# Protocoles cryptographiques post-quantiques de préservation de l'anonymat et du secret des communications

Hugo Senet

PSL
UNIVERSITÉ PARIS

**THÈSE DE DOCTORAT**
**DE L'UNIVERSITÉ PSL**

Préparée à l'École normale supérieure
et à Thales

# Protocoles cryptographiques post-quantiques de préservation de l'anonymat et du secret des communications


Soutenue par
**Hugo Senet**
le 22 septembre 2023


École doctorale n°386
**Sciences mathématiques de Paris-Centre**

Spécialité
**Informatique**

Composition du jury :

| | |
|---|---|
| Olivier BLAZY<br>École polytechnique | *Président* |
| Dương Hiệu PHAN<br>Télécom Paris | *Rapporteur* |
| Adeline ROUX-LANGLOIS<br>CNRS | *Rapporteur* |
| David POINTCHEVAL<br>École normale supérieure, CNRS | *Examinateur* |
| Damien VERGNAUD<br>Sorbonne Université | *Examinateur* |
| Céline CHEVALIER<br>École normale supérieure<br>Université Paris-Panthéon-Assas | *Directrice de thèse* |
| Thomas RICOSSET<br>Thales | *Directeur de thèse en entreprise* |

ENS
ÉCOLE NORMALE SUPÉRIEURE

THALES

# Remerciements

À tous seigneurs tous honneurs : je tiens à commencer ces quelques lignes de remerciements en exprimant la gratitude que j'ai pour Céline Chevalier et Thomas Ricosset, qui furent pendant plus de trois ans mes directeurs de thèse.

Céline Chevalier, avec la bienveillance qu'elle a constamment manifestée à mon égard, s'est toujours empressée de m'assurer de sa disponibilité pour répondre à la moindre de mes questions et j'ai pu ainsi profiter tant de ses conseils scientifiques éclairés que de son aide pour affronter les problèmes pratiques ou administratifs qui se sont présentés. Pour cela et pour tout le reste, je lui adresse mes plus chaleureux remerciements.

Un grand merci également à Thomas Ricosset qui, par sa conduite avisée de mes travaux et ses remarques judicieuses sur ceux-ci, a fortement contribué à faire de la préparation de cette thèse l'expérience, des plus formatrices, qu'elle a été.

Je ne veux bien sûr pas manquer de remercier vivement les membres de mon jury de thèse pour avoir accepté d'en faire partie et de prendre de leur temps pour remplir comme ils l'ont fait le rôle de juré avec une diligence qui s'est reflétée dans la qualité des questions qui m'ont été posées lors de ma soutenance. Il s'agit, outre de mes directeurs de thèse, d'Olivier Blazy, de Dương Hiệu Phan, d'Adeline Roux-Langlois, de Damien Vergnaud et du grand manitou du département d'informatique de l'École normale supérieure, David Pointcheval, dont l'assistance et les instructions m'ont été plus que précieuses au cours de ces dernières années dans le laboratoire qu'il dirige.

Je suis tout particulièrement reconnaissant aux rapporteurs, Adeline Roux-Langlois et Dương Hiệu Phan, qui par ma faute ont dû s'atteler à examiner mon mémoire en un temps restreint en plein milieu de l'été alors qu'à n'en pas douter ils se rêvaient à ce moment sur quelque plage paradisiaque d'une île du Pacifique... Je leur sais gré de l'effort auquel ils ont consenti ; je souhaiterais qu'ils sachent que j'ai apprécié le soin manifeste qu'ils ont mis dans la lecture critique de mon manuscrit et la rédaction des comptes rendus.

Merci aux membres du laboratoire de l'École, actuels et anciens, étudiants et chercheurs, dont j'ai pu croiser la route ou avec lesquels j'ai pu travailler pour le très bon accueil qu'ils m'ont fait dans l'équipe et pour tous les moments privilégiés d'échange intellectuel qui ont contribué à rendre agréable mes recherches en leur compagnie. Qu'on me pardonne de n'en citer que quelques uns : Michel Abdalla, Léonard Assouline, Hugo Beguinet, Wissam Ghantous, Brice Minaud, Ky Nguyen (Nguyễn Ngọc Kỷ), Paola de Perthuis, Guillaume Renaut, Mélissa Rossi, Éric Sageloli, Robert Schädlich

et Quoc-Huy Vu.

Je remercie aussi, pour les mêmes raisons, mes collègues de Thales à Gennevilliers, que furent, entre autres, Zoé Amblard, Loïc Demange, Éric Garrido, Matthieu Giraud, David Lefranc, Lucie Mousson et Philippe Painchault.

Pour la « préservation de l'anonymat » de mes proches, je ne vais pas les citer ici mais je pense bien sûr à eux avec gratitude car, dans cette aventure qu'a été ma formation doctorale, il est arrivé qu'en plus de m'offrir leur soutien, ils m'aient offert leur secours.

À présent, cher lecteur, si vous vous trouvez au nombre de ceux à qui j'ai coupablement omis de témoigner de ma reconnaissance, je n'aurai qu'un mot à dire pour ma défense : merci !

# Table des matières

Je ne pourrais vous dire, les amis,
même si je le voulais, si je suis ou
non un espion français, car je suis
tenu au plus absolu secret.

*Là où il y a de l'homme...* –
Hervé Le Corneur

# 1 Introduction

Tant pour ce qui est de la préservation du secret des informations confidentielles que de l'authentification des entités numériques ou de la vérification de l'intégrité des données, les systèmes cryptographiques apportent une sécurité indispensable à l'existence même du monde numérique tel que nous le connaissons. Ces systèmes cryptographiques sont fondés sur des protocoles, qui sont des ensembles de règles qui déterminent la manière dont les divers agents d'un système doivent fonctionner et interagir. Le possible essor des calculateurs quantiques est de nature à mettre à mal la sécurité de systèmes cryptographiques qui sont parmi les plus utilisés actuellement. Les protocoles se doivent donc d'évoluer en protocoles dits « post-quantiques » pour offrir aux systèmes de demain la meilleure résistance possible face à cette menace.

## 1.1 La cryptographie

Dans son *Dictionnaire de la langue française*, Émile Littré définissait la cryptographie comme l'« Art d'écrire en caractères secrets qui sont ou de convention ou le résultat d'une transposition des lettres de l'alphabet[1] ». Il est vrai qu'en général, en cette seconde moitié du XIX^e^ siècle, pour rendre un message inintelligible au cas où celui-ci, confidentiel, viendrait à être intercepté, on écrivait un caractère pour un autre, ou l'on remplaçait une suite de lettres par une autre (par exemple, selon une substitution dite polyalphabétique) ; la cryptographie était alors, pour ainsi dire, balbutiante. Pourtant, à cette époque déjà, dans ses *Recherches arithmétiques* [Gau01], Carl Friedrich Gauss avait jeté les bases de la théorie des nombres moderne, et Évariste Galois, dans son fameux mémoire — publié après sa mort par le mathématicien Joseph Liouville [Gal46] —, celles de la théorie qui portera son nom.

La constatation de l'insuffisante valeur de la cryptographie du XIX^e^ siècle se retrouve dans l'article en deux parties d'Auguste Kerckhoffs, paru en 1883, intitulé *La cryptographie militaire* [Ker83a] [Ker83b], où le cryptographe s'étonne de voir savants et professeurs « enseigner et recommander pour les usages de la guerre des systèmes dont un déchiffreur tant soit peu expérimenté trouverait certainement la clef en moins d'une heure de temps » ; l'auteur ne voit guère d'autre explication à cet « excès de confiance dans certains chiffres » que « l'abandon dans lequel la suppression des cabinets noirs et la sécurité des relations postales ont fait tomber les études cryptographiques ».

Ce n'est qu'au siècle suivant, par l'effet, notamment, des grands conflits qui l'ont déchiré, qu'on vit la cryptographie tirer réellement parti des outils de la mathématique moderne et muer en une science complexe, si bien que dans les années 1970 apparut une nouvelle sorte de cryptographie, qu'on dit asymétrique par opposition à la cryptographie symétrique, son pendant plus ancien.

Jusqu'alors, en effet, pour établir une communication chiffrée, il fallait que les correspondants convinssent au préalable d'une règle secrète de chiffrement fixant notamment les caractéristiques à connaître du système pour réaliser la transformation du texte d'origine, le *clair*. Avec cette configuration, celle de la cryptographie symétrique, tout correspondant était en mesure de chiffrer, *à la place* d'un autre, un clair donné, et tout chiffré produit par l'un des correspondants pouvait être déchiffré naturellement par un autre, sans même qu'il ait été convenu d'une méthode de déchiffrement, mais simplement par l'application d'un procédé inverse à celui employé pour chiffrer. Par exemple, deux personnes pouvaient convenir qu'à chacune des lettres d'un clair qu'ils souhaiteraient transmettre serait substituée une autre lettre de l'alphabet selon une table de correspondance définie à l'avance : qu'ainsi le A se verrait, en chacune de ses occurrences, remplacer par un G, que le B serait remplacé par un A, le C par un M, et ainsi de suite, le mot « BAC » devenant alors ici « AGM », suite de lettres que ces deux personnes seraient en mesure non seulement de produire, mais aussi de déchiffrer aisément en appliquant la table de substitution dans le sens inverse. La correspondance

[1] *Dictionnaire de la langue française*, d'Émile Littré, édition de 1873, tome premier, page 922, entrée « cryptographie ».

entre les lettres du clair et du chiffré est dans cet exemple l'information qui doit rester secrète pour que le chiffrement le reste aussi.

Cette information secrète — il en faut une — sur laquelle se fonde le chiffrement peut être de deux natures : soit elle correspond au système de chiffrement lui-même, et la sécurité de la communication repose alors sur la méconnaissance par l'adversaire du système employé, soit elle se réduit à un petit ensemble de paramètres du système, appelé *clef*, et la connaissance de la méthode générale de chiffrement employée est alors supposée ne pas compromettre le système. De ces deux approches du chiffrement, la première a fini par être largement rejetée par les cryptographes. L'article de Kerckhoffs exprimait déjà, au deuxième chef d'une liste de six « *desiderata* de la cryptographie militaire », la nécessité que le système « n'exige pas le secret et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi » (ici, le système est ce qui correspondrait, à l'ère numérique, à l'algorithme de chiffrement). En effet, l'auteur explique qu'« il n'est pas nécessaire de se créer des fantômes imaginaires et de mettre en suspicion l'incorruptibilité des employés ou agents subalternes, pour comprendre que, si un système exigeant le secret se trouvait entre les mains d'un trop grand nombre d'individus, il pourrait être compromis à chaque engagement auquel l'un ou l'autre d'entre eux prendrait part ». Ce *desideratum* de la cryptographie militaire est ce qui est maintenant connu sous le nom de *principe de Kerckhoffs* ; il s'applique tout aussi bien en dehors du domaine militaire. À l'époque actuelle, en outre, le fait qu'un système de chiffrement soit connaissable du monde entier — donc largement susceptible d'être étudié et mis à l'épreuve par les spécialistes — et qu'aucune faille critique ne se fasse connaître malgré cela, tend à être perçu comme un gage de sa bonne qualité.

Prenons en exemple le système AES[2], qui est le système de chiffrement symétrique par bloc (c'est-à-dire traitant les données à chiffrer bloc par bloc) recommandé par l'ANSSI[3]. Ce système résulte d'un concours public du NIST[4] dont l'ambition affichée était de choisir un algorithme de chiffrement, dans plusieurs déclinaisons déterminées précisément pour que le système fût à la fois robuste et efficace, d'en faire une norme dont les spécifications[5] fussent accessibles à tous, et d'en permettre un usage non dissimulé comme celui qu'en fait par exemple l'environnement de messagerie Signal[6], dont le code-source est ouvert. De ce concours du NIST est sorti gagnant l'algorithme Rijndael, du nom de ses deux concepteurs Joan Daemen et Vincent Rijmen, dans trois déclinaisons spécifiques correspondant à trois tailles de clef différentes : 128, 192 et 256 bits. Le libre accès aux spécifications du système AES a permis la réalisation d'analyses précises de celui-ci par le monde de la recherche et la publication de méthodes d'attaques qui, bien que de nature à le fragiliser un peu en certains aspects, ne se sont pas montrées suffisamment puissantes pour le rendre caduc.

Notons cependant qu'il apparaît, au nombre des révélations dont fut à l'origine Ed-

[2] Le sigle AES correspond à « *advanced encryption standard* », littéralement « norme de chiffrement avancé ».

[3] L'ANSSI est l'Agence nationale de la sécurité des systèmes d'information, en France.

[4] Le NIST est l'Institut national des normes et de la technologie des États-Unis.

[5] https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf

[6] https://signal.org/docs/specifications/doubleratchet/doubleratchet.pdf

ward Snowden, que la NSA[7], tout en recommandant l'utilisation d'AES, s'est employée à essayer de trouver des attaques sur ce système[8] ; il ne semble pas déraisonnable de penser qu'un tel organisme de renseignement pourrait garder pour lui toute trouvaille offensive déterminante dans un système de chiffrement à spécifications publiques.

Pour que des systèmes informatiques, directement associés à des êtres humains ou non, puissent employer un système de chiffrement symétrique tel qu'AES, il est nécessaire comme nous l'avons dit, qu'ils aient en commun une clef. Or, si ces systèmes sont distants et sont supposés communiquer de façon sécurisée pour la première fois, comment faire en sorte qu'ils puissent convenir d'une clef qui doit être secrète et le rester suffisamment longtemps ?

C'est ce que permet la cryptographie *asymétrique*. Par l'emploi d'un chiffrement dit *asymétrique*, deux individus ou davantage peuvent établir une communication secrète sans pourtant détenir une clef commune. Il leur faut pour cela être chacun pourvus de deux clefs propres, l'une publique, connaissable des autres correspondants, l'autre privée, secrète. Imaginons que l'un de ces individus, qu'on appellera Alice, entreprenne d'envoyer un message chiffré à un autre individu qu'on appellera Benoît. C'est la clef publique de Benoît qui doit être employée par Alice pour chiffrer le message qu'elle lui destine, mais c'est la clef privée de Benoît, qu'il est normalement seul à posséder, que celui-ci doit utiliser pour déchiffrer le message qui lui est adressé.

La sécurité de ces systèmes asymétriques repose sur la difficulté de résoudre certains problèmes mathématiques. Or ces problèmes, comme celui du logarithme discret ou celui de la factorisation de grands entiers, pourraient ne plus être suffisamment difficiles pour qui aurait à sa disposition la puissance de calculateurs quantiques.

Afin de parer à la menace que ferait peser l'essor de tels calculateurs sur ces systèmes, d'importants efforts sont faits pour que voient le jour et se développent des systèmes dits « post-quantiques » de chiffrement, d'authentification et d'identification, analogues aux systèmes classiques mais qui résisteraient à des attaques rendues possibles par ces calculateurs, y compris si ces derniers venaient à devenir particulièrement puissants.

Parmi les constructions qui se développent en tant que systèmes cryptographiques post-quantiques, celles qui sont fondées sur les problèmes difficiles relatifs aux réseaux euclidiens s'imposent comme étant parmi les plus efficaces. C'est le sujet de la section suivante.

[7] *National Security Agency*, « Agence nationale de la sécurité » aux États-Unis.

[8] https://www.spiegel.de/international/germany/inside-the-nsa-s-war-on-internet-security-a-1010361.html

## 1.2 La cryptographie fondée sur les réseaux euclidiens

Un *réseau euclidien* $\mathcal{L} \subset \mathbb{R}^n$ est, pour un entier naturel $n$, un sous-groupe discret de l'espace vectoriel $\mathbb{R}^n$ pour lequel il existe une base $(\mathbf{b}_i)_{i\in[\![1,n]\!]}$ de $\mathbb{R}^n$ telle que

$$\mathcal{L} = \left\{ \sum_{i=1}^{n} x_i \, \mathbf{b}_i : x_i \in \mathbb{Z} \right\}.$$

### 1.2.1 Production d'instances difficiles de problèmes sur les réseaux

Dans un article publié en 1996 [Ajt96], Miklós Ajtai présente la réduction de trois problèmes, considérés difficiles *dans le pire des cas*, au *cas moyen* d'un problème qu'il définit à partir d'une classe particulière de réseaux. En d'autres termes, à partir de la difficulté supposée de trois problèmes dans les instances les plus défavorables à leur résolution — sans que ces instances spécifiques ne soient précisées — Ajtai démontre la difficulté d'un problème dont l'instance correspond à un réseau tiré aléatoirement dans la classe qu'il spécifie.

