doit ainsi être ajouté dans l'amplification d'amplitude ;
2. Les facteurs logarithmiques dûs au fait que les calculs sont effectués sur des registre à n qubits, par exemple dans l'oracle d'appartenance ;
3. Les erreurs dans l'amplification d'amplitude ;
4. Le coût d'une requête à O_H.

De toutes ces sources de coût supplémentaire, la dernière est en fait largement supérieure à toutes les autres. Ainsi, en considérant le *nombre d'évaluations de* O_H et non le *nombre d'opérations élémentaires*, les complexités obtenues sont précises.

Mieux encore, si nous écrivons 2^c le coût d'une requête à O_H, l'exposant de complexité en temps devient par exemple $\frac{2n}{5} + \frac{4c}{5} + \frac{\ln_2(n)}{5}$ pour la collision : le coût de O_H lui-même est amorti en augmentant la taille de la liste L. Quant au facteur n, il intervient également de façon anodine (si $n = 128$, on gagne un facteur 4 ; 16 si $n = 256$).

Nous donnons dans les tables 3 et 4 quelques valeurs (arrondies à l'entier) de ces exposants, dans des cas usuels, sans tenir compte des facteurs d'erreur. Des calculs existants sur l'implémentation quantique de fonctions cryptographiques, comme [Gra+16] ou [Amy+16], peuvent servir d'indicateur sur le coût d'une implémentation effective de O_H et le nombre de qubits ancillaires nécessaires.

TABLE 3 – Recherche de collisions

n	Espace (qubits)	Mémoire classique	Temps quantique	Produit temps-mém. quant.	Produit temps-mém. class.
128	$O(1)(s = 0)$	26	51	51	64
128	$s = n/6 = 21$	30	39	60	64
256	$O(1)(s = 0)$	51	102	102	128
256	$s = n/6 = 43$	60	77	119	128

TABLE 4 – Recherche de préimages multi-cibles

n	Espace (qubits)	Nombre de cibles	Mémoire classique	Temps quantique	Produit temps-mém. quant.	Produit temps-mém. class.
128	1	55	18	55	55	73
128	$s = n/8 = 16$	62	21	46	62	66
256	1	110	37	110	110	146
256	$s = n/8 = 32$	124	41	91	123	132

Propagation des erreurs. En prenant par exemple la sous-procédure $\mathcal{A}$ construisant l'état $|\phi_r\rangle$, des erreurs proviennent de :

- La déviation de $|S_r^H|$ par rapport à sa valeur moyenne : en réalité, cette déviation est très faible et demeure négligeable, même amplifiée ;
- Le fait que l'état obtenu après amplification dévie légèrement de $|\phi_r\rangle$. Une construction explicitée dans [Bra+02] permet de corriger cette erreur moyennant un petit facteur polynomial.

Ainsi, nous pouvons nous assurer que cette procédure construit un état $|\phi_{output}\rangle$ tel que :

$$|\langle\phi_{output}|\phi_r\rangle| \geq \cos(2^{r/2-n/2} + o(2^{r/2-n/2})).$$

L'erreur se propage dans la procédure $\mathcal{B}$:

- Dans le cas de la recherche de collisions, la préparation $\mathcal{A}$ est répétée $2^{n/5}$ fois ; l'erreur finale est donc $\approx 2^{n/5}2^{r/2-n/2} \ll 1$.
- Pour la recherche de préimage, il y a $2^{2n/7}$ itérations ; l'erreur finale est donc $\approx 2^{2n/7}2^{r/2-n/2} \ll 1$.

4.6 Trouver plusieurs collisions

Le partage de la liste L entre différentes instances de l'algorithme 4 (ou 5) permet également d'accélérer la recherche de *plusieurs* paires. Supposons que nous en voulons 2^c. Le calcul de L est effectué une seule fois désormais, mais la procédure $\mathcal{B}$ est appelée 2^c fois. Les paramètres optimaux deviennent $t = \frac{3n}{5} + \frac{6c}{5}$ et $r = \frac{2n}{5} + \frac{4c}{5}$. Comme les collisions font intervenir un élément d'une liste arbitraire qui n'en contient que 2^{t-r}, pour nous assurer que les 2^c paires en sortie sont bien distinctes, nous devons avoir $t - r \gg c$ soit $c \ll \frac{n}{3}$. Au-delà, c'est c qui contraint la taille de L.