Par exemple, le premier des trois problèmes supposés difficiles, consiste à trouver, approximativement, la plus petite norme parmi celles des vecteurs non nuls d'un réseau.

Les réseaux euclidiens de la classe sont définis comme les suites finies de longueur $m$, où $m$ est un entier qui ne dépend que de $n$.

Ajtai démontre que si un algorithme probabiliste qui opère en temps polynomial trouve, avec une probabilité de $1/2$ au moins, un vecteur court dans un réseau tiré aléatoirement dans cette classe, il existe un autre algorithme en temps polynomial qui résout les trois problèmes difficiles avec une probabilité qui tend exponentiellement vers 1. En démontrant ainsi la difficulté du problème du plus court vecteur, et des deux autres problèmes, avec des instances aisément engendrables, Ajtai a ouvert la voie à la création de primitives cryptographiques reposant sur des problèmes attachés aux réseaux euclidiens.

### 1.2.2 Le problème du vecteur le plus proche

Le problème du vecteur le plus proche (ou CVP pour *closest vector problem*) est un problème calculatoire fondamental associé aux réseaux euclidiens. Étant donné un réseau $\mathcal{L}$ dans $\mathbb{R}^n$ généré par une base $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ et un point quelconque $\mathbf{t}$ dans $\mathbb{R}^n$, le problème CVP demande de trouver le vecteur dans $\mathcal{L}$ qui est le plus proche de $\mathbf{t}$ en termes de distance euclidienne. Formellement, il s'agit de chercher un vecteur $\mathbf{v}$ dans $\mathcal{L}$ tel que $\|\mathbf{v} - \mathbf{t}\| = \min_{\mathbf{w}\in\mathcal{L}} \|\mathbf{w} - \mathbf{t}\|$.

Ce problème est connu pour être NP-difficile. En outre, il a été démontré que la version approximative du problème, où l'on cherche un vecteur qui est presque le plus proche de $\mathbf{t}$, est également difficile pour certaines approximations.

### 1.2.3 Le problème du plus court vecteur

Le problème du plus court vecteur (ou SVP pour *Shortest Vector Problem*) dans un réseau euclidien est un autre problème calculatoire supposé difficile. Étant donné un réseau $\mathcal{L}$ dans $\mathbb{R}^n$ généré par une base $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$, ce problème demande de trouver le vecteur non nul le plus court dans $\mathcal{L}$. Formellement, il s'agit de chercher un vecteur $\mathbf{v}$ dans $\mathcal{L}$ tel que $\|\mathbf{v}\| = \min_{\mathbf{w}\in\mathcal{L}\setminus\{\mathbf{0}\}} \|\mathbf{w}\|$. Ce problème est NP-difficile, et même sa version approximative est difficile dans certains cas.

### 1.2.4 Le problème de l'apprentissage avec erreurs

Le problème de l'*apprentissage avec erreurs*, communément appelé problème LWE (de l'anglais *Learning With Errors*) est un problème calculatoire supposé difficile introduit en 2005 par Oded Regev [Reg05], ce qui lui a valu d'obtenir le prix Gödel en 2018.

Soit $q$ un entier naturel et $\chi$ une distribution de probabilité sur l'ensemble des entiers. Le problème de décision LWE consiste à distinguer entre deux types de paires $(\mathbf{a}, b)$, où $\mathbf{a}$ est un vecteur à $n$ composantes tiré aléatoirement, selon la distribution uniforme, dans $\mathbb{Z}_q^n$ (c'est-à-dire dans les vecteurs à $n$ composantes d'entiers relatifs modulo $q$) et $b$ est un scalaire dans $\mathbb{Z}_q$. Dans le premier type de paire, $b$ est égal à $\langle \mathbf{a}, \mathbf{s}\rangle + e \mod q$ pour un certain vecteur secret $\mathbf{s}$ dans $\mathbb{Z}_q^n$ et une erreur $e$ tiré selon la distribution $\chi$. Dans le deuxième type de paire, $b$ est tiré selon la distribution uniforme sur $\mathbb{Z}_q$.

Les réseaux euclidiens sont un élément clef dans la preuve de la sécurité du problème LWE. En particulier, Regev a montré que si certaines variantes du problème du plus court vecteur (SVP) et du problème du plus proche vecteur (CVP) dans les réseaux euclidiens sont difficiles à résoudre, alors le problème LWE est aussi difficile à résoudre.

Cela a été démontré en procédant à une réduction quantique de ces problèmes sur les réseaux euclidiens vers le problème LWE. En d'autres termes, si nous avions un algorithme efficace pour résoudre le problème LWE, alors nous pourrions utiliser cet algorithme pour résoudre ces problèmes sur les réseaux en temps polynomial, en supposant que nous ayons accès à un ordinateur quantique.

Le problème LWE a trouvé de nombreuses applications en cryptographie, notamment dans la construction de primitives pour certains systèmes post-quantiques de chiffrement, de signatures numériques ou d'échange de clefs. Parmi ces primitives, on peut citer les preuves à divulgation nulle de connaissance, très importantes dans les schémas de signatures de groupe et d'accréditations anonymes sur lesquels j'ai été amené à travailler. Ces preuves sont le sujet de la section suivante.

## 1.3 Preuves à divulgation nulle de connaissance

Le principe des preuves à divulgation nulle de connaissance, que j'appelle aussi *preuves opaques* repose sur l'interaction entre un prouveur et un vérificateur. Le prouveur souhaite convaincre le vérificateur de la vérité d'une assertion sans divulguer d'informations supplémentaires. Pour cela, un protocole interactif est établi, où le

prouveur envoie des messages au vérificateur et où ce dernier effectue des vérifications pour s'assurer de la validité de l'assertion.

Le prouveur s'engage sur une certaine valeur secrète, le vérificateur émet un défi basé sur cet engagement, et le prouveur répond de manière à prouver la validité de l'assertion sans révéler la valeur secrète.

Les preuves opaques doivent satisfaire trois propriétés essentielles :

**La consistance (*completeness* en anglais).** Le protocole doit être complet, ce qui signifie que si l'assertion est vraie, le vérificateur acceptera la preuve avec une probabilité élevée. En d'autres termes, si le prouveur est honnête et suit le protocole correctement, le vérificateur sera convaincu de la validité de l'assertion.

**La robustesse (*soundness*).** Le protocole doit être sûr, ce qui signifie que si l'assertion est fausse, aucun prouveur ne peut tromper le vérificateur avec succès, sauf avec une probabilité négligeable. Le vérificateur doit pouvoir détecter toute tentative de tromperie de la part du prouveur.

**L'opacité (*zero-knowledge*).** Le protocole doit être à divulgation nulle de connaissance, ce qui signifie que la preuve ne doit révéler aucune information supplémentaire liée à l'assertion autre que le fait qu'elle est vraie. Le vérificateur apprend seulement la validité de l'assertion sans obtenir de connaissances sur les informations secrètes du prouveur.

Les preuves à divulgation nulle de connaissance non interactives permettent de réduire l'interaction requise entre le prouveur et le vérificateur, ce qui améliore l'efficacité des protocoles basés sur les preuves. La non-interactivité a ouvert la voie à l'utilisation plus généralisée des preuves opaques dans les systèmes distribués et les environnements où l'interaction en temps réel n'est pas possible. Ces preuves non interactives sont aussi utilisées par le système d'accréditations anonymes post-quantique sur lequel j'ai travaillé. Vous trouverez une définition de ce type d'accréditations dans la section suivante.

## 1.4 Accréditations anonymes

Les systèmes d'accréditations anonymes permettent à un fournisseur de services de vérifier les droits d'un utilisateur à bénéficier de ces services en vertu de la garantie — donnée antérieurement par une autorité de confiance — du fait que cet utilisateur possède bien les attributs requis, cela en préservant le secret des autres attributs, tels que l'identité de l'utilisateur, dont l'autorité aurait eu à prendre connaissance, cela en permettant aussi la levée de l'anonymat de l'utilisateur par un autre agent, disposant d'une clef secrète spéciale.

Un billet pour un spectacle peut être vu comme une accréditation anonyme dans le monde physique s'il ne comporte pas le nom de son possesseur. Supposons que des

interdictions d'accès à certaines représentations s'appliquent pour les plus jeunes. Le vendeur de places de spectacle s'assure au moyen d'une vérification des documents d'identité que l'âge de l'acheteur est conforme aux restrictions qui s'appliquent : il est alors l'autorité qui atteste, par la délivrance de billet, du droit à assister au spectacle. Dans la salle de spectacle, une ouvreuse peut être amenée à vérifier le billet de l'acheteur dont elle ne vérifie pas elle-même l'âge, le spectateur étant pour elle anonyme. Un autre acteur peut exister, le responsable de la salle de spectacle qui, grâce à un numéro inscrit sur le billet, peut être en mesure de retrouver l'identité du spectateur.

On peut se rendre compte ici qu'il y a donc quatre rôles différents : celui de l'utilisateur, lié à l'identité, qui, même dans le monde numérique correspond souvent à un véritable être humain qui cherche à préserver au mieux sa vie privée, celui de l'autorité de confiance, qui a vue sur tous les éléments permettant l'attestation de certaines propriétés relatives au possesseur de l'identité (comme son âge exacte et son nom), celui du vérificateur d'attributs, pour lequel l'utilisateur reste anonyme, et enfin celui de l'agent d'ouverture qui, muni d'une clef secrète spéciale dans le monde numérique, est en mesure de lever l'anonymat de l'utilisateur, d'obtenir son identité.

Notre protocole d'accréditations anonymes est fondé sur une primitive appelée signature de groupe. Celle-ci permet aux membres d'un groupe de signer au nom du groupe, tout en restant anonyme au sein de ce groupe.

## 1.5 Contributions

J'ai participé, en préparant ma thèse, à la rédaction de trois articles de recherche en cryptographie et à l'écriture du code informatique d'un démonstrateur. Le premier de ces trois articles s'intitule *A Protocol for Secure Verification of Watermarks Embedded into Machine Learning Models* [KTB+21] (« Un protocole de vérification sécurisée de tatouages numériques inclus dans des modèles d'apprentissage automatique ») ; j'ai choisi de concentrer ce mémoire sur les deux autres articles, que je présente dans cette section : ils s'intitulent *Efficient Implementation of a Post-Quantum Anonymous Credential Protocol* [BCR+23] (« Mise en œuvre efficace d'un protocole post-quantique d'accréditations anonymes ») et *Formal Verification of a Post-Quantum Signal Protocol with Tamarin* [BCRS23] (« Vérification formelle d'un protocole Signal post-quantique avec Tamarin »).

### 1.5.1 Mise en œuvre efficace d'un protocole post-quantique d'accréditations anonymes [BCR+23]

Je me suis appliqué à étudier, dans la littérature, les systèmes de signatures de groupe et d'accréditations anonymes fondés sur les problèmes difficiles relatifs aux réseaux euclidiens, puis j'ai participé à l'élaboration d'un schéma d'accréditations anonymes post-quantiques optimisé, pour ensuite contribuer au développement du premier programme informatique à sources ouvertes d'accréditations anonymes post-quantiques qui soit une démonstration directe de la compatibilité d'un tel schéma avec

des cas d'usages concrets. C'est dans le cadre du projet européen H2020 PROMETHEUS que mon équipe et moi avons développé la permière version de ce programme, en tant que démonstrateur. J'y ai par exemple programmé, en langage C, certains éléments de bas niveau comme la transformée de Fourier numérique (souvent appelée NTT, pour *number theoretic transform*), qui permet la réalisation de multiplications rapides dans les anneaux de polynômes à coefficients entiers, modulo un polynôme cyclotomique $X^d + 1$, que nous manipulons dans notre système. Ensuite, afin d'optimiser notre programme, j'ai notamment mis en œuvre un système de compression reposant principalement sur le système de Golomb-Rice, particulièrement bien adapté à la compression des nombreuses valeurs gaussiennes utilisées dans notre schéma.

### 1.5.2 Vérification formelle d'un protocole Signal post-quantique avec Tamarin [BCRS23]

Le protocole Signal, sur lequel se fondent de nombreux systèmes de messagerie instantanée actuels, est appelé à connaître une adaptation post-quantique.

Nous avons proposé la première vérification symbolique, réalisée avec le prouveur Tamarin, de l'une des variantes post-quantiques de Signal présentées dans la littérature, en mettant l'accent sur ses deux principales composantes, le protocole d'échange de clefs X3DH et le protocole du « double-cliquet » de mise à jour des clefs de session.

Tamarin est un outil de preuve automatisée qui utilise la logique du premier ordre pour vérifier que, étant donné une description d'un protocole sous forme de machine à états et étant donné des propositions correspondant à des propriétés de sécurité, lesdites propriétés sont bien satisfaites par le protocole, sous l'hypothèse que les primitives de celui-ci se comportent comme des primitives idéales.

Cette analyse nous a permis de vérifier que, instancié avec le protocole d'échange de clefs de Hashimoto, Katsumata, Kwiatkowski et Prest [HKKP21a] et le double-cliquet de Alwen, Coretti, et Dodis [ACD19], ce protocole Signal post-quantique présente des propriétés de sécurité équivalentes au protocole classique.

# Références

[ACD19] Joël Alwen, Sandro Coretti et Yevgeniy Dodis. *The Double Ratchet : Security Notions, Proofs, and Modularization for the Signal Protocol*. In : *EUROCRYPT 2019, Part I*. Sous la dir. d'Yuval Ishai et Vincent Rijmen. T. 11476. LNCS. Springer, Heidelberg, mai 2019, p. 129-158 (cf. p. 11, 40, 41, 48, 51).

[Ajt96] Miklós Ajtai. *Generating Hard Instances of Lattice Problems (Extended Abstract)*. In : *28th ACM STOC*. ACM Press, mai 1996, p. 99-108 (cf. p. 7).

[BCR+23] Olivier Blazy, Céline Chevalier, Thomas Ricosset, Guillaume Renaut, Éric Sageloli et Hugo Senet. *Efficient Implementation of a Post-Quantum Anonymous Credential Protocol*. In : *ARES'23 – Proceedings of the 18th International Conference on Availability, Reliability and Security*. Association for Computing Machinery, 2023 (cf. p. 10).

[BCRS23] Hugo Beguinet, Céline Chevalier, Thomas Ricosset et Hugo Senet. *Formal Verification of a Post-Quantum Signal Protocol with Tamarin*. In : *VECoS'23 – Proceedings of the 16th International Conference on Verification and Evaluation of Computer and Communication Systems*. LNCS. Springer, 2023 (cf. p. 10, 11).

[Gal46] Évariste Galois. *Mémoire sur les conditions de résolubilité des équations par radicaux*. In : *Journal de mathématiques pures et appliquées* XI (1846), p. 417-433 (cf. p. 4).

[Gau01] Johann Carl Friedrich Gauss. *Disquisitiones arithmeticae*. Gerhard Fleischer, 1801 (cf. p. 4).

[HKKP21a] Keitaro Hashimoto, Shuichi Katsumata, Kris Kwiatkowski et Thomas Prest. *An Efficient and Generic Construction for Signal's Handshake (X3DH) : Post-Quantum, State Leakage Secure, and Deniable*. In : *PKC 2021, Part II*. Sous la dir. de Juan Garay. T. 12711. LNCS. Springer, Heidelberg, mai 2021, p. 410-440 (cf. p. 11).

[Ker83a] Auguste Kerckhoffs. *La cryptographie militaire*. In : *Journal des sciences militaires* IX (jan. 1883), p. 5-38 (cf. p. 4).

[Ker83b] Auguste Kerckhoffs. *La cryptographie militaire*. In : *Journal des sciences militaires* IX (février 1883), p. 161-191 (cf. p. 4).

[KTB+21] Katarzyna Kapusta, Vincent Thouvenot, Olivier Bettan, Hugo Beguinet et Hugo Senet. *A Protocol for Secure Verification of Watermarks Embedded into Machine Learning Models*. In : *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security*. IH&MMSec '21. Virtual Event, Belgium : Association for Computing Machinery, 2021, p. 171-176. isbn : 9781450382953. (Cf. p. 10).

[Reg05] Oded Regev. *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*. In : *37th ACM STOC*. Sous la dir. d'Harold N. Gabow et Ronald Fagin. ACM Press, mai 2005, p. 84-93 (cf. p. 8).

« Three may keep a Secret, if two of them are dead. »[a]

[a] « Trois personnes peuvent garder un secret, si deux d'entre elles sont mortes. »

*Poor Richard's Almanack* –
Benjamin Franklin

# Efficient Implementation of a Post-Quantum Anonymous Credential Protocol

2



Authentication on the Internet usually has the drawback of leaking the identity of the users, or at least allowing to trace them from a server to another. Anonymous credentials overcome this issue, by allowing users to reveal the attributes necessary for the authentication, without revealing any other information (in particular not their identity). In this chapter, we provide a generic framework to construct anonymous credential schemes and use it to give a concrete construction of post-quantum (lattice-based) anonymous credential protocol. Our protocol thus allows for long-term security even when one considers the emergence of quantum computers able to break widely used traditional computational assumptions, such as RSA, the discrete logarithm or Diffie-Hellman. We also give a concrete implementation of our protocol, which is only one order of magnitude slower and bandwidth consuming than previous anonymous credentials that are not post-quantum.

## 2.1 Introduction

**Authentication techniques.** The traditional way to prove one's identity on the Internet is to ask an authority to provide a credential. These techniques can deliver a high level of security, however, the most commonly used ones result in some privacy issues. By not respecting the user's anonymity, they can allow him to be traced from server to server. For instance, signature schemes suffer from an *over identification* issue, meaning that due to the static nature of public keys, users have to reveal their identities (i.e., their public keys) to be authenticated. *Anonymous credentials*, initially proposed by Chaum [Cha85], can be used to solve these privacy issues by revealing only the attributes needed for the authentication, or proving some properties on them, or answering questions about a subset of these attributes, without having to reveal more information, such as the identity of the users.

An anonymous credential may be, for example, used to encode identity documents issued by the government. Through this credential, the state certifies attributes such as citizenship and date of birth. The citizen can store this credential, for example, on his or her smartphone and use it to prove statements about his or her certified attributes while staying unlinkable across services. Nowadays, conventional anonymous

credential systems are used in more than 500 million trusted platform modules embedding direct anonymous attestations and billions of Intel processors implementing EPID systems. The IBM Identity Mixer and Microsoft U-Prove are also based on conventional anonymous credentials.

In an anonymous credential protocol, a user is collecting credentials from various issuers. He or she can finely control which information from which credentials he or she presents to which verifier by the means of a *presentation token*. An issuer issues credentials to users, thereby vouching for the correctness of the attributes contained in the credential with respect to the user to whom the credential is issued. By verifying the presentation token that a user presents, a verifier protects access to a resource or service that it offers by imposing restrictions on the credentials that users must own and the information from the attributes contained in these credentials that users must present in order to access the service. Finally, an inspector (or opener) is a trusted authority who can de-anonymize presentation tokens under specific circumstances.

**Constructions of anonymous credentials.** Due to their practical interest, these schemes have received a lot of attention from the cryptographic community, for instance [Bra00; CL01; CL02; CL03; CL04; BCKL08; ILV11; GGM14; CDHK15; BCN18].

To the best of our knowledge, [BCN18] yields to the only post-quantum scheme that has been proposed so far, but with no concrete instantiation nor implementation. Indeed, all the others schemes rely on classical assumptions, based for instance on discrete logarithm or Diffie-Hellman. It raises concerns about their long-term security, as the emergence of quantum computing in recent years threatens any cryptographic protocol based on assumptions that are not assumed quantum-safe. Indeed, quantum computers would potentially break, even retroactively, the mathematical foundations of many current cryptographic systems including the difficulty of the Diffie-Hellman problem. In response to this potential threat to current cryptographic systems, the National Institute of Standards and Technology (NIST) has launched a standardization process for post-quantum cryptographic primitives in 2016. The goal of this campaign is to provide new post-quantum standards for two basic and crucial cryptographic building blocks: Key Encapsulation Mechanisms and digital signatures. A few families of mathematical problems are considered quantum-safe nowadays and can be considered for the design of candidate algorithms like error correcting codes or lattices. We focus in this article on lattice-based constructions.

The usual way of constructing an anonymous credential protocol is either from a signature scheme with efficient protocols or from group signatures. We focus here on the latter case. A group signature scheme, first introduced by Chaum and van Heyst in 1991 [Cv91], allows a member of a group to anonymously sign a message on behalf of the group. It can be used in settings where it is sufficient for a verifier to know that a message was sent by a person from a certain group, but without needing to know who in particular signed it. Such schemes intrinsically allow for anonymity of the users. In our setting, it allows to reveal an attribute but not the identity of the user. It generally consists of a signature on a committed value, commitments being an analogue of a sealed envelope, allowing one to hide a value without being able to change

his or her mind later on. Intuitively, the choice of the signature scheme determines the efficiency and the security of the resulting anonymous credential protocol. Each such lattice-based signature schemes has its own strengths, but those with tight reduction to standard lattice problems are not practical enough to lead to practical anonymous credential protocols.

**Properties for Anonymous Credentials.** Three notions capture the security of our anonymous credential protocol: *anonymity*, *unforgeability* and *traceability*. For anonymity, we consider a probabilistic polynomial-time adversary who has access to all the credentials as well as to a quantum computer with polynomial space. The adversary chooses policy rules for the attributes and two credentials whose attributes pass the verification of this policy. The challenger then returns a presentation token for one of these credentials. The adversary's goal is to guess which credential has been used to issue this presentation token, i.e. to distinguish between presentation tokens for these credentials.

Unforgeability captures the notion that an adversary cannot create a valid presentation token without using a credential or restraining a valid presentation token to a smaller subset of attributes.

Traceability captures the notion that all presentation tokens, even when computed by a collusion of users and inspectors, should trace to a member of the forging coalition. For the traceability game, the adversary has access to all the inspectors' decryption oracles, as well as all the credentials for any subset of identities, and has access to a quantum computer with polynomial space. The adversary's goal is to produce a valid presentation token, i.e. which passes verification, such that an inspection of this presentation token fails or traces to an identity which is not in the given subset of identities.

For all security properties, the proof is done by reduction, meaning that the idea is to prove that if an adversary breaks either property of the anonymous credential protocol in quantum polynomial time with a non-negligible advantage, then one can construct an adversary which breaks the underlying computational problem (such as MLWE, MSIS or NTRU) in polynomial time with a non-negligible advantage.

**Our Contributions.** Following the definitions given in [BCN18] and the framework presented in [dLS18], we give a generic framework for anonymous credential schemes, which encompasses the construction given in [BCN18].

Furthermore, we use this framework to give a concrete construction of lattice-based anonymous credential system, which is instantiated with the group signature from [dLS18] and zero-knowledge arguments from [ALS20] (further improved in [LNP22]). We use the lattice estimator [APS15] to estimate the security of the resulting anonymous credential protocol and choose a concrete set of parameters.

We also give a concrete implementation of our protocol, using the NTT along with other optimizations, thanks to the zero-knowledge arguments from [ALS20]. As a result, our implementation of a post-quantum anonymous credential is only an order of magnitude slower and bandwidth consuming than previous anonymous credentials that are not post-quantum.

## 2.2 Technical Overview

We construct an anonymous credential scheme (the generic construction is presented in Section 2.4) using a (relaxed) signature scheme and a (relaxed) verifiable encryption scheme that are compatible, a verifiable encryption scheme being simply an encryption scheme that allows to prove some properties about the cleartext using the ciphertext. These properties are encoded using relations and we use non-interactive zero-knowledge proofs (NIZK) to show the existence of a witness showing that an element fulfills the relation (without revealing any other information). The relaxation aspect essentially means that:

- for the signature: we allow the verification of more signatures that the ones that can be constructed by the signature algorithm Sign. These are called relaxed signatures, and are in relation with the "normal" signatures: each relaxed signature being associated to a normal signature, and called its relaxation. For the forgery game, we forbid the adversary to send a relaxation of a signature it obtained from the signature oracle.
- for the NIZK proofs of the existence of a witness $w$ for an element $x$, we allow the extractor to output a relaxed witness $\bar{w}$ of $x$, a relaxed witness of $x$ being a witness of a relaxation of $x$, for a relation associated to the first one, called the relaxed relation.
- for the verifiable encryption, we allow the decryption algorithm applied to a ciphertext $c$ and a NIZK proof $\pi$ for this cipher to either extract a witness, leading to a cleartext, which is a relaxed message, or decrypt the ciphertext leading to a message $\mu$ or relaxed message. These two distinct ways have to be compatible in the sense that both relaxed messages obtained must lead to the same message.

The issuer public and secret keys are the public and secret keys of a signature scheme and a verifiable encryption scheme.

An improvement upon the protocol in [dLS18] is that the credential for an identity id and attribute $\alpha$ is given by a signature of both id and $\alpha$, allowing us to use a unique group signature for issuing a credential. In our construction, the signature is a short $(s_1, s_2, s_3)$ such that

$$\left[\mathbf{a}^T \mid \mathbf{b}^T + \mathrm{id}\mathbf{g}^T \mid (1 \quad a_3)\right] \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \mathcal{H}(\alpha) \mod q_2 \tag{2.1}$$

where $(\mathbf{a}, \mathbf{b})$ is the public key and $\mathbf{g} = (1 \ \delta)$ a gadget matrix.

A presentation of an attribute $\alpha$ then consists of the encryption of id and of a zero-knowledge proof of knowledge of id and a (relaxed) signature of the identity and $\alpha$.

The verifiable encryption and associated proof need the introduction of a commitment (we choose the one of presented in [BDL+18] and improved in [ALS20]) which will have two aims. First, it is the main tool in order to transform the signature of a (visible) identity into a simple linear relation that hides the identity and will be used

for the NIZK proof (note that the attribute $\alpha$ is still visible in the signature). To obtain this relation, the user computes two commitments of id and $\delta$id:

$$\mathbf{t} \leftarrow \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \mathbf{A}\mathbf{r} + \begin{bmatrix} \mathbf{0} \\ \mathrm{id} \end{bmatrix} \mod \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$
$$\mathbf{t}' \leftarrow \begin{bmatrix} t'_1 \\ t'_2 \end{bmatrix} = \mathbf{A}\mathbf{r}' + \begin{bmatrix} \mathbf{0} \\ \delta\mathrm{id} \end{bmatrix} \mod \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

where $\mathbf{A} = \begin{bmatrix} 1 & a_1 & a_2 \\ 0 & 1 & a_3 \end{bmatrix} \in \mathcal{R}^{3\times 2}$ and observes that, taking $\bar{\mathbf{r}}$ (resp. $\bar{\mathbf{r}}'$) the last two coordinates of $\mathbf{r}$ (resp. $\mathbf{r}'$), $\mathbf{comsign} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 - [\bar{\mathbf{r}}\ \bar{\mathbf{r}}']s_2 \end{bmatrix}$ is a short solution of:

$$\left[\mathbf{a}^T \mid \mathbf{b}^T + [t_2\ t'_2] \mid (1\ a_3)\right]\mathbf{comsign} = \mathcal{H}(\alpha) \mod q_2 \tag{2.2}$$

Secondly, this **comsign** element is used in two NIZK proofs: The first one proves the knowledge of a small solution **comsign**, and the second one shows (by considering the random coins) that the committed value id is the same as the encrypted value (encrypted using an auxiliary verifiable encryption scheme, introduced in[LN17]).

Such a verifiable encryption and proof of knowledge was already present in the signature group construction of [dLS18]. The main differences are as follows:

- Their signature did not contain the term $\mathcal{H}(\alpha)$:

$$\left[\mathbf{a}^T \mid \mathbf{b}^T + \mathrm{id}\mathbf{g}^T \mid (1 \quad a_3)\right] \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = 0 \mod q_2$$

  This term is present in our scheme, which allows to link the attribute and the identity, and thus optimize our protocol, by allowing to construct NIZK proofs that hide the identity while leaving the attribute visible.

- It was necessary that the nonzero differences of challenges were invertible modulo multiple primes involved in the scheme. In order to ensure the existence of enough invertible nonzero differences of challenges, these primes numbers $u$ were taken in a such a way that $X^d + 1$ would split into (only) two components in the ring $\mathcal{R}_u$. We consider in this article the framework of [ALS20] that only needs the nonzero differences of challenges to be "sufficiently often" invertible. This allows to use prime numbers $u$ such that the rings $\mathcal{R}_u$ split into (way) more components. This allows for faster polynomial multiplications, and thus an efficiency gain. More precisely, when we use "partial splitting", i.e. when we choose in our implementation most prime numbers $(q_1, q_2, Q)$ such that each element of $\mathcal{R}_u$ is represented as 2048 polynomials of degree 4 via NTT (instead of 2 polynomials of degree 4096 when $X^d + 1$ split into two elements), we obtain

a term-to-term multiplication of these polynomials, which is way faster than a direct multiplication in $\mathcal{R}_u$ (which would lead to a unique multiplication between polynomials modulo $X^{8196}+1$). It is interesting to note that most of the changes implied by this framework do not change the algorithms but only the security proofs of the scheme.

As for the anonymity of credentials and identities, the security of our scheme relies on the zero-knowledge property of the NIZK proofs, the hiding property of the commitment scheme and the IND-CPA security of the encryption scheme. As for the unforgeability and traceability, it relies on the special soundness properties of the NIZK proofs and of the unforgeability of the underlying signature scheme (formal lemmas and proofs can be found in the full version).

## 2.3 Preliminaries

**Notation.** Throughout the paper, we will write $\mathcal{R} = \mathbb{Z}[X]/(X^d+1)$ for a $d$ that will be a power of two, in order for $X^d+1$ to be cyclotomic. For a prime $q$, we will note $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d+1)$.

We will work with the usual norms, for $w \in \mathcal{R}$, $w = \sum_{i=1}^{d} w_i X^i$,

$$\|w\| = \sqrt{\sum_{i=0}^{d-1} |w_i|^2} \quad \|w\|_1 = \sum_{i \in [\![0,d-1]\!]} |w_i| \quad \|w\|_\infty = \max_{i \in [\![0,d-1]\!]} |w_i|$$

for $\mathbf{w} \in \mathcal{R}^n$, $\mathbf{w} = (w_1, \cdots, w_n)$ we note

$$\|\mathbf{w}\| = \sqrt{\sum_{i=1}^{n} \|w_i\|^2} \quad \|\mathbf{w}\|_1 = \max_{i \in [\![1,n]\!]} \|w_i\|_1 \quad \|\mathbf{w}\|_\infty = \max_{i \in [\![1,n]\!]} \|w_i\|_\infty$$

We will also use the norms modulo multiples primes $u$, defined as the previous norms for the representative with coefficients taken in $[\![-(u-1)/2, (u-1)/2]\!]$. The prime will be indicated as is $\|\cdot\|_u, \|\cdot\|_{u,1}, \|\cdot\|_{u,\infty}$, unless if the context is clear enough to remove it. For $\beta > 0$, we will note $S_\beta = \{a \in \mathcal{R} : \|a\|_\infty \leq \beta\}$. For a product of spaces $X \times Y$, we will note $\mathsf{proj}_X$ the projection on $X$. We write $x \leftarrow\!\!\$\, X$ the uniform sampling of an element $x \in X$.