Théorème 4.4 (Plusieurs collisions)**.** *Si $c \ll \frac{n}{3}$, il est possible de calculer 2^c collisions pour H en temps $\widetilde{O}\left(2^{2n/5+4c/5}\right)$, avec $O(n)$ qubits et $2^{n/5+2c/5}$ mémoire classique.*

4.7 Conséquences en cryptographie

Les algorithmes 4 et 5, bien que vraisemblablement sub-optimaux, montrent qu'il est possible d'obtenir un produit temps-mémoire inférieur à $2^{n/2}$, même avec une parallélisation. Nous en tirons quelques conclusions immédiates, en renvoyant à [CNS17] pour plus de détails.

Sécurité des fonctions de hachage. Les algorithmes présentés ci-dessus constituent des attaques quantiques génériques sur les fonctions de hachage, qui font donc mécaniquement décroître le niveau de sécurité espéré de ces primitives. Avec un nombre polynomial de qubits, similaire à celui nécessaire pour implémenter l'algorithme de Grover, nous obtenons :

- Un temps $\widetilde{O}\left(2^{2n/5}\right)$ pour une collision, contre $O(2^{n/2})$ en classique ;
- Un temps $\widetilde{O}\left(2^{3n/7}\right)$ pour la recherche d'une préimage parmi $2^{3n/7}$, contre $O(2^{4n/7})$ en classique.

Utilisateurs multiples. Notre algorithme de recherche de préimages multi-cibles résout également le problème suivant : 2^t utilisateurs chiffrent chacun le même message connu m, de taille k égale à celle de la clé, avec leurs clés privées respectives $k_1, \ldots, k_{2^t}$. Lorsque $2^{3k/7}$ utilisateurs entrent en jeu, l'algorithme 5 améliore la recherche d'une de ces clés de $O(2^{4k/7})$ en temps classique (et $\widetilde{O}(2^{k/2})$ quantique) à $\widetilde{O}(2^{3k/7})$. L'oracle étant $k \to E_k(m)$, le modèle Q1 suffit dans ce cas.

Modes opératoires. Dans un modèle plus fort, il est possible d'attaquer un mode opératoire tel que CBC en moins de $2^{n/2}$ opérations. L'adversaire doit alors disposer d'un accès en superposition à un oracle de chiffrement utilisant la même clé que celle qui a permis de chiffrer les blocs $m_1, \ldots m_{2^t}$. La recherche de préimage s'applique.

Brique de base. Les attaques différentielles et linéaires classiques ont déjà été analysées de manière générique ([Kap+16b]). Mais dans [Kap+16b], la recherche de collisions est considérée comme ayant un coût $2^{n/3}$, ce qui n'est vrai, comme nous l'avons vu plus haut, que lorsqu'une importante mémoire quantique est disponible. Les nouveaux algorithmes présentés ici peuvent intervenir comme brique de base de ces techniques cryptanalytiques, afin de fournir des estimations de complexité plus rigoureuses.

5 Attaques par différentielles impossibles

Notre objectif dans cette partie est de montrer comment des briques de base quantiques, notamment l'algorithme 4, pourraient être intégrées dans des procédures visant à améliorer les attaques *classiques* sur des primitives symétriques. Nous nous sommes intéressés à l'exemple des attaques par différentielles impossibles, qui donnent de très bons résultats de cryptanalyse, principalement en vue des attaques de type *meet-in-the-middle* (MIM) sur le standard AES (voir à ce sujet [Jea13]).

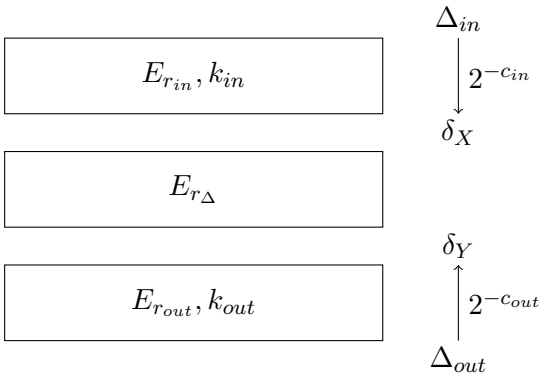

FIGURE 3 – Schéma d'une différentielle impossible

5.1 Principe des attaques par différentielles impossibles

Notre présentation des attaques par différentielles impossibles, introduites par Knudsen ([Knu98]) et Biham, Biryukov et Shamir ([BBS99]), reprend dans les grandes lignes celle faite dans [BNS14]. Nous commençons par les attaques différentielles.