For $\sigma > 0$, we denote by $\mathcal{D}_{\mathcal{R},\sigma}$ the discrete Gaussian distribution on $\mathcal{R}$ with standard deviation $\sigma$. More precisely, each coordinate is taken by a discrete Gaussian distribution on $\mathbb{Z}$, of standard deviation $\sigma$. The $\mathcal{R}$ will be omitted when it is clear from context.

**Prime Splitting and Galois Automorphisms.** As in [ALS20], we take $\tau$, $\theta$, $d$ powers of two such that $\tau \mid \theta \mid d$ and we will consider primes numbers $q_1$, $q_2$, $Q$ such that

$$\forall u \in \{q_1, q_2, Q\} : u - 1 \equiv 2\theta \mod 4\theta. \tag{2.3}$$

[ALS20] shows that for each $u \in \{q_1, q_2, Q\}$, there exist a primitive $2\theta$ root of unity $\zeta_u \in \mathbb{Z}_u$. Using the Chinese remainder theorem we know that there is a $\zeta \in \mathbb{Z}$, $\zeta \leq q_1 \cdot q_2 \cdot Q$, such that $\zeta \mod u = \zeta_u$ for $u \in \{q_1, q_2, Q\}$. Let $\mathsf{I} = \mathbb{Z}_{2d}^{\times}/\langle 2\theta + 1 \rangle$ and $\phi_i = X^{d/\theta} - \zeta^i$.

As shown in [ALS20, Section2.2], the $\phi_i$ form the irreducible decomposition of $X^d+1$ modulo $u$: $X^d + 1 \mod q = \prod_{i \in \mathsf{I}} \phi_i$.

Following the work of [ALS20], we will note:

$$\mathsf{I}[\tau] = \mathbb{Z}_{2\theta/\tau}^{\times} \times [\![0, \tau - 1]\!] \qquad \sigma[\tau] := \sigma_{2(\theta/\tau)+1}$$
$$\Phi[\tau]_i = X^{\tau d/\theta} - \zeta^{i\tau} \qquad \phi[\tau]_{i,j} = \sigma[\tau]^j (X^{d/\theta} - \zeta^i)$$

**MSIS, MLWE and NTRU problems.** We consider the same definitions than in [BCN18].

**Definition 2.1.** *The* $\mathsf{MSIS}_{q,n,m,\beta}$ *problem (over an implicit ring* $\mathcal{R}$*) is defined as follows. Given* $\mathbf{A} \in \mathcal{R}_q^{n \times m}$ *sampled uniformly at random. Find* $\mathbf{s} \in \mathcal{R}^m$ *such that* $\mathbf{As} = 0$ *and* $0 < \|\mathbf{s}\| \leq \beta$*.*

**Definition 2.2.** *The decision* $\mathsf{MLWE}_{q,m,n}$ *problem (over an implicit ring* $\mathcal{R}$*) is defined as follows. Given* $\mathbf{s} \xleftarrow{\$} S_1^n$*, let* $A_{q,\mathbf{s}}$ *the distribution obtained by sampling* $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q^n$*,* $e \xleftarrow{\$} S_1$*, and returning* $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathcal{R}_q^n \times \mathcal{R}_q$*. The goal is to distinguish between* $m$ *samples from either* $A_{q,\mathbf{s}}$ *or the uniform distribution in* $\mathcal{R}_q^n \times \mathcal{R}_q$*.*

**Definition 2.3.** *The* $\mathsf{NTRU}_{q,r}$ *problem (over an implicit ring* $\mathcal{R}$*) is defined as follows. The distribution* $A$ *is defined by sampling ring elements* $f, g \xleftarrow{\$} D_r$ *and outputting* $h = f/g$ *if* $g$ *is invertible in* $\mathcal{R}_q$ *(otherwise, re-sample* $g$*). The goal is to distinguish* $h$ *from a random element in* $\mathcal{R}_q$*.*

**Relations, relaxed sets and relaxed relations.** A relaxed set is a triplet

$$(X, \bar{X}, \mathsf{proj}_X)$$

where $X$, $\bar{X}$ are sets and $\mathsf{proj}_X$ is a surjection $\bar{X} \to X$. We will sometimes call $\bar{X}$ the relaxation of $X$ and see $\mathsf{proj}_X$ as a function $\bar{X} \sqcup X \to X$ that is the identity on X.

An element $\bar{x}$ will be called a relaxation of $x$ if $\mathsf{proj}_X(\bar{x}) = x$ or if $\bar{x} = x$. If $\bar{x} \neq x$, we will speak of strict relaxation. The elements of $\bar{X}$ will be called the relaxed elements.

We define the language and the witness associated to a relation $\mathcal{R}$ as usual by setting $\mathcal{L}(\mathcal{R}) = \{x : \exists y, \mathcal{R}(x, y)\}$ and $\mathcal{W}(\mathcal{R}) = \{y : \exists x, \mathcal{R}(x, y)\}$.

For an element $x = (a, b) \in \mathcal{R}$, we will write $a = \mathsf{proj}_{\mathcal{L}}(x)$ its projection in $\mathcal{L}(\mathcal{R})$ and $b = \mathsf{proj}_{\mathcal{W}}(x)$ its projection in $\mathcal{W}(\mathcal{R})$.

A relaxed relation is a triplet $(\mathcal{R}, \overline{\mathcal{R}}, \mathsf{proj}_R)$ where $\mathcal{R}$ and $\overline{\mathcal{R}}$ are relations and $(\mathcal{L}(\mathcal{R}), \mathcal{L}(\overline{\mathcal{R}}), \mathsf{proj}_R)$ is a relaxed set. We will call $\overline{\mathcal{R}}$ the relaxation of $\mathcal{R}$. A relaxed witness of an element $x$ will be any witness of $\bar{x}$ such that $\bar{x}$ is a relaxation of $x$. We will sometime speak of strict witness if $\bar{x}$ is a strict relaxation of $x$.

## 2.4 Anonymous Credential Scheme

### 2.4.1 Definition

An *anonymous credential scheme* for a set of identities Identities and a set of attributes Attributes is a tuple of algorithms (SPGen, IKGen, OKGen, Issue, Present, Open, VerifyCred, VerifyPt, ExtractAttr):

- The **parameter generation algorithm SPGen** returns the public parameters acpar.
- The **issuer key generation algorithm IKGen(acpar)** returns an issuer (public, secret) key pair (ipk, isk).
- The **opener key generation algorithm OKGen(acpar, ipk)** returns an opener (public, secret) key pair (opk, osk).
- The **credential issuance algorithm Issue(isk, id, $((\alpha_i)_{i\in R})$)** creates a credential cred associated to an identity id and a set of attributes $(\alpha_i)_{i\in R}$. It is possible to recover the set of attribute $(\alpha_i)_{i\in R}$ from cred computing ExtractAttr(cred). We call $(\alpha_i)_{i\in R}$ the attributes associated to cred.
- The **presentation algorithm Present(acpar, ipk, opk, cred, $S$)** creates a presentation token pt using a credential cred with associated attributes $(\alpha_i)_{i\in R}$ and $S \subset R$. It is possible to recover the subset of attributes $(\alpha_i)_{i\in S}$ from pt computing ExtractAttr(pt). We call $(\alpha_i)_{i\in S}$ the attributes associated to pt.
- The **opening algorithm Open(acpar, ipk, osk, pt)** returns the identity of the credential used to create the presentation (or $\perp$).
- The **credential verification algorithm VerifyCred(acpar, ipk, cred)** checks the validity of the credential for the associated attributes $(\alpha_i)_{i\in R}$ it contains and returns 1 (if valid) or 0.
- The **token verification algorithm VerifyPt(acpar, ipk, opk, pt)** checks the validity of the presentation token for the associated attributes $(\alpha_i)_{i\in S}$ it contains and returns 1 (if valid) or 0.

We consider the following security properties (formally stated in [BCN18, Appendix G.1])

- **Anonymity:** A user cannot distinguish from which credential a presentation is made.
- **Unforgeability:** A user cannot create a valid presentation token without using a credential or restraining a valid presentation token to a smaller subset of attributes.
- **Traceability:** All presentations should trace to an issuer. If computed by a collusion of users, it should trace to a member of this collusion.

### 2.4.2 Construction

Our anonymous credential scheme builds upon a relaxed signature scheme and a relaxed verifiable encryption scheme that are compatible, along with associated (relaxed) NIZK proofs. We state in the following subsections our definitions for such schemes

before presenting our generic scheme.

**Relaxed Signature Scheme.** We use the same definition as in [BCN18, Section 5.1]. The only difference is our definition of a relaxed set: instead of using a function $g : X \to 2^{\bar{X}}$, we consider a surjection $\bar{X} \to X$. One can easily switch from one vision to the other.

A relaxed signature scheme associated with a relaxed message space

$$(\mathcal{M}_{\mathsf{sign}}, \overline{\mathcal{M}}_{\mathsf{sign}}, \mathsf{proj}_{\mathsf{sign}})$$

consists of the following algorithms:

- The **key generation algorithm SignKeyGen()** returns a couple of (public, secret) signature keys $(\mathsf{ssk}, \mathsf{spk})$.
- The **signature algorithm Sign($\mathsf{ssk}, m$)** computes the signature $\sigma$ of a message $m$ with secret key $\mathsf{ssk}$.
- The **verification algorithm SignVf($\mathsf{ssk}, \sigma, m$)** verifies if $\sigma$ is a valid signature of $m \in \mathcal{M}_{\mathsf{sign}} \sqcup \overline{\mathcal{M}}_{\mathsf{sign}}$ for the public key $\mathsf{spk}$. It returns 1 if it is valid and 0 otherwise.

The idea is to create signatures for messages, but verify both messages and relaxed messages.

In the definition of the forgery game, we forbid the adversary to send a signature of a relaxation of a message it obtained the signature from the signature oracle. The detailed game is described in [BCN18, Section 5.1].

**Relaxed NIZK proofs (in the random oracle model).** Our general scheme uses the notion of non-interactive zero-knowledge (NIZK) proof system for a relaxed relation $(\mathcal{R}, \overline{\mathcal{R}}, \mathsf{proj}_R)$. Their definition is a slight adaption of the one stated in [BCN18, Section 3.1]. It is important to note that such a protocol only allows to compute proofs for elements $x \in \mathcal{L}(\mathcal{R})$, when a witness $w$ is known, and that the relaxed special soundness property applied for proofs of $x$ only allows to extract a relaxed witness $\bar{w}$ of $x$, so that $(x, \bar{w}) \in \overline{\mathcal{R}}$.

As a side note, the challenges for the NIZK proofs are not chosen uniformly but according to a distribution $\mathcal{X}(\rho)$ (see Table 2.6), which is thus also the case for the hash function $\mathcal{H}^{ZK}$.

**Relaxed Verifiable Encryption.** Even if our instantiation makes use of the verifiable encryption described in [LN17], our definition of a verifiable encryption is slightly different and its instantiation actually uses both the verifiable encryption of [LN17] and the commitment scheme of [ALS20]. This is explained by the fact that a verifiable encryption does not only encrypt an element, but a relation fulfilled by the element.

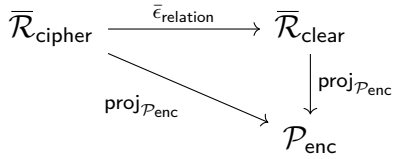
We consider a relaxed space of clear messages $(\mathcal{M}_{\mathsf{enc}}, \overline{\mathcal{M}}_{\mathsf{enc}}, \mathsf{proj}_{\mathsf{mess}})$ and probabilistic algorithms $(\mathsf{EncKeyGen}, \mathsf{Enc}, \mathsf{Dec})$ such that

- $\mathsf{EncKeyGen}$ outputs a couple of (public, secret) keys $(\mathsf{epk}, \mathsf{esk}) \in \mathsf{Pk}_{\mathsf{enc}} \times \mathsf{Sk}_{\mathsf{enc}}$.
- For $(\mathsf{epk}, \mu) \in \mathsf{Pk}_{\mathsf{enc}} \times \mathcal{M}_{\mathsf{enc}}$, $\mathsf{Enc}(\mathsf{epk}, \mu)$ outputs a ciphertext $c \in \mathcal{C}$.
- For a ciphertext $c \in \mathcal{C}$ and a secret key $\mathsf{esk} \in \mathsf{Sk}_{\mathsf{enc}}$, $\mathsf{Dec}(\mathsf{esk}, c)$ outputs a message $\mu \in \mathcal{M}_{\mathsf{enc}} \sqcup \overline{\mathcal{M}}_{\mathsf{enc}} \sqcup \{\perp\}$. We say that the decryption fails if $\mu = \perp$.

We require that $(\mathsf{EncKeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is a IND-CPA encryption scheme with message space $\mathcal{M}_{\mathsf{enc}}$ and ciphertext space $\mathcal{C}$.

Moreover, we consider

- two relaxed relations $(\mathcal{R}_{\mathsf{clear}}, \overline{\mathcal{R}}_{\mathsf{clear}}, \mathsf{proj}_{\mathsf{clear}})$, $(\mathcal{R}_{\mathsf{cipher}}, \overline{\mathcal{R}}_{\mathsf{cipher}}, \mathsf{proj}_{\mathsf{cipher}})$ with: $\mathcal{L}(\mathcal{R}_{\mathsf{clear}}) = \mathcal{M}_{\mathsf{enc}} \times \mathcal{P}_{\mathsf{enc}}$, $\mathcal{L}(\overline{\mathcal{R}}_{\mathsf{clear}}) = \overline{\mathcal{M}}_{\mathsf{enc}} \times \mathcal{P}_{\mathsf{enc}}$, $\mathsf{proj}_{\mathsf{clear}} = (\mathrm{id}, \mathsf{proj}_{\mathsf{mess}})$. $\mathcal{L}(\mathcal{R}_{\mathsf{cipher}}) = \mathcal{L}(\overline{\mathcal{R}}_{\mathsf{cipher}}) = \mathsf{Pk}_{\mathsf{enc}} \times \mathcal{C} \times \mathcal{P}_{\mathsf{enc}}$ and $\mathsf{proj}_{\mathsf{cipher}} = \mathrm{id}$.
- a NIZK proof system for the relaxed relation $(\mathcal{R}_{\mathsf{cipher}}, \overline{\mathcal{R}}_{\mathsf{cipher}}, \mathsf{proj}_{\mathsf{cipher}})$,
- a subset $\mathsf{DomEncRel} \subset \mathsf{Pk}_{\mathsf{enc}} \times \mathcal{R}_{\mathsf{clear}}$,
- a probabilistic algorithm $\mathsf{EncRel} : \mathsf{DomEncRel} \to \mathcal{R}_{\mathsf{cipher}}$ that extends the function $\mathsf{Enc}$ to relation thanks to the requisite $\mathsf{proj}_{\mathcal{L}} \circ \mathsf{EncRel} = (\mathsf{Enc}, \mathsf{id}) \circ (\mathsf{id}, \mathsf{proj}_{\mathcal{L}})$,
- a relaxed extraction of relation function $\bar{\epsilon}_{\mathsf{relation}} : \overline{\mathcal{R}}_{\mathsf{cipher}} \to \overline{\mathcal{R}}_{\mathsf{clear}}$ such that the following diagram commutes:




Finally, we ask the following property, that expresses the compatibility of the decryption algorithm and of the definition of the witness of $\overline{\mathcal{R}}_{\mathsf{cipher}}$.

- For $(\mathrm{epk}, \mathrm{esk}) \xleftarrow{\$} \mathsf{EncKeyGen}$, $c \in \mathcal{C}$ and for all $X \in \overline{\mathcal{R}}_{\mathsf{cipher}}$ such that

$$\mathrm{epk} = \mathsf{proj}_{\mathsf{Pk}_{\mathsf{enc}}}(\mathsf{proj}_{\mathcal{L}}(X)), \;\; c = \mathsf{proj}_{\mathcal{C}}(\mathsf{proj}_{\mathcal{L}}(X))$$

the probability to not have

$$\mathsf{proj}_{\mathsf{mess}}\Big(\mathsf{Dec}(\mathrm{esk}, c)\Big) = \mathsf{proj}_{\mathsf{mess}}(\mathsf{proj}_{\overline{\mathcal{M}}_{\mathsf{enc}}}\Big(\bar{\epsilon}_{\mathsf{relation}}(X)\Big))$$

is negligible. Here, we take the convention that $\mathsf{proj}_{\mathsf{mess}}(m) = m$ if $m \in \mathcal{M}_{\mathsf{enc}}$.

**Anonymous Credential.** Our generic construction will need to use relaxed signature and relaxed verifiable encryption schemes with the following requirements:

- $\mathcal{M}_{\mathsf{sign}} = \mathcal{M}_{\mathsf{enc}} \times \mathsf{Attributes}$, $\overline{\mathcal{M}}_{\mathsf{sign}} = \overline{\mathcal{M}}_{\mathsf{enc}} \times \mathsf{Attributes}$
- The relaxed function of the signature scheme preserves the attributes: $\mathsf{projAttributes} \circ \mathsf{proj}_{\mathsf{sign}} = \mathsf{projAttributes}$.
- The relations of the relaxed verifiable encryption are such that:

$$\begin{aligned}
\mathcal{R}_{\mathsf{clear}} := \Big\{ \Big(\big(\mathrm{id}, \mathsf{spk}, R, (\alpha_i)_{i\in R}\big), \big((\sigma_i)_{i\in R}\big)\Big) \;:\; \\
\forall i \in R,\, \mathsf{SignVf}(\mathsf{spk}, (\mathrm{id}, \alpha_i), \sigma_i) = 1 \Big\} \\
\overline{\mathcal{R}}_{\mathsf{clear}} := \Big\{ \Big(\big(\overline{\mathrm{id}}, \mathsf{spk}, R, (\alpha_i)_{i\in R}\big), \big((\overline{\mathrm{id}}_i)_{i\in R}, (\sigma_i)_{i\in R}\big)\Big) \;:\; \\
\forall i \in R, \mathsf{SignVf}(\mathsf{spk}, (\mathrm{id}, \alpha_i), \sigma_i) = 1 \\
\wedge\, \mathsf{proj}_{\mathsf{sign}}(\overline{\mathrm{id}}_i) = \mathsf{proj}_{\mathsf{sign}}(\overline{\mathrm{id}}) \Big\}
\end{aligned}$$

$\mathsf{SPGen}()$

$\mathsf{acpar} \leftarrow \perp$
**return** $\mathsf{acpar}$

$\mathsf{IKGen}(\mathsf{acpar})$

$(\mathsf{spk}, \mathsf{ssk}) \leftarrow \mathsf{SignKeyGen}()$
$\mathsf{ipk} \leftarrow \mathsf{spk}$
$\mathsf{isk} \leftarrow \mathsf{ssk}$
**return** $(\mathsf{ipk}, \mathsf{isk})$

$\mathsf{OKGen}(\mathsf{acpar}, \mathsf{ipk})$

$(\mathsf{epk}, \mathsf{esk}) \leftarrow \mathsf{EncKeyGen}()$
$\mathsf{opk} \leftarrow \mathsf{epk}$
$\mathsf{osk} \leftarrow \mathsf{esk}$
**return** $(\mathsf{opk}, \mathsf{osk})$

$\mathsf{Issue}(\mathsf{isk}, \mathrm{id}, (\alpha_i)_{i\in R})$

**for** $i \in R$:
  $\sigma_i \leftarrow \mathsf{Sign}(\mathsf{ipk}, (\mathrm{id}, \alpha_i \sqcup i))$
$\mathsf{cred} \leftarrow (\mathrm{id}, (\alpha_i)_{i\in R}, (\sigma_i)_{i\in R})$
**return** $\mathsf{cred}$

$\mathsf{Present}\left(\mathsf{acpar}, \mathsf{ipk}, \mathsf{opk}, \mathsf{cred} = \left(\mathrm{id}, (\alpha_i)_{i\in R}, (\sigma_i)_{i\in R}\right), S\right)$

$x \leftarrow \left(\left(\mathrm{id}, \mathsf{spk}, S, (\alpha_i)_{i\in S}\right), \left((\sigma_i)_{i\in S}\right)\right)$
  // $(\mathsf{epk}, x) \in \mathsf{DomEncRel}$ by hypothesis
$C = \mathsf{EncRel}(\mathsf{epk}, x)$
creates a NIZK proof $\pi$ for $C$
**return** $\left((\alpha_i \sqcup i)_{i\in S}, \mathsf{proj}_{\mathcal{L}}(C), \pi\right)$

$\mathsf{Open}\left(\mathsf{acpar}, \mathsf{ipk}, \mathsf{osk}, \mathrm{pt}\right)$

$c :=$ the cipher included in pt
$\mathrm{id} \leftarrow \mathsf{Dec}(\mathsf{esk}, c)$
**return** id

$\mathsf{VerifyCred}\left(\mathsf{acpar}, \mathsf{ipk}, \mathsf{cred} = \left(\mathrm{id}, (\alpha_i)_{i\in R}, (\sigma_i)_{i\in R}\right)\right)$

**for** $i \in R$:
  **if** $\mathsf{SignVf}(\mathsf{spk}, \sigma_i, \alpha_i \sqcup i) = 0$ **then return** 0
**return** 1

$\mathsf{VerifyPt}\left(\mathsf{acpar}, \mathsf{ipk}, \mathsf{opk}, \mathrm{pt}\right)$

Verify the proof $\pi$
**if** the proof is valid **then return** 1
**else return** 0

**Figure 2.1** – General construction of Anonymous Credentials

- For simplicity, we suppose that $\mathsf{DomEncRel} = \mathsf{Pk}_{\mathsf{enc}} \times \mathcal{R}_{\mathsf{clear}}$. In the practical instantiation of Section 2.5, the keygens for Issuer and Opener will be made such that the keys ipk and opk share an element (this is why $\mathsf{OKGen}$ have ipk in input), $\mathsf{DomEncRel}$ will thus only contain the couples such that ipk and opk contain this element.

The complete scheme can be found in Figure 2.1.

### 2.4.3 Sketch of proofs

We now informally show why this is a secure anonymous credential scheme, for each of the security properties explained in Section 2.4.1 and that are formally defined in [BCN18, Appendix G.1].

- **Anonymity:** We can replace the NIZK proofs by simulations and the ciphertext by the encryption of any fixed clear message. This shows that $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{AnonCredAC}} \leq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{INDCPA}} + \mathsf{Adv}_{\mathcal{A}}^{\mathsf{NIZKAC}}$ where $\mathsf{AnonCredAC}$ is the anonymity game of the anonymous credential, $\mathsf{INDCPA}$ the IND-CPA game of the encryption scheme $(\mathsf{EncKeyGen}, \mathsf{Enc}, \mathsf{Dec})$ and $\mathsf{NIZKAC}$ the game of distinction of a NIZK proof and its simulation.
- **Unforgeability:** If a user can forge with a sufficiently good advantage an anony-

mous credential, one can extract a witness from the proof and use the function $\bar{\epsilon}_{\mathsf{relation}}$ to obtain an element $\left(\left(\overline{\mathrm{id}}, \mathsf{spk}, R, (\alpha_i)_{i\in R}\right), \left((\overline{\mathrm{id}}_i)_{i\in R}, (\sigma_i)_{i\in R}\right)\right) \in \overline{\mathcal{R}}_{\mathsf{clear}}$. The conditions for winning of the game then imply that one of the signatures $\sigma_i$ is a forgery for the relaxed signature scheme.

- **Traceability:** If a user can forge with a sufficiently good advantage an anonymous credential, opened in an element $\tilde{\mathrm{id}}$ that is not the relaxation of the identity of the collusion (note that $\tilde{\mathrm{id}}$ can even be equal to $\perp$), one can extract a witness from the proof and use the function $\bar{\epsilon}_{\mathsf{relation}}$ to obtain an element $\left(\left(\overline{\mathrm{id}}, \mathsf{spk}, R, (\alpha_i)_{i\in R}\right), \left((\overline{\mathrm{id}}_i)_{i\in R}, (\sigma_i)_{i\in R}\right)\right) \in \overline{\mathcal{R}}_{\mathsf{clear}}$. Moreover, the compatibility of the decryption algorithm and the definition of the witness of $\overline{\mathcal{R}}_{\mathsf{cipher}}$ implies that $\tilde{\mathrm{id}}$ is a relaxation of $\overline{\mathrm{id}}$ (in particular, it is not equal to $\perp$). This means that each $\overline{\mathrm{id}}_i$ is a relaxation of an identity that is not in the collusion. The conditions for winning of the game then imply that one of the signatures $\sigma_i$ is a forgery of the relaxed signature scheme.

## 2.5 Instantiation and Parameters

### 2.5.1 Instanciation

For lack of space, we do not present the whole instanciation of the protocol here. It follows the generic construction presented above and we now informally describe the necessary building blocks.

**Relaxed Signature.** As explained in Section 2.2, the relaxed signature scheme that we use is a generalization of the scheme used in [dLS18].

The algorithms are presented in Figure 2.2. They use the same GPV and NTRU trapdoors than the original scheme [dLS18], the NTRU trapdoor being only used in security proofs.

**Verifiable Encryption.** Our scheme is based on the verifiable encryption scheme presented in [LN17] and adapted to [ALS20]. The most interesting part is the $\mathsf{EncRel}$ algorithm (presented in Figure 2.3), due to our new formalism for verifiable encryption. As shown in Section 2.2, this algorithm builds upon the verifiable encryption scheme from [LN17] and the commitment from [dLS18], in order to obtain a short element **comsign** satisfying Equation (2.2).

Note that according to the way the one-shot verifiable encryption is used in the $\mathsf{EncRel}$ algorithm, it does not encrypt the message $m$ but the random coin $\mathbf{r}$ used to commit to $m$. This is completely equivalent, since one only needs to recover $\mathbf{r}$ by using the decryption algorithm and then open the commitment to get the message $m$.

Furthermore, the decryption algorithm (presented in Figure 2.4) decrypts as in the scheme presented in [LN17] (adapted to the formalism of [ALS20]), then uses the decrypted value to open the commitment $\mathbf{t}$ and obtain the message.

$\mathsf{SignKeyGen}()$

$a_3 \leftarrow\!\$\, \mathcal{R}_{q_2}$
$\mathbf{a} \leftarrow\!\$\, \mathcal{R}_{q_2}^2$
$\mathbf{R} = \begin{pmatrix} r_1 & r_2 \\ e_1 & e_2 \end{pmatrix} \leftarrow\!\$\, S_1^{2\times 2} \subset \mathcal{R}^{2\times 2}$
$\mathbf{b} = [b_1\ b_2] \leftarrow \mathbf{a}^T\mathbf{R}$
$\mathsf{spk} \leftarrow (\mathbf{a}^t, \mathbf{b}^t, a_3) \in \mathcal{R}_{q_2}^2 \times \mathcal{R}_{q_2}^2 \times \mathcal{R}_{q_2}$
$\mathsf{ssk} \leftarrow \mathbf{R} \in \mathcal{R}^{2\times 2}$
**return** $(\mathsf{spk}, \mathsf{ssk})$

$\mathsf{Sign}(\mathsf{spk}, (m, \alpha))$

$\mathbf{s}_3 \leftarrow \mathcal{D}^2_{\mathcal{R},\sigma_2^{\mathsf{sign}}}$
$\mathbf{g}^T = [1 \quad \delta]$ // recall that we note $\delta = \lceil \sqrt{q_2} \rceil$
Using GPV trapdoor, samples
$(\mathbf{s}_1, \mathbf{s}_2) \leftarrow D^2_{\mathcal{R},\sigma_1^{\mathsf{sign}}} \times D^2_{\mathcal{R},\sigma_1^{\mathsf{sign}}}$
such that:

$$\left[\mathbf{a}^T \mid \mathbf{b}^T + m\mathbf{g}^T \mid (1 \quad a_3)\right] \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \end{bmatrix} = \mathcal{H}(\alpha)$$

$\mathsf{SignVf}\,(\mathsf{spk}, (m, \alpha), \sigma)$

// Verification of the signature of a message
// we note $\sigma = (\mathbf{s}_{m,\alpha,1}, \mathbf{s}_{m,\alpha,2}, \mathbf{s}_{m,\alpha,2})$
$\mathbf{g}^T = [1 \quad \delta]$ // recall that we note $\delta = \lceil \sqrt{q_2} \rceil$
**if** :

$$\left[\mathbf{a}^T \mid \mathbf{b}^T + m\mathbf{g}^T \mid (1 \quad a_3)\right] \begin{bmatrix} \mathbf{s}_{m,\alpha,1} \\ \mathbf{s}_{m,\alpha,2} \\ \mathbf{s}_{m,\alpha,3} \end{bmatrix} = \mathcal{H}(\alpha)$$

$\wedge\ \|(\mathbf{s}_{m,\alpha,1}, \mathbf{s}_{m,\alpha,2})\| \leq 2B_1^{\text{sign}}$
$\wedge\ \|\mathbf{s}_{m,\alpha,3}\| \leq 2B_2^{\text{sign}}$ **then**
**return** 0
**else**
**return** 1

$\mathsf{SignVf}\,(\mathsf{spk}, ((m, \tilde{c}_\iota), \alpha), \bar{\sigma})$

// Verification of the signature of a relaxed message
// we note $\bar{\sigma} = \left(\mathbf{s}_{m,\alpha,1,\iota}, \mathbf{s}_{m,\alpha,2,\iota}, \mathbf{s}_{m,\alpha,3,\iota}\right)_{\iota\in\mathsf{I}}$
$\mathbf{g}^T = [1 \quad \delta]$ // recall that we note $\delta = \lceil \sqrt{q_2} \rceil$
**if** $\forall \iota \in \mathsf{I}$:

$$\left[\mathbf{a}^T \mid \mathbf{b}^T + m\mathbf{g}^T \mid (1 \quad a_3)\right] \begin{bmatrix} \mathbf{s}_{m,\alpha,1,\iota} \\ \mathbf{s}_{m,\alpha,2,\iota} \\ \mathbf{s}_{m,\alpha,3,\iota} \end{bmatrix} = \tilde{c}_\iota \mathcal{H}(\alpha)$$

$\wedge\ \|\tilde{c}_\iota\|_1 \leq 4\kappa^2$
$\wedge\ \|(\mathbf{s}_{m,\alpha,1,\iota}, \mathbf{s}_{m,\alpha,2,\iota})\| \leq 2B_1^{\text{sign}}$
$\wedge\ \|\mathbf{s}_{m,\alpha,3,\iota}\| \leq 2B_2^{\text{sign}}$
$\wedge\ \tilde{c}_\iota$ invertible modulo $\phi_\iota$ **then**
**return** 0
**else**
**return** 1

**Figure 2.2** – Signature algorithms

For a unique attribute, the relations $\mathcal{R}_{\mathsf{clear}}$ et $\overline{\mathcal{R}}_{\mathsf{clear}}$ are defined as follows:

$$\begin{aligned}
\mathcal{R}_{\mathsf{clear}} = \Big\{ & \big( \big(m, \alpha, \mathsf{spk} = (\mathbf{a}^t, \mathbf{b}^t, a_3)\big), \sigma\big) \\
& \in \mathbb{Z}_{q_2} \times \mathsf{Attributes} \times \mathsf{Pk}_{\mathsf{sign}} \times \mathcal{R}_{q_2}^2 \times \mathcal{R}_{q_2}^2 \times \mathcal{R}_{q_2}^3 : \\
& \mathsf{SignVf}(\mathsf{spk}, (m, \alpha), \sigma) = 1 \Big\} \\
\overline{\mathcal{R}}_{\mathsf{clear}} = \Big\{ & \Big( \big( (m, \alpha, (\bar{c}_\iota)_{\iota \in \mathsf{I}}), \mathsf{spk} \big), \big( ((\bar{m}, (\tilde{c}_\iota)_{\iota \in \mathsf{I}}), \alpha), (\sigma_\iota)_{\iota \in \mathsf{I}} \big) \Big) \\
& \in \bar{\mathcal{C}}^{\mathsf{I}} \times \mathbb{Z}_{q_2} \times \mathsf{Attributes} \times \mathsf{Pk}_{\mathsf{sign}} \times \big(\mathcal{R}_{q_2}^2 \times \mathcal{R}_{q_2}^2 \times \mathcal{R}_{q_2}^3\big)^{\mathsf{I}} : \\
& \forall \iota \in \mathsf{I} \colon \mathsf{SignVf}(\mathsf{spk}, (\bar{m}, (\tilde{c}_\iota)_{\iota \in \mathsf{I}}), \alpha), (\sigma_\iota)_{\iota \in \mathsf{I}}) = 1 \wedge m = \bar{m} \Big\}
\end{aligned}$$

We focus on the algorithms $\mathsf{KeyGen}$ and $\mathsf{EncRel}$ since the other algorithms are based on [LN17] and adapted to [ALS20].