Différentielles. Inventée par Biham et Shamir ([BS90]), la cryptanalyse différentielle étudie la propagation de différences dans une fonction de chiffrement, qui causent une déviation observable vis-à-vis d'une permutation aléatoire. Une différentielle δ_0, δ_1 sera telle que :

$$\mathrm{E_K}(x \oplus \delta_0) = \mathrm{E_K}(x) \oplus \delta_1$$

arrive avec une probabilité supérieure à 2^{-n} (disons 2^{-t}), indépendamment de la clé k utilisée. Supposons qu'une telle différentielle existe sur r_Δ tours d'une fonction de chiffrement et qu'on puisse écrire :

$$\mathrm{E_K} = E_{r_{out}} \circ E_{r_\Delta}$$

en ajoutant un certain nombre de tours à la fin, tels que $E_{r_{out}}$ dépend d'une sous-clé de petite taille k_{out}. Il est possible de monter une attaque des derniers tours (*last-rounds attack*) de la manière suivante : calculer un grand nombre de paires $\mathrm{E_K}(x), \mathrm{E_K}(x \oplus \delta_0)$, puis pour chaque choix possible de k_{out}, vérifier en inversant les r_{out} derniers tours si la différentielle apparaît bien au bout de 2^t tests (en coût classique, cela s'entend).

Dans le modèle Q2, en ayant accès en superposition à $\mathrm{E_K}$, il est possible de mener une telle attaque en temps $2^{k_{out}/2} \times 2^{t/2}$ ([Kap+16b]).

Différentielles impossibles et paramètres d'attaques. Comme leur nom l'indique, les différentielles impossibles n'apparaissent pas : de la structure de la primitive de chiffrement, on déduit une paire δ_X, δ_Y telle qu'une différence $x_1 \oplus x_2 = \delta_X$ ne *peut pas*, après r_Δ tours, produire une différence $E_{r_\Delta}(x_1) \oplus E_{r_\Delta}(x_2) = \delta_Y$. Cette propriété structurelle est obtenue par analyse de la primitive.

Le but étant d'attaquer un maximum de tours, on s'autorise à en ajouter au début et à la fin : $\mathrm{E_K} = E_{r_{out}} \circ E_{r_\Delta} \circ E_{r_{in}}$. Le nombre de bits de clé à deviner pour ces tours supplémentaires est noté $|k_{in} \cup k_{out}|$, car la quantité doit tenir compte des éventuels liens entre k_{in} et k_{out}.

Toujours structurellement, on construit un ensemble Δ_{in} de *différences* en entrée tel qu'une différence dans Δ_{in} produit δ_X après r_{in} tours avec probabilité $2^{-c_{in}}$, et Δ_{out} qui produit δ_Y après r_{out} tours inverses avec probabilité $2^{-c_{out}}$.

Attaque et complexité. Le principe de l'attaque est le suivant (on note C_E le coût d'un chiffrement) :

- On obtient N paires avec une différentielle dans Δ_{in} en entrée et dans Δ_{out} en sortie (coût $C_N C_E$). Souvent, ces paires sont générées grâce au chiffrement de *structures* (des ensembles de textes clairs prenant toutes les valeurs dans Δ_{in}) ;
- Pour chaque choix des $|k_{in} \cup k_{out}|$ bits de clés partielles, on vérifie s'il existe une paire parmi les N qui produit la différentielle impossible δ_X, δ_Y. S'il y en a une, ce choix est éliminé ;
- Lorsque suffisamment de clés ont été éliminées par cette méthode, on complète par recherche exhaustive.

Le coût total de l'attaque, en utilisant des stratégies de *early abort* (on calcule rapidement les clés concernées par chaque paire), est donc :

$$\left(C_N + \left(N + 2^{|k_{in} \cup k_{out}|} \frac{N}{2^{c_{in}+c_{out}}}\right) \alpha + 2^{|K|} \left(1 - 2^{-(c_{in}+c_{out})}\right)^N\right) C_E$$

où αC_E est le coût d'un chiffrement partiel sur les $r_{in} + r_{out}$ tours.