We define $\mathsf{DomEncRel} \subset \mathsf{Pk}_{\mathsf{enc}} \times \mathcal{R}_{\mathsf{clear}}$ as the subset of public keys of encryption and relation elements that share the same element $a_3$.

**$\mathsf{EncKeyGen}$**

$a_1, a_2 \leftarrow\$\, \mathcal{R}_{q_1}$
$a_3 \leftarrow\$\, \mathcal{R}_{q_2}$
$\mathbf{A}_1 \leftarrow \begin{pmatrix} 1 & a_1 & a_2 \end{pmatrix}$
$\mathbf{A}_2 \leftarrow \begin{pmatrix} 0 & 1 & a_3 \end{pmatrix}$
$\mathbf{A} \leftarrow \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{pmatrix}$
$a \leftarrow\$\, \mathcal{R}_Q$
$\mathbf{s} \leftarrow\$\, S_1^3$
$\mathbf{e} \leftarrow\$\, S_1^3$
$\mathbf{b} \leftarrow a\mathbf{s} + \mathbf{e} \in \mathcal{R}_Q^3$
$\mathsf{epk} \leftarrow (\mathbf{A}, a, \mathbf{b})$
$\mathsf{esk} \leftarrow \mathbf{s}$
**return** $(\mathsf{epk}, \mathsf{esk})$

**$\mathsf{EncRel}(\mathrm{epk}, x \in \mathcal{R}_{\mathsf{clear}})$**

// we note $x = \big(\big(m, \alpha, \mathsf{spk} = (\mathbf{a}^t, \mathbf{b}^t, a_3)\big), \sigma\big)$
$\omega \leftarrow\$\, S_1$
$r, e_1 \leftarrow\$\, S_1$
$\mathbf{e}_2 \leftarrow\$\, S_1^3$
$\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) \leftarrow\$\, S_1^3$
$\mathbf{r}' = (\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{r}'_3) \leftarrow\$\, S_1^3$
$\mathrm{enc}_1 \leftarrow p(ar + e_1)$
$\mathbf{enc}_2 \leftarrow p(\mathbf{b}\omega + \mathbf{e}_2) + \mathbf{r}$
$\mathbf{t} \leftarrow \mathbf{A}\mathbf{r} + \begin{bmatrix} 0 \\ m \end{bmatrix} \mod \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$
$\mathbf{t}' \leftarrow \mathbf{A}\mathbf{r}' + \begin{bmatrix} 0 \\ \delta m \end{bmatrix} \mod \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$
// we note $\sigma = (\mathbf{s}_{m,\alpha,1}, \mathbf{s}_{m,\alpha,2}, s_{m,\alpha,3})$
$\bar{\mathbf{r}} = (\mathbf{r}_2, \mathbf{r}_3); \bar{\mathbf{r}}' = (\mathbf{r}'_2, \mathbf{r}'_3)$
$\mathbf{comsign} \leftarrow \begin{bmatrix} \mathbf{s}_{m,\alpha,1} \\ \mathbf{s}_{m,\alpha,2} \\ \mathbf{s}_{m,\alpha,3} - [\bar{\mathbf{r}}\ \ \bar{\mathbf{r}}']\mathbf{s}_{m,\alpha,2} \end{bmatrix} \in {\mathcal{R}_{q_2}}^{7\times 1}$
$c \leftarrow (\mathbf{t}, \mathbf{t}', \mathrm{enc}_1, \mathbf{enc}_2)$
$w_c \leftarrow (m, \alpha, \mathbf{r}, \mathbf{r}', r, e_1, \mathbf{e}_2, \mathbf{comsign})$
**return** $((\mathrm{epk}, c, \mathsf{spk} = (\mathbf{a}^t, \mathbf{b}^t, a_3)), w_c)$

**Figure 2.3** – Encryption and encryption of relations algorithms

$\mathsf{Dec}\Big(\text{esk} = \mathbf{s}, \text{epk} = (\mathbf{A}, a, \mathbf{b}), c = (\mathbf{t}, \mathbf{t}', \text{enc}_1, \mathbf{enc}_2)\Big)$

$\mathbf{tmp\_dec} \leftarrow (\mathbf{enc}_2 - \text{enc}_1\mathbf{s})$
**if** $\|\mathbf{tmp\_dec}\|_\infty < Q/4\kappa$ **then** // if no $\bar{c}$ is needed
  **return** $(\mathbf{tmp\_dec}, t_2 - \mathbf{A}_2\,\mathbf{tmp\_dec})$
$J \leftarrow \varnothing$
**while** $I \neq J$ **do**
  $\iota^* \leftarrow J - I$ // Can be taken in any prefered order
  c_found $\leftarrow$ **false**
  **while** c_found $\leftarrow$ **false**
    $c' \leftarrow\$\, \mathcal{C}$ such that $c \neq c' \mod \phi_{\iota^*}$
    $\bar{c} \leftarrow c - c'$
    **if** $\|\bar{c}\,\mathbf{tmp\_dec}\|_\infty \leq Q/4\kappa$ **then**
      c_found $\leftarrow$ **true**
  **endwhile**
set_index $\leftarrow \{j \notin J : \bar{c} \neq 0 \mod \phi_j\}$
// we will use the $\bar{c}$ for each $i \notin J$ where it works
**for** $\iota \in$ set_index
  $\bar{c}_\iota \leftarrow \bar{c}$
  $\mathbf{r}_\iota \leftarrow \bar{c}\,\mathbf{tmp\_dec}$
$J \leftarrow J \cup$ set_index
**endwhile**
**construct** $\mathbf{r} \in \mathcal{R}$ such that :
  $\|\mathbf{r}\|_\infty \leq q_1 q_2 Q/2$
  $\forall \iota \in \mathsf{I}$:
  $\mathbf{r} \mod \phi_\iota \mod q_1 q_2 Q = \dfrac{\mathbf{r}_\iota}{\bar{c}_\iota} \mod \phi_\iota \mod q_1 q_2 Q$
$m \leftarrow t_2 - \mathbf{A}_2\mathbf{r}$
**return** $(\mathbf{r}, (m, (c_\iota)_{\iota \in \mathsf{I}}))$

**Figure 2.4** – Decryption algorithm

As explained in Section 2.2, the associated NIZK proof then is the union of two NIZK proofs, linked by the commitment:

- a proof of knowledge of a short element **comsign** satisfying the non-homogeneous linear equation (2.2),
- a proof of knowledge of an element $r$ which is the cleartext and which opens the commitment on a message $m$.

These two proofs were already used in [dLS18] but we adapted them to [ALS20] and also took into account the more general definition of the signature.

## 2.5.2 Parameters

The security of the anonymous credential relies on the security of the primitives it uses, that are the signature and verifiable encryption scheme, defined in Section 2.5.1.

We list in this section the main conditions needed for these primitives and the choices we made. The meaning of the parameters, and the values we choose, are indicated in Table 2.6.

First, a forgery of the signature (with bounds $B_1^{\text{sign}}$, $B_2^{\text{sign}}$) led to a solution of $\mathsf{MSIS}_{q_2,1,4,B^{\text{forgsign}}}$ with

$$B^{\text{forgsign}} = 8\kappa^2\sqrt{d}\,\sigma_1^{\mathsf{sign}} + B_1^{\text{sign}} + 24\sqrt{2}\,d\kappa^2\sigma_1^{\mathsf{sign}} + 3\sqrt{d}B_1^{\text{sign}} + 4\kappa^2\sqrt{6d}\,\sigma_2^{\mathsf{sign}} + B_2^{\text{sign}}$$

The security of the signature scheme thus needs $\mathsf{MSIS}_{q_2,1,4,B^{\text{forgsign}}}$ to be hard, as well as $\mathsf{NTRU}_{q_2,\sigma_2^{\mathsf{sign}}}$ and $\mathsf{MLWE}_{q_2,1,\sigma_1^{\mathsf{sign}}}$, and a few other conditions.

Then, the security of the verifiable encryption requires that

$$\begin{aligned} 4\kappa N_1^{\mathsf{comsign}} &\leq B_1^{\mathrm{sign}} \\ 4\kappa N_1^{\mathsf{comsign}} + 8N_2^{\mathsf{comsign}} B^{\mathrm{com}} &\leq B_2^{\mathrm{sign}} \end{aligned} \tag{2.4}$$

As for the security of the auxiliary commitment scheme used by the verifiable encryption, the binding property needs $\mathsf{MSIS}_{q_1,1,3,8\kappa B^{\mathrm{com}}}$ to be hard. The hiding property behaves the same way as in [dLS18], so that we need $\mathsf{MLWE}_{q_1,X,d}$ and $\mathsf{MLWE}_{q_2,X,d}$ to be hard for a little number X of samples, with errors sampled in $S_1^d$. For the IND-CPA propriety of the underlying encryption scheme of the verifiable encryption, we need, as shown in [LN17], $\mathsf{MLWE}_{Q,1,s}$ hard with a little $s$, and this is already true if $\mathsf{MLWE}$ is hard with $q_2 > Q$.

For the first part of the NIZK proof, using the notations $B^{\mathrm{s}_1} = 2\sqrt{2d}s$ and $B^{\mathrm{s}_2} = \sqrt{6}\left(\sqrt{d}r + 4ds\right)$, we need $\mathsf{MSIS}_{q_1,1,3,4B^{\mathrm{com}}}$ to be hard and also the following conditions fulfilled:

$$\begin{aligned} &\sigma_1^{\mathsf{sign}} \geq 11\kappa\sqrt{17d} \\ &B^{\mathrm{com}} \geq \sqrt{6d}\,\sigma_1^{\mathsf{sign}} \quad N_2^{\mathsf{dec}} \geq \sqrt{10d}\;\sigma_1^{\mathsf{sign}} \\ &N_{1,\infty}^{\mathsf{dec}} \geq 12\sigma_1^{\mathsf{sign}} \quad N_{2,\infty}^{\mathsf{dec}} \geq 12\sigma_1^{\mathsf{sign}} \\ &2pN_{2,\infty}^{\mathsf{dec}}(1+6d) + p/4\kappa \leq Q/4\kappa \\ &8\kappa N_{1,\infty}^{\mathsf{dec}} \leq p \end{aligned}$$

For the second part of the NIZK proof, the conditions are as follows:

$$\begin{aligned} &\sigma_2^{\mathsf{nizk}} \geq 11\kappa B^{\mathrm{s}_1} = 22\kappa\sqrt{2d}\,\sigma_1^{\mathsf{sign}} \\ &\sigma_3^{\mathsf{nizk}} \geq 11\kappa B^{\mathrm{s}_2} = 11\sqrt{6}\,\kappa\left(\sqrt{d}\sigma_2^{\mathsf{sign}} + 4d\sigma_1^{\mathsf{sign}}\right) \\ &N_1^{\mathsf{comsign}} \geq \sqrt{8d}\;\sigma_2^{\mathsf{nizk}} \quad N_2^{\mathsf{comsign}} \geq \sqrt{6d}\;\sigma_3^{\mathsf{nizk}} \end{aligned}$$

To ensure the correctness of the zero knowledge proofs, we need to consider the inversion of the challenges in four rings: $\mathcal{R}_p$, $\mathcal{R}_{q_1}$, $\mathcal{R}_{q_2}$, $\mathcal{R}_Q$.

We use [dLS18, Lemma 2.1] with $k = 2$ to ensure that each $\bar{c} \in \bar{\mathcal{C}}$ is invertible in $\mathcal{R}_p$. We thus ask $p = 5 \mod 8$.

We also use the framework of [ALS20] for $Q, q_1$ and $q_2$, with multiple computations of answers by the prover, depending on the NIZK proofs. This is why we ask the equation (2.3).

Finally, all these conditions lead to the table of parameters presented in Table 2.6. The precise values are chosen using the lattice estimator [APS15] to assess the security of the computational problems based on lattices.

## 2.6 Implementation

In order to assess the performance of our anonymous credential protocol, we have implemented in C its most time and bandwidth critical operations: the presentation

token generation and the presentation token verification. The source code of our implementation is attached to the submission for easy reproducibility.

**Polynomial arithmetic.** The NTT is a fast Fourier transform algorithm commonly used in lattice-based cryptography to efficiently compute polynomial products over $\mathbb{Z}_q[X]/(X^d + 1)$. The NTT algorithm has a time complexity of $O(d \log d)$, which is significantly faster than other known polynomial multiplication algorithms.

Our moduli are chosen such that there exists a primitive $2\ell$-th root of unity modulo $q \in \{Q, q_1, q_2\}$. Thus, we use the *partial-splitting* Number Theoretic Transform (NTT) to speed up polynomial multiplications in our implementation. Cooley-Tukey butterflies are used in the forward transform and Gentleman-Sande butterflies in the inverse transform, and most of the polynomials are represented in NTT.

**Modular arithmetic.** Our code uses the low-level GMP [Gt20] functions for single and multi-precision modular arithmetic. Low-level GMP functions manipulate *limbs* to perform basic arithmetic operations. They are highly optimized and then significantly improve the efficiency of our implementation.

**Pseudo-random generator.** We use the NIST AES256 CTR DRBG software[1] as pseudo-random generator in our implementation. The AES256 CTR DRBG is a deterministic random bit generator, based on the AES block cipher, specified in NIST Special Publication 800-90A[2].

The NIST AES256 CTR DRBG software calls the OpenSSL AES implementation, which in our architecture uses the AES-NI instructions. AES-NI instructions significantly reduce the random generation time compared to non-hardware-based implementations of AES while providing a high level of security for our anonymous credential system.

**Gaussian sampling.** We use the DGS Gaussian sampler [AW18] implementation to generate samples from Gaussian distributions. More precisely we use the `DGS_DISC_SIGMA2_LOGTABLE` algorithm which is based on [DDLL13, Algorithm 12].

It is worth noting that unlike the rest of our implementation, DGS uses the random number generators from GMP [Gt20] and MPFR [The23]. These generators are not as fast as the AES256 CTR DRBG used in the rest of our implementation.

**Performance.** We have performed timing experiments with our implementation on a AMD® Ryzen 7 5800h CPU. The results are presented in Table 2.1 for the generation and in Table 2.2 for the verification. They include the minimum, median and mean number of CPU cycles and time of 1000 executions needed for each operation. The code was compiled with gcc 9.4.0.

The significant gap between the minimum and median time for the presentation generation operation is explained by the rejection sampling step at the end of the generation which requires to restart the generation procedure in case of rejection.

[1] https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/example-files/source-code-files-for-kats.zip

[2] http://dx.doi.org/10.6028/NIST.SP.800-90Ar1

**Table 2.1** – Performance of the presentation generation.

| **Presentation token generation** | | |
|---|---|---|
| | Time (s) | Cycles (×1000) |
| Lowest | 0.67297 | 1216651 |
| Median | 1.28102 | 4091981 |
| Average | 1.84276 | 5887531 |

**Table 2.2** – Performance of the presentation verification.

| **Presentation token verification** | | |
|---|---|---|
| | Time (s) | Cycles (×1000) |
| Lowest | 0.16554 | 528951 |
| Median | 0.17174 | 548732 |
| Average | 0.17187 | 549226 |

**Call-graph profiling.** Our code has been profiled using callgrind [Jos22] in order to obtain the CPU time spent in each funstion. Results are presented in Table 2.3 for the presentation generation, and in Table 2.4 for the presentation verification.