Des calculs supplémentaires peuvent être nécessaires, comme retransformer les bits $|k_{in} \cup k_{out}|$ choisis en véritables bits de clé.

Quant au nombre de chiffrements nécessaires pour trouver les N paires :

$$C_N = \max\left\{\sqrt{N 2^{n+1-|\Delta|}}, N 2^{n+1-|\Delta_{in}|-|\Delta_{out}|}\right\}$$

Où $|\Delta| = \log_2(card(\Delta))$ et $|\Delta| = \max(|\Delta_{in}|, |\Delta_{out}|)$.

5.2 Peut-on améliorer le temps de recherche des paires ?

Notre première remarque est que la recherche des bonnes paires à différences dans $(\Delta_{in}, \Delta_{out})$ peut être améliorée pour un adversaire quantique dans le modèle Q2 (accès en superposition à la fonction de chiffrement), par une recherche de collisions.

Supposons, pour simplifier, que l'adversaire a également accès à $\mathrm{E}_\mathrm{K}^{-1}$. Nous cherchons x, y tels que $x \oplus y \in \Delta_{in}$ et $\mathrm{E}_\mathrm{K}(x) \oplus \mathrm{E}_\mathrm{K}(y) \in \Delta_{out}$. Le plus souvent, cela se traduit par une collision sur $n - |\Delta_{in}|$ bits en position fixées de x, y et sur $n - |\Delta_{out}|$ bits en positions fixées de $\mathrm{E}_\mathrm{K}(x), \mathrm{E}_\mathrm{K}(y)$. Sans perte de généralité, supposons que $|\Delta_{in}| > |\Delta_{out}|$. Il suffira d'échanger les rôles sinon (ainsi que d'échanger les antécédents et les images).

Supposons que $|\Delta_{in}| > n - |\Delta_{out}|$. On fixe déjà $n - |\Delta_{in}|$ bits des entrées et on obtient une fonction $H : \{0,1\}^{\Delta_{in}} \to \{0,1\}^{n-|\Delta_{out}|}$. En fixant de nouveaux bits, aux positions choisies cette fois, on réduit les entrées de H pour obtenir une fonction aléatoire $H' : \{0,1\}^{n-|\Delta_{out}|} \to \{0,1\}^{n-|\Delta_{out}|}$. Toute collision de H' donne une bonne paire x, y.

En choisissant de nouveaux bits de x, y, il est possible de construire une fonction : $H : \{0,1\}^{n-|\Delta_{out}|} \to \{0,1\}^{n-|\Delta_{out}|}$ telle que toute collision $H(x) = H(x')$ donne une bonne paire x, y avec : x et y égaux sur les $n-|\Delta_{in}|$ bits imposant l'appartenance à Δ_{in}, et fixés à une valeur choisie ; x et y égaux sur $|\Delta_{in}|+|\Delta_{out}|-n$ bits supplémentaires choisis.

Malheureusement, le cas inverse se présente le plus souvent, soit $n-|\Delta_{in}| > |\Delta_{out}|$. le problème manque de structure dans ce cas, la recherche quantique de collisions telle que nous envisageons de l'utiliser ne parvient pas à améliorer le temps des procédures classiques basées sur le chiffrement de *structures*.

Remarquons que, toujours si $n - |\Delta_{in}| < |\Delta_{out}|$, nous pouvons obtenir un grand nombre de collisions avec la méthode envisagée plus haut.

Un autre problème intervient alors : pour les quantités de paires de collisions partielles habituellement nécessaires à générer, notre procédure quantique pour des collisions multiples va atteindre le maximum de sa capacité, sans amélioration par rapport à la meilleure procédure classique correspondante (lorsque trop de collisions sont demandées, on retombe sur le chiffrement de *structures* complètes).

Conclusion. L'algorithme de recherche quantique de collisions présenté plus haut ne permet pas, pour les applications pratiques (qui demandent notamment beaucoup de paires), d'améliorer la recherche des paires.

5.3 Attaque générique

Une procédure d'attaque « générique » pour une différentielle impossible existe également, bien que nous n'ayons pas obtenu d'application satisfaisante à ce jour. Elle consiste à effectuer une recherche exhaustive sur les clés partielles $|k_{in} \cup k_{out}|$ avec l'algorithme de Grover. La fonction de test d'une clé partielle, appelée en superposition, vérifie si cette clé fait apparaître la différentielle impossible, à l'aide d'une autre instance de Grover.

En effet, on écrit $\mathrm{E}_\mathrm{K}(x) = E_{r_{out}} \circ E_{r_\Delta} \circ E_{r_{in}}(x)$ soit $E_{r_\Delta}(x) = E_{r_{out}}^{-1} \circ \mathrm{E}_\mathrm{K} \circ E_{r_{in}}^{-1}(x)$. La fonction E_{r_Δ} est celle qui admet la différentielle impossible. Pour tout choix de clé partielle, on cherche donc x tel que $E_{r_\Delta}(x \oplus \delta_X) = E_{r_\Delta}(x) \oplus \delta_Y$. Si c'est la bonne clé, un tel x n'existe pas. Si c'est une mauvaise clé, la fonction appelée est aléatoire et on trouvera un x solution.

Complexité. L'algorithme de Grover « externe » requiert $2^{|k_{in} \cup k_{out}|/2}$ appels à l'algorithme de Grover « interne », lequel cherche une solution parmi 2^n possibles. Au total, l'attaque s'effectue en temps $2^{\frac{|k_{in} \cup k_{out}|+n}{2}} \times (C_E + \alpha C_E)$ où αC_E est le temps requis au calcul des tours additionnels.

En théorie, cette attaque (simple recherche exhaustive avec un distingueur, similaire aux attaques présentées dans [Kap+16b]) peut donc battre l'algorithme de Grover sur les clés $(2^{k/2} C_E)$. Cependant, nous n'en avons pas trouvé d'application probante. Les attaques par différentielles impossibles classiques font un usage intensif de la mémoire classique (ne fût-ce que par le partage des N paires). Cette procédure n'est en aucun cas fidèle à leur état d'esprit. En revanche, n'étant applicable que si la taille de bloc n est inférieure à la taille de clé k, elle permet de rappeler le paradigme (déjà évoqué dans [Kap+16a]) selon lequel une taille d'*état interne* trop faible dessert la sécurité post-quantique des primitives.

6 Conclusion

Collision et recherche de préimages J'ai mis en évidence un algorithme de recherche de collisions reposant sur des principes simples, sub-optimal en nombre de requêtes, mais améliorant le meilleur algorithme quantique connu avec peu de mémoire. Cette procédure générique peut être parallélisée.

Trouver les meilleurs algorithmes quantiques en mémoire polynomiale pour la recherche de collisions ou les éléments distincts, ou éventuellement borner leur complexité en nombre de requêtes, est une question ouverte et d'importance cruciale pour la cryptographie. Par exemple, il n'existe pas à notre connaissance d'algorithme pour les éléments distincts en mémoire polynomiale qui effectue moins de $O(2^n)$ requêtes. Un tel algorithme aurait des conséquences importantes sur la sécurité des chiffrements par blocs itératifs, car il offrirait une amélioration quantique des attaques de type *meet-in-the-middle* ([Kap14]). De notre côté, il est possible que l'exposant $2n/5$ puisse être encore amélioré.

En attaquant les primitives en fonction de la taille de *bloc* et non pas seulement de la taille de *clé*, ces algorithmes invitent aussi à remettre en question la croyance habituelle en un simple « doublement de la taille de clé » : nous pensons que les assertions de sécurité en cryptographie symétrique doivent bénéficier d'une mise à niveau post-quantique systématique et précise, tenant compte à la fois des modèles d'attaquant et de leurs capacités.

Travaux futurs. Ma thèse, qui commencera à l'Inria l'an prochain dans le cadre de l'ERC Quasymodo, offrira la possibilité de creuser plus avant les problèmes ouverts durant ce stage ainsi que les directions abordées :

- Algorithmes quantiques en mémoire polynomiale (éléments distincts, collisions) appliqués au contexte cryptographique. Sous nos contraintes fortes de coût, la borne $2n/5$ peut-elle descendre de nouveau ?
- Applications à des attaques : éponges, attaques basées sur des invariants, chiffrements à flots, réalisabilité d'attaques de type *meet-in-the-middle* ;
- Évaluation précise de la sécurité de chiffrements courants tel que l'AES, notamment envers des attaques de type *meet-in-the-middle* ([Jea13]) adaptées en procédures quantiques ;
- Design d'un chiffrement par bloc de 256 bits, offrant de bien meilleures garanties de sécurité face aux attaques quantiques basées sur des collisions ou des préimages multi-cibles.