**Table 2.3** – Generation call-graph profiling.

| **Presentation token generation** | |
|---|---|
| Function | Time |
| Gaussian sampling | 39.36 % |
| NTT forward transform | 16.93 % |
| NTT inverse transform | 14.73 % |
| Rejection sampling | 6.84 % |
| Golomb-Rice encoding | 4.72 % |

For the generation of a presentation token, one can observe that Gaussian sampling is the most time consuming operation. Actually, 73.61 % of the time spent sampling Gaussian samples corresponds to the generation of the random bits needed for the Gaussian sampling. Thus, an optimization would consist in replacing the DGS pseudo-random number generators by the AES256 CTR DRBG.

Obviously, the verification can be speeded up by removing the compression step, but at the cost of a significant increase in bandwidth usage due to sending uncompressed presentation tokens.

**Golomb-Rice coding.** The Golomb-Rice coding is a lossless data compression method that is particularly well-suited for compressing Gaussian samples as explained

**Table 2.4** – Verification call-graph profiling.

| **Presentation token verification** | |
|---|---|
| Function | Time |
| NTT forward transform | 50.90 % |
| Golomb-Rice encoding | 20.52 % |
| NTT multiplication | 20.35 % |
| NTT inverse transform | 5.02 % |



in [ETWY22]. The basic idea behind Golomb-Rice coding for integers is to use a variable-length unary prefix code to represent the quotient and a fixed-length binary code to represent the remainder of the integer divided by a chosen parameter $\sigma$, the standard deviation of the distribution in our case.

More specifically, suppose we have a discrete Gaussian sample $z$, and we want to compress it using Golomb-Rice coding with parameter $\sigma$. We first compute the quotient $c$ and remainder $r$ of $z$ divided by $\sigma$, i.e., $z = c\sigma + r$, where $c$ and $r$ are integers and $0 \leq r < \sigma$. We then use a variable-length prefix code to represent $c$ and separately represent $r$ in binary.

The code for $z$ is constructed as follows. We first define a unary code, which represents a non-negative integer $n$ as a sequence of $n$ 1's followed by a 0. For example, the unary code for 3 is 1110. We then represent the absolute value of $c$ using its unary code and $r$ or $-r$, according to the sign of $c$, using its binary representation with a fixed number of bits $b$, which depends on the value of $\sigma$. The value of $b$ is chosen so that $2^{b+1} \geq \sigma$, and $b$ is typically chosen to be the smallest value that satisfies this condition.

**Bandwidth usage.** The presentation token is the only data that is sent by a user to prove the possession of a credential with the right attribute values. Without encoding or compressing, the size of a presentation token in our protocol is about 3.7 MB. For this reason, presentation tokens are compressed in our implementation with the Golomb-Rice coding described above. The precise minimum, maximum and average sizes for a presentation token are given in Table 2.5.

**Table 2.5** – Size of a presentation token.

| **Presentation token size in bytes** | | |
|---|---|---|
| Lowest | Average | Highest |
| 1 922 876 | 1 923 044 | 1 923 107 |

Gaussian elements and the trinary challenge benefit greatly from the Golomb-Rice compression. Using this, the size of a compressed presentation token in less than 2 MB with little variation due to those elements that follow non-uniforme probability distributions.

| Element | Description | Value |
|---|---|---|
| d | such that $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$ | 8192 |
| $\theta$ | number of irreducible divisor of $X^d + 1$ in $\mathcal{R}_q$ for $q \in \{Q, p_1, p_2\}$ | 2048 |
| $\tau_{\mathsf{sign}}$ | number of questions asked for each challenges in the second part of the NIZK proof | 1 |
| $\tau_{\mathsf{enc}}$ | number of questions asked for each challenges in the first part of the NIZK proof | 1 |
| $\sigma_1^{\mathsf{sign}}$ | standard deviation for the GPV trapdoor | $6\sqrt{dq_2} \approx 2^{84.5}$ |
| $\sigma_2^{\mathsf{sign}}$ | standard deviation for the NTRU trapdoor | $21.17\sqrt{q_2} \approx 2^{76.7}$ |
| $p$ | prime modulus for clears in verifiable encryption | $\approx 2^{75.4}$ (p33, 8.1) |
| $Q$ | prime modulus for ciphertexts in verifiable encryption | $\approx 2^{85.1}$ (p33, 8.1) |
| $q_1$ | prime modulus for the upper part of commitment | $q_1 \approx 2^{40.1}$ |
| $q_2$ | prime modulus for the lower part of commitment | $q_2 \approx 2^{150.9}$ |
| $\delta$ | $\lceil \sqrt{q_2} \rceil$ | $q_2 \approx 2^{80}$ |
| $\sigma_1^{\mathsf{nizk}}$ | standard deviation for discrete gaussian in the NIZK proof | $11\kappa\sqrt{17}\, d \approx 2^{20}$ |
| $\sigma_2^{\mathsf{nizk}}$ | standard deviation for discrete gaussian in the NIZK proof | $22\kappa\sqrt{2d}\, \sigma_1^{\mathsf{sign}} \approx 2^{103.9}$ |
| $\sigma_3^{\mathsf{nizk}}$ | standard deviation for discrete gaussian in the NIZK proof | $11\sqrt{6}\kappa \left(\sqrt{d}\sigma_2^{\mathsf{sign}} + 4d\sigma_1^{\mathsf{sign}}\right) \approx 2^{112.2}$ |
| $N_1^{\mathsf{comsign}}$ | bound used in NIZK proof | $88\kappa ds$ |
| $N_2^{\mathsf{comsign}}$ | bound used in NIZK proof | $66\kappa d \left(\sigma_2^{\mathsf{sign}} + 4\sqrt{d}\, \sigma_1^{\mathsf{sign}}\right)$ |
| $B_1^{\mathsf{sign}}$ | bound used in verification of signatures | $88\kappa d\sigma_1^{\mathsf{sign}}$ |
| $B_1^{\mathsf{sign}}$ | bound used in verification of signatures | $11\kappa\sqrt{102}\, d$ |
| $N_2^{\mathsf{dec}}$ | bound used in NIZK proof | $11\kappa\sqrt{170}\, d$ |
| $B^{\mathrm{com}}$ | bound used in verification of commitments and in NIZK proof. | |
| $\mathcal{C}$ | space of challenges | $S_1$ |
| $\bar{\mathcal{C}}$ | $\{c - c' : (c, c') \in \mathcal{C}, c \neq c'\}$ | |
| $\rho$ | probability used to define the distribution of the choice of challenges | 63/64 |
| $\mathcal{X}(\rho)$, | distribution on $\mathcal{C}$: each coefficient is chosen independently 0 with a probability $\rho$, -1 and 1 with a probability $(1 - \rho)/2$ | |
| $\kappa$ | bound such that the distribution $\mathcal{X}(\rho)$ output elements $c \in \mathcal{C}$ such that $\|c\|_1 \leq \kappa$ with an overwhelming probability | 240 |
| $\mathcal{H}$ | hash function for the signature. | $\mathcal{H} : \{0,1\}^* \to \mathcal{R}_{q_2}$ |
| $\mathcal{H}^{ZK}$ | hash function for the NIZK proofs | $\mathcal{H}^{ZK} : \{0,1\}^* \to \mathcal{C}$ |

**Table 2.6** – Parameters of the Anonymous Credential Scheme

# Références

[ALS20] Thomas Attema, Vadim Lyubashevsky et Gregor Seiler. *Practical Product Proofs for Lattice Commitments*. In : *CRYPTO 2020, Part II*. Sous la dir. de Daniele Micciancio et Thomas Ristenpart. T. 12171. LNCS. Springer, Heidelberg, août 2020, p. 470-499 (cf. p. 17-21, 23, 26, 28-30).

[APS15] Martin R. Albrecht, Rachel Player et Sam Scott. *On the concrete hardness of Learning with Errors*. In : *J. Math. Cryptol.* 9.3 (2015), p. 169-203. (Cf. p. 17, 30).

[AW18] Martin R. Albrecht et Michael Walter. "dgs, Discrete Gaussians over the Integers". Available at https://bitbucket.org/malb/dgs. 2018. (Cf. p. 31).

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss et Anna Lysyanskaya. *P-signatures and Noninteractive Anonymous Credentials*. In : *TCC 2008*. Sous la dir. de Ran Canetti. T. 4948. LNCS. Springer, Heidelberg, mars 2008, p. 356-374 (cf. p. 16).

[BCN18] Cecilia Boschini, Jan Camenisch et Gregory Neven. *Relaxed Lattice-Based Signatures with Short Zero-Knowledge Proofs*. In : *ISC 2018*. Sous la dir. de Liqun Chen, Mark Manulis et Steve Schneider. T. 11060. LNCS. Springer, Heidelberg, sept. 2018, p. 3-22 (cf. p. 16, 17, 21-23, 25).

[BDL+18] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner et Chris Peikert. *More Efficient Commitments from Structured Lattice Assumptions*. In : *SCN 18*. Sous la dir. de Dario Catalano et Roberto De Prisco. T. 11035. LNCS. Springer, Heidelberg, sept. 2018, p. 368-385 (cf. p. 18).

[Bra00] Stefan A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates : Building in Privacy*. Cambridge, MA, USA : MIT Press, 2000. isbn : 0262024918 (cf. p. 16).

[CDHK15] Jan Camenisch, Maria Dubovitskaya, Kristiyan Haralambiev et Markulf Kohlweiss. *Composable and Modular Anonymous Credentials : Definitions and Practical Constructions*. In : *ASIACRYPT 2015, Part II*. Sous la dir. de Tetsu Iwata et Jung Hee Cheon. T. 9453. LNCS. Springer, Heidelberg, nov. 2015, p. 262-288 (cf. p. 16).

[Cha85] David Chaum. *Security Without Identification : Transaction Systems to Make Big Brother Obsolete*. In : *Commun. ACM* 28.10 (1985), p. 1030-1044. (Cf. p. 15).

[CL01] Jan Camenisch et Anna Lysyanskaya. *An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation*. In : *EUROCRYPT 2001*. Sous la dir. de Birgit Pfitzmann. T. 2045. LNCS. Springer, Heidelberg, mai 2001, p. 93-118 (cf. p. 16).

[CL02] Jan Camenisch et Anna Lysyanskaya. *Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials*. In : *CRYPTO 2002*. Sous la dir. de Moti Yung. T. 2442. LNCS. Springer, Heidelberg, août 2002, p. 61-76 (cf. p. 16).

[CL03] Jan Camenisch et Anna Lysyanskaya. *A Signature Scheme with Efficient Protocols*. In : *SCN 02*. Sous la dir. de Stelvio Cimato, Clemente Galdi et Giuseppe Persiano. T. 2576. LNCS. Springer, Heidelberg, sept. 2003, p. 268-289 (cf. p. 16).

[CL04] Jan Camenisch et Anna Lysyanskaya. *Signature Schemes and Anonymous Credentials from Bilinear Maps*. In : *CRYPTO 2004*. Sous la dir. de Matthew Franklin. T. 3152. LNCS. Springer, Heidelberg, août 2004, p. 56-72 (cf. p. 16).

[Cv91] David Chaum et Eugène van Heyst. *Group Signatures*. In : *EUROCRYPT'91*. Sous la dir. de Donald W. Davies. T. 547. LNCS. Springer, Heidelberg, avr. 1991, p. 257-265 (cf. p. 16).

[DDLL13] Léo Ducas, Alain Durmus, Tancrède Lepoint et Vadim Lyubashevsky. *Lattice Signatures and Bimodal Gaussians*. In : *CRYPTO 2013, Part I*. Sous la dir. de Ran Canetti et Juan A. Garay. T. 8042. LNCS. Springer, Heidelberg, août 2013, p. 40-56 (cf. p. 31).

[dLS18] Rafaël del Pino, Vadim Lyubashevsky et Gregor Seiler. *Lattice-Based Group Signatures and Zero-Knowledge Proofs of Automorphism Stability*. In : *ACM CCS 2018*. Sous la dir. de David Lie, Mohammad Mannan, Michael Backes et XiaoFeng Wang. ACM Press, oct. 2018, p. 574-591 (cf. p. 17-19, 26, 29, 30).

[ETWY22] Thomas Espitau, Mehdi Tibouchi, Alexandre Wallet et Yang Yu. *Shorter Hash-and-Sign Lattice-Based Signatures*. In : *CRYPTO 2022, Part II*. Sous la dir. d'Yevgeniy Dodis et Thomas Shrimpton. T. 13508. LNCS. Springer, Heidelberg, août 2022, p. 245-275 (cf. p. 33).

[GGM14] Christina Garman, Matthew Green et Ian Miers. *Decentralized Anonymous Credentials*. In : *NDSS 2014*. The Internet Society, fév. 2014 (cf. p. 16).

[Gt20] Torbjörn Granlund et the GMP development team. *GNU MP : The GNU Multiple Precision Arithmetic Library*. 6.2.1. 2020. (Cf. p. 31).

[ILV11] Malika Izabachène, Benoît Libert et Damien Vergnaud. *Block-Wise P-Signatures and Non-interactive Anonymous Credentials with Efficient Attributes*. In : *Cryptography and Coding - 13th IMA International Conference, IMACC 2011, Oxford, UK, December 12-15, 2011. Proceedings*. Sous la dir. de Liqun Chen. T. 7089. Lecture Notes in Computer Science. Springer, 2011, p. 431-450. (Cf. p. 16).

[Jos22] Josef Weidendorfer. *Callgrind : a call-graph generating cache and branch prediction profiler*. 3.20.0. 2022. (Cf. p. 32).

[LN17] Vadim Lyubashevsky et Gregory Neven. *One-Shot Verifiable Encryption from Lattices*. In : *EUROCRYPT 2017, Part I*. Sous la dir. de Jean-Sébastien Coron et Jesper Buus Nielsen. T. 10210. LNCS. Springer, Heidelberg, avr. 2017, p. 293-323 (cf. p. 19, 23, 26, 28, 30).

[LNP22] Vadim Lyubashevsky, Ngoc Khanh Nguyen et Maxime Plançon. *Lattice-Based Zero-Knowledge Proofs and Applications : Shorter, Simpler, and More General*. In : *CRYPTO 2022, Part II*. Sous la dir. d'Yevgeniy Dodis et Thomas Shrimpton. T. 13508. LNCS. Springer, Heidelberg, août 2022, p. 71-101 (cf. p. 17).

[The23] The MPFR team. *GNU MPFR : The Multiple Precision Floating-Point Reliable Library*. 4.2.0. 2023. (Cf. p. 31).

« la cryptologie n'est plus seulement
la science du secret mais aussi la
science de la confiance[a] »

[a] « cryptology is no longer only the science of secrecy, but also the science of trust »

Discours au CNRS –
Jacques Stern

# Formal Verification of a Post-Quantum Signal Protocol with Tamarin

3

The Signal protocol is used by billions of people for instant messaging in applications such as Facebook Messenger, Google Messages, Signal, Skype, and WhatsApp. However, advances in quantum computing threaten the security of the cornerstone of this protocol: the Diffie-Hellman key exchange. There actually are resistant alternatives, called post-quantum secure, but replacing the Diffie-Hellman key exchange with these new primitives requires a deep revision of the associated security proof. While the security of the current Signal protocol has been extensively studied with hand-written proofs and computer-verified symbolic analyses, its quantum-resistant variants lack symbolic security analyses.