Appendice A : Index

Appendice B : Références

Cryptographie classique

[And+08] Elena Andreeva, Charles Bouillaguet, Pierre-Alain Fouque, Jonathan J. Hoch, John Kelsey, Adi Shamir et Sébastien Zimmer. "Second Preimage Attacks on Dithered Hash Functions". In : *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*. T. 4965. Lecture Notes in Computer Science. Springer, 2008, p. 270–288.

[BBS99] E. Biham, A. Biryukov et A. Shamir. "Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials". In : *EUROCRYPT 1999*. T. 1592. LNCS. Springer, 1999, p. 12–23.

[Bel+97] Mihir Bellare, Anand Desai, E. Jokipii et Phillip Rogaway. "A Concrete Security Treatment of Symmetric Encryption". In : *FOCS*. IEEE Computer Society, 1997, p. 394–403.

[Bih02] Eli Biham. "How to decrypt or even substitute DES-encrypted messages in 2^{28} steps". In : *Inf. Process. Lett.* 84.3 (2002), p. 117–124.

[BL16] Karthikeyan Bhargavan et Gaëtan Leurent. "On the Practical (In-)Security of 64-bit Block Ciphers : Collision Attacks on HTTP over TLS and OpenVPN". In : *IACR Cryptology ePrint Archive* 2016 (2016), p. 798.

[BMS06] Alex Biryukov, Sourav Mukhopadhyay et Palash Sarkar. "Improved Time-Memory Trade-Offs with Multiple Data". In : *Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers*. T. 3897. Lecture Notes in Computer Science. Springer, 2006, p. 110–127.

[BNS14] Christina Boura, María Naya-Plasencia et Valentin Suder. "Scrutinizing and Improving Impossible Differential Attacks : Applications to CLEFIA, Camellia, LBlock and Simon". In : *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*. Sous la dir. de Palash Sarkar et Tetsu Iwata. T. 8873. Lecture Notes in Computer Science. Springer, 2014, p. 179–199.

[BS90] Eli Biham et Adi Shamir. "Differential Cryptanalysis of DES-like Cryptosystems". In : *CRYPTO*. T. 537. Lecture Notes in Computer Science. Springer, 1990, p. 2–21.

[CMS12] Sanjit Chatterjee, Alfred Menezes et Palash Sarkar. "Another Look at Tightness". In : *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*. T. 7118. Lecture Notes in Computer Science. Springer, 2012, p. 293–319.

[Jea13] Jérémy Jean. "Cryptanalysis of Symmetric-Key Primitives Based on the AES Block Cipher. (Cryptanalyse de primitives symétriques basées sur le chiffrement AES)". Thèse de doct. École Normale Supérieure, Paris, France, 2013.

[Knu94] Lars R. Knudsen. "Truncated and Higher Order Differentials". In : *Fast Software Encryption : Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*. T. 1008. Lecture Notes in Computer Science. Springer, 1994, p. 196–211.

[Knu98] L. R. Knudsen. *DEAL – A 128-bit cipher*. Technical Report, Department of Informatics, University of Bergen, Norway. 1998.

[Men12] Alfred Menezes. "Another Look at Provable Security". In : *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*. T. 7237. Lecture Notes in Computer Science. Springer, 2012, p. 8.

[OW94] Paul C. van Oorschot et Michael J. Wiener. "Parallel Collision Search with Application to Hash Functions and Discrete Logarithms". In : *CCS '94, Proceedings of the 2nd ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 2-4, 1994*. ACM, 1994, p. 210–218.

[Pol75] J. M. Pollard. "A monte carlo method for factorization". In : *BIT Numerical Mathematics* 15.3 (1975), p. 331–334.

Calcul quantique

[Amb05] Andris Ambainis. "Polynomial Degree and Lower Bounds in Quantum Complexity : Collision and Element Distinctness with Small Range". In : *Theory of Computing* 1.1 (2005), p. 37–46.