In this work, we present the first symbolic security model for post-quantum variants of the Signal protocol. Our model focuses on the core state machines of the two main sub-protocols of Signal: the X3DH handshake, and the so-called *double ratchet* protocol. Then we show, with an automated proof using the Tamarin prover, that instantiated with the Hashimoto-Katsumata-Kwiatkowski-Prest post-quantum Signal's handshake from PKC'21, and the Alwen-Coretti-Dodis KEM-based double ratchet from EUROCRYPT'19, the resulting post-quantum Signal protocol has equivalent security properties to its current classical counterpart.

## 3.1 Introduction

The Signal protocol is divided into two sub-protocols: X3DH [Treb], and the double ratchet protocol [Trea]. The X3DH protocol can be seen as an Authenticated Key Exchange (AKE) protocol. It ensures the authenticity of an initial key shared between two users. It is an asynchronous protocol, which means that there is no need for users to be online at the same time to initialize the protocol. To use the X3DH protocol, each user must first generate a long-term static pair of public and private keys for them to be authenticated, as well as a batch of ephemeral pairs of public and private keys. Both long-term public keys and ephemeral key batches are then stored on an honest intermediate server which acts as a buffer. When Bob wants to start a conversation with Alice, he sends a request to the server, and then receives the Alice's long-term public key and a fresh Alice's ephemeral public key from her batch. These two public keys enable Bob to perform the X3DH handshake protocol by sending a message to Alice, which will enables her to derive their X3DH pre-shared secret key when she is online.

The double ratchet protocol is used to encrypt messages to send through the Signal protocol with an Authenticated Encryption with Associated Data (AEAD) scheme and a session key that is shared between the two parties. The session key is renewed each time a message is sent, using symmetric and asymmetric mechanisms called ratchets. The double ratchet protocol is initialized with the X3DH pre-shared key as session key, and an ephemeral public key from the corresponding batch as public key. Then, to send a message, the public key is used to exchange a fresh secret key, from which the new session key is derived with the output of a one-way function applied to the current session key. In addition, a new ephemeral key pair is generated whose public key is encrypted then sent with the message, using this new session key. This protocol is repeated for each message sent to ensure strong security properties such as forward secrecy and post-compromise recovery against passive adversaries.

The current Signal protocol heavily uses the well-known, flexible, and efficient, but vulnerable to quantum attacks, Diffie-Hellman (DH) key exchange protocol. However, with the threat of upcoming quantum computers, post-quantum alternatives are subject to extensive analysis in order to gain assurance in their security. In 2016, the NIST initiated a process to evaluate and standardize quantum-resistant key-establishment and signature schemes, but all remaining candidates in the key-establishment category are key encapsulation mechanisms (KEMs) like RSA, and not key exchanges like DH. Consequently, the integration of post-quantum KEMs in cryptographic protocols is quite challenging due to the differences between KEMs and DH, which requires some fundamental adjustments to these protocols to maintain the same security guarantees.

Aside from that, the active area of formal protocol verification is increasingly accompanying protocol specifications. Designing cryptographic protocols is known to be hard to get right and hand-written proofs remain highly complex and error-prone. At the design level, automatic verification aims to manage the complexity of security proofs and even reveal subtle flaws or as-yet-unknown attacks as the historic example of the man-in-the-middle attack [Low96]. Efficient automatic verification tools as Tamarin [MSCB13] or ProVerif [Bla16] have been used to analyze large, real-world protocols. For instance, ProVerif has been used to analyze TLS 1.3 [BBK17] and Signal [KBB17] and Tamarin as been used to analyze the 5G AKE protocol [BDH+18] and TLS 1.3 [CHH+17].

**Related Works.** The security of the (EC)DH-based Signal protocol has been extensively studied using hand-written proofs [CCD+20]. Those proofs were completed with a symbolic analysis [KBB17] using the ProVerif prover. Regarding the transition to post-quantum cryptography, there are KEM-based alternatives to the Signal subprotocols X3DH [HKKP21b; BFG+20] (the security properties in [HKKP21b] being closer to that of X3DH, in particular thanks to the encryption of the signature) and double ratchet [ACD19], with hand-written proofs for the same security properties as the current Signal protocol. Such KEM-based protocols can be instantiated with post-quantum KEMs from the NIST competition such as Kyber [ABD+21], which will be the first NIST PQC standard for key-establishment. However, those potential replace-

ments for X3DH and double ratchet have so far lacked computer-verified symbolic analyses which results in a limited trust in these protocols. By contrast, some other protocols, such as WireGuard [HNS+21] and (KEM)TLS [SSW20; CHSW22], already have computer-verified symbolic analyses for post-quantum variants, both using the Tamarin prover.

**Contributions.** We present the first symbolic proof of a post-quantum variant of the Signal protocol. Our model focuses on the core state machines of the two main sub-protocols of this variant: the Hashimoto-Katsumata-Kwiatkowski-Prest post-quantum X3DH handshake [HKKP21b] which we refer to as *PQ-X3DH*, and the Alwen-Coretti-Dodis KEM-based double ratchet [ACD19] that we call *KEM-Double-Ratchet*. Then we show, using the Tamarin prover, that these two protocols meet the same security properties as classical X3DH and double ratchet protocols. In addition, we prove the well-formedness of the two models, which informally means that their behavior is as expected.

Our PQ-X3DH Tamarin symbolic analysis ensures the integrity of the two exchanged messages, the authentication of users, the resistance to unknown key-share attacks and replay attacks, and other properties, such as the weak forward secrecy [Kra05] and the key compromise attack resistance, to mitigate the leak of secret information.

With regard to KEM-Double-Ratchet, our Tamarin model ensures the integrity of all the messages, the forward secrecy, and the post-compromise recovery [ACD19]. It is worth noting that in the particular case of Signal, post-compromise recovery is met only if the adversary is passive during the recovering process. While within the double ratchet protocol two parties can exchange a potentially infinite number of messages, we model only three exchanges, which represents the minimum number of exchanges for each security property to hold. A simple induction argument then enables us to generalize these properties to any number of exchanges. To our knowledge, our formal verification model is the first one that covers the post-compromise recovery security property.

**Outline.** In section 3.2 we present the two sub-protocols of the considered variant of the Signal protocol and their Tamarin model. Then we present in section 3.3 the Tamarin formalism used in our symbolic analysis, the different security properties verified, and the results of our formal verification.

## 3.2 A KEM-Based Signal Protocol

In this section, we describe the KEM-based variant of the Signal protocol that is the subject of our symbolic analysis. As explained in the introduction, the Signal protocol is separated in two sub-protocols providing different functionalities, we respect this separation for this KEM-based variant in order to facilitate its analysis and clearly identify the contribution of each sub-protocol in the security of the whole protocol. The first sub-protocol named PQ-X3DH is used as authenticated key agreement while

the second one named KEM-double-ratchet is used for secure instant messaging by refreshing the session key at each time a message is sent.

### 3.2.1 The PQ-X3DH Protocol

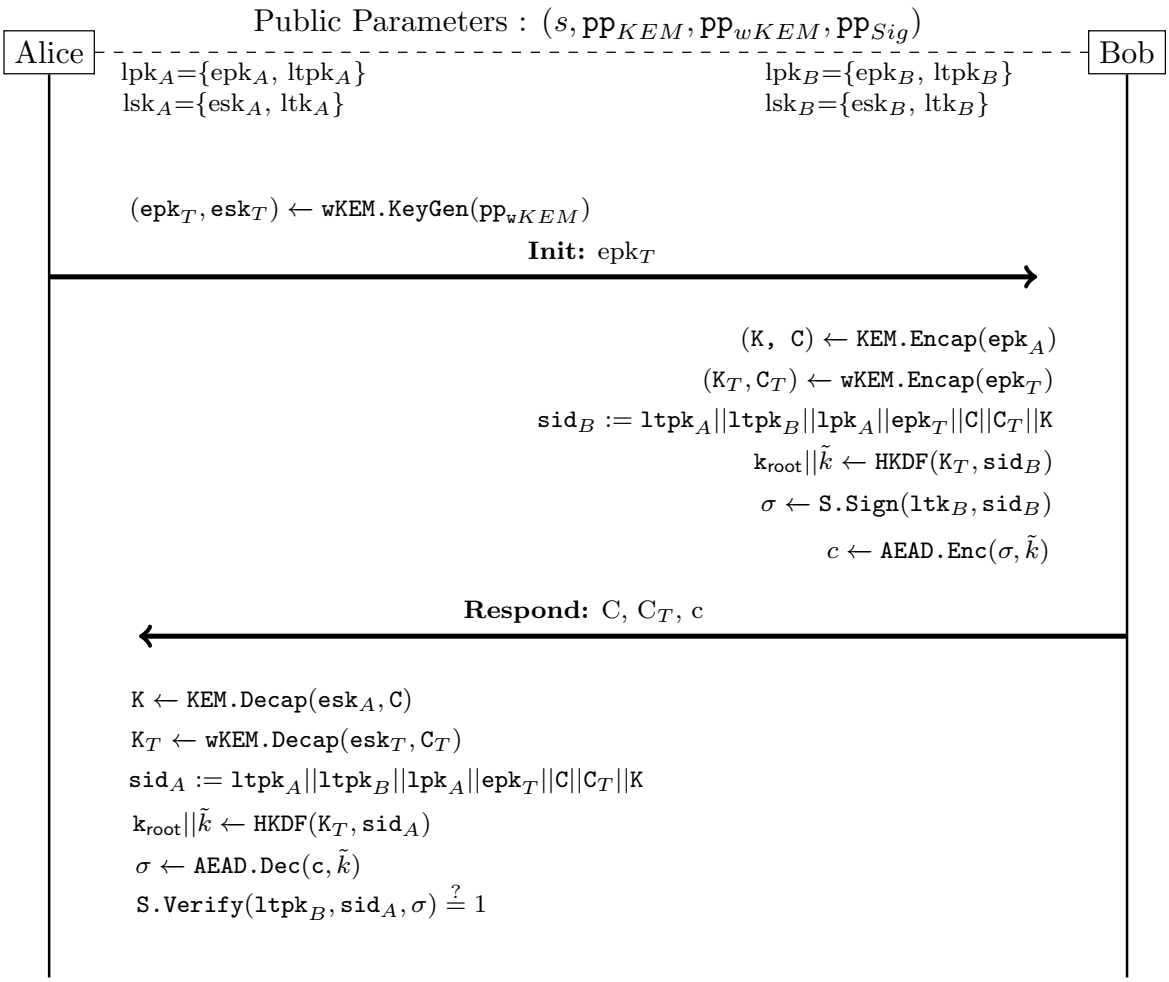



**Figure 3.1** – The PQ-X3DH protocol.

X3DH [Treb] is an asynchronous protocol that generates a shared secret between the communicating parties to initialize their communication as well as authenticate themselves. It fully authenticates the receiver Bob and partially authenticates the initiator Alice. It is called asynchronous because both parties can initiate the connection while the other is offline. Such property provides flexibility but could completely break the protocol in the case of a malicious server. Apart from such a case the asynchronous protocol is highly secure. We consider here the PQ-X3DH presented in [HKKP21b] which preserve the security properties of the classical protocol and we focus on the variant of PQ-X3DH that does not use a signature to fully authenticate Alice. The

motivation for this change is to allow Alice to deny having taken part in the exchange, in the same way that Bob can deny it thanks to the encryption of his signature.

The PQ-X3DH sub-protocol is presented in Figure 3.1. Two key encapsulation mechanisms, KEM and wKEM, are employed as building blocks in this key agreement protocol. wKEM, which is IND-CPA secure, is for ephemeral use. KEM is IND-CCA secure here. Tamarin considers the public-key encryption as ideal (thus IND-CCA), but for an ephemeral use, IND-CPA is sufficient.

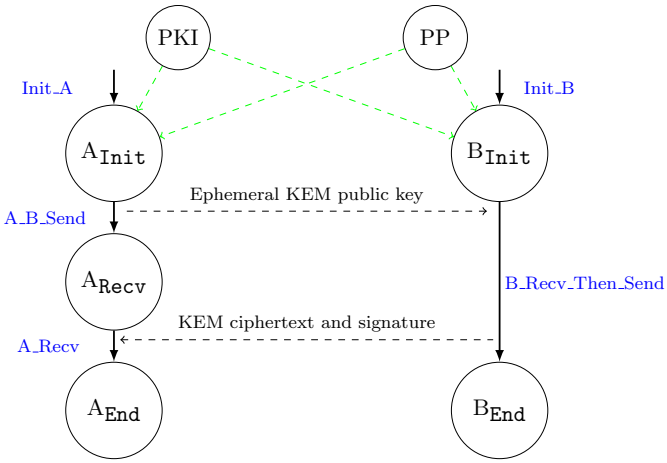



**Figure 3.2** – Graph of the Tamarin PQ-X3DH model.

#### 3.2.1.1 Tamarin model.

Since Tamarin has no built-in KEM we replace the KEM with an asymmetric encryption scheme encrypting a fresh ephemeral key. The two approaches are equivalent considering the idealization of cryptographic primitives used in Tamarin. Our model for the PQ-X3DH protocol is represented as the state machine in Figure 3.2 with nine transition rules:

- `PKI` and `PP`: these rules formalize the Public Key Infrastructure (PKI), the Public Parameters (PP). PKI assigns only once a long term key with an ephemeral key to a user. Instead of handling non-replayability with a batch of ephemeral one-time keys we directly use restrictions to ensure the a key can only be used once.
- `Init_A` and `Init_B`: each user get from PKI and PP the public parameters and public keys needed for the PQ-X3DH protocol.
- `A_B_Send`: Alice sends an ephemeral public key to initiate a key encapsulation.
- `B_Recv_Then_Send`: Bob receives Alice's ephemeral public key, encapsulates two secret keys into two KEM ciphertexts, one with Alice's ephemeral public key and wKEM, the other by using Alice's long-term public key and KEM. Bob also sign the protocol transcript and send his signature encrypted with an AEAD scheme.
- `A_Recv`: Alice receives Bob's ciphertext and signature and derives a session key that will be used by Alice and Bob to communicate. Then, she decrypts and verifies the signature.

- `RevealE`: Reveals to the attacker the ephemeral secret keys.
- `RevealL`: Reveals to the attacker the long-term secret keys.

### 3.2.2 The KEM-double-ratchet Protocol

The double ratchet protocol (DR for short) is used for securing an ongoing exchange of messages between two peers by repeatedly producing fresh session keys while saving the authentication made with the PQ-X3DH initialization.

This protocol is self-healing, which means that it is made so that if at some point a user's key is intercepted by an attacker, the upcoming renewal of the session key is there to protect the secrecy of the future messages. This property is sometimes called post-compromise security. To satisfy this property, a cryptographic ratchet based on a key exchange method, such as Diffie-Hellman in the classical case, is used in the protocol, and a ratchet based on key derivation functions enables key renewal without interaction between the peers.

In order to communicate securely, the double ratchet protocol derives three types of shared secrets: *root*, *chain* and *message* secrets. They are used respectively as *master*, *derivation* and a *message* keys [Trea]. Since we consider a KEM-based double ratchet, we deviate a bit from this definition. As specified in Figure 3.3, in KEM-Double-Ratchet the two communicating peers Alice and Bob start the KEM-DR sub-protocol with a common pre-shared key $\mathsf{k}_{\mathsf{root}}$. This key comes from the key agreement protocol PQ-X3DH.

#### 3.2.2.1 Tamarin model.

As shown in Figure 3.4 we only perform three exchanges of the KEM-double-ratchet protocol. Three exchanges are sufficient to verify all considered security properties as discussed in section 3.3. Our KEM-DR model has nine transition rules:

- `Init_A` and `Init_B`: each user get a secret preshared key and Bob gets an Alice's KEM ephemeral public key.
- `Send_B1`, `Send_A`, and `Send_B2`: the user encapsulates a fresh secret key with the current KEM public key, derives a new session key, encrypts the message, then sends it encrypted with the KEM ciphertext and a new ephemeral KEM public key.
- `Recv_A1`, `Recv_B`, and `Recv_A2`: the user receives a message, derives the new session key, and verifies the integrity.
- `LeakState`: Reveals