[Amb07] Andris Ambainis. "Quantum Walk Algorithm for Element Distinctness". In : *SIAM J. Comput.* 37.1 (2007), p. 210–239.

[AS04] Scott Aaronson et Yaoyun Shi. "Quantum lower bounds for the collision and the element distinctness problems". In : *J. ACM* 51.4 (2004), p. 595–605.

[Bea+13] Robert Beals, Stephen Brierley, Oliver Gray, Aram W Harrow, Samuel Kutin, Noah Linden, Dan Shepherd et Mark Stather. "Efficient distributed quantum computing". In : *Proc. R. Soc. A*. T. 469. 2153. The Royal Society. 2013, p. 20120686.

[BHT98] Gilles Brassard, Peter Høyer et Alain Tapp. "Quantum Cryptanalysis of Hash and Claw-Free Functions". In : *LATIN*. T. 1380. Lecture Notes in Computer Science. Springer, 1998, p. 163–169.

[Bra+02] Gilles Brassard, Peter Hoyer, Michele Mosca et Alain Tapp. "Quantum amplitude amplification and estimation". In : *Contemporary Mathematics* 305 (2002), p. 53–74.

[Bra+11] Gilles Brassard, Peter Høyer, Kassem Kalach, Marc Kaplan, Sophie Laplante et Louis Salvail. "Merkle puzzles in a quantum world". In : *Advances in Cryptology–CRYPTO 2011*. Springer, 2011, p. 391–410.

[DEL00] David Deutsch, Artur Ekert et Rossella Lupacchini. "Machines, logic and quantum physics". In : *Bulletin of Symbolic Logic* 3.3 (2000), p. 265–283.

[DW13] Ronald De Wolf. *Quantum Computing : Lecture Notes*. 2013.

[Fey82] Richard P Feynman. "Simulating physics with computers". In : *International journal of theoretical physics* 21.6 (1982), p. 467–488.

[GR04] Lov K. Grover et Terry Rudolph. "How significant are the known collision and element distinctness quantum algorithms ?" In : *Quantum Information & Computation* 4.3 (2004), p. 201–206.

[Gro02] Lov K Grover. "Trade-offs in the quantum search algorithm". In : *Physical Review A* 66.5 (2002), p. 052314.

[Gro96] Lov K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In : *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*. Sous la dir. de Gary L. Miller. ACM, 1996, p. 212–219.

[Gro98] Lov K Grover. "How fast can a quantum computer search ?" In : *arXiv preprint quant-ph/9809029* (1998).

[Kut05] Samuel Kutin. "Quantum Lower Bound for the Collision Problem with Small Range". In : *Theory of Computing* 1.2 (2005), p. 29–36.

[NC02] Michael A Nielsen et Isaac Chuang. *Quantum computation and quantum information*. 2002.

[Sho94] Peter W. Shor. "Algorithms for Quantum Computation : Discrete Logarithms and Factoring". In : *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*. IEEE Computer Society, 1994, p. 124–134.

[Unr15] Dominique Unruh. "Non-interactive zero-knowledge proofs in the quantum random oracle model". In : *Eurocrypt 2015*. T. 9057. Preprint on IACR ePrint 2014/587. Springer, 2015, p. 755–784.

[Zal99] Christof Zalka. "Grover's quantum searching algorithm is optimal". In : *Physical Review A* 60.4 (1999), p. 2746.

[Zha15a] Mark Zhandry. "A Note on the Quantum Collision and Set Equality Problems". In : *Quantum Info. Comput.* 15.7-8 (mai 2015), p. 557–567.

[Zha15b] Mark Zhandry. "Secure identity-based encryption in the quantum random oracle model". In : *International Journal of Quantum Information* 13.04 (2015), p. 1550014.

Cryptographie post-quantique et cryptanalyse quantique

[Amy+16] Matthew Amy, Olivia Di Matteo, Vlad Gheorghiu, Michele Mosca, Alex Parent et John M. Schanck. "Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3". In : *IACR Cryptology ePrint Archive* 2016 (2016), p. 992.

[Ana+16] Mayuresh Vivekanand Anand, Ehsan Ebrahimi Targhi, Gelo Noel Tabia et Dominique Unruh. "Post-Quantum Security of the CBC, CFB, OFB, CTR, and XTS Modes of Operation". In : *PQCrypto*. T. 9606. Lecture Notes in Computer Science. Springer, 2016, p. 44–63.

[BB17] Gustavo Banegas et Daniel J. Bernstein. "Low-communication parallel quantum multi-target preimage search". In : *SAC 2017* (2017).

[BBD09] Daniel J Bernstein, Johannes Buchmann et Erik Dahmen. *Post-quantum cryptography*. Springer Science & Business Media, 2009.

[Ber09] Daniel J Bernstein. "Cost analysis of hash collisions : Will quantum computers make SHARCS obsolete ?" In : *SHARCS'09 Special-purpose Hardware for Attacking Cryptographic Systems* (2009), p. 105.

[BNP17] Xavier Bonnetain et María Naya-Plasencia. *On Concrete Quantum Security of Symmetric Primitives with Modular Additions*. Communication personnelle. 2017.

[Bon+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner et Mark Zhandry. "Random Oracles in a Quantum World". In : *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*. Sous la dir. de Dong Hoon Lee et Xiaoyun Wang. T. 7073. Lecture Notes in Computer Science. Springer, 2011, p. 41–69.

[BZ13] Dan Boneh et Mark Zhandry. "Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World". In : *CRYPTO (2)*. T. 8043. Lecture Notes in Computer Science. Springer, 2013, p. 361–379.

[CNS17] André Chailloux, María Naya-Plasencia et André Schrottenloher. "An efficient quantum collision search algorithm and implications on symmetric cryptography". In : ASIACRYPT 2017, à paraître (2017).

[Dam+13] Ivan Damgård, Jakob Funder, Jesper Buus Nielsen et Louis Salvail. "Superposition Attacks on Cryptographic Protocols". In : *Information Theoretic Security - 7th International Conference, ICITS 2013, Singapore, November 28-30, 2013, Proceedings*. Sous la dir. de Carles Padró. T. 8317. Lecture Notes in Computer Science. Springer, 2013, p. 142–161.

[GHS16] Tommaso Gagliardoni, Andreas Hülsing et Christian Schaffner. "Semantic security and indistinguishability in the quantum world". In : *Annual Cryptology Conference*. Springer. 2016, p. 60–89.

[Gra+16] Markus Grassl, Brandon Langenberg, Martin Roetteler et Rainer Steinwandt. "Applying Grover's Algorithm to AES : Quantum Resource Estimates". In : *PQCrypto*. T. 9606. Lecture Notes in Computer Science. Springer, 2016, p. 29–43.

[Kap+16a] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier et María Naya-Plasencia. "Breaking Symmetric Cryptosystems Using Quantum Period Finding". In : *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*. Sous la dir. de Matthew Robshaw et Jonathan Katz. T. 9815. Lecture Notes in Computer Science. Springer, 2016, p. 207–237.

[Kap+16b] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier et María Naya-Plasencia. "Quantum Differential and Linear Cryptanalysis". In : *IACR Trans. Symmetric Cryptol.* 2016.1 (2016), p. 71–94.

[Kap14] Marc Kaplan. "Quantum attacks against iterated block ciphers". In : *CoRR* abs/1410.1434 (2014).

[KM10] Hidenori KUWAKADO et Masakatu MORII. "Quantum distinguisher between the 3-round Feistel cipher and the random permutation". In : *IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings*. IEEE, 2010, p. 2682–2685.

[KM12] Hidenori KUWAKADO et Masakatu MORII. "Security on the quantum-type Even-Mansour cipher". In : *Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2012, Honolulu, HI, USA, October 28-31, 2012*. IEEE, 2012, p. 312–316.

[LL17] Fanbao LIU et Fengmei LIU. *Universal Forgery and Key Recovery Attacks : Application to FKS, FKD and Keyak*. Cryptology ePrint Archive, Report 2017/691. `http://eprint.iacr.org/2017/691`. 2017.

[TU17] Ehsan Ebrahimi TARGHI et Dominique UNRUH. "Quantum Collision-Resistance of Non-uniformly Distributed Functions : Upper and Lower Bounds". In : *IACR Cryptology ePrint Archive* 2017 (2017), p. 575.

[Zha12] Mark ZHANDRY. "How to Construct Quantum Random Functions". In : *FOCS*. IEEE Computer Society, 2012, p. 679–687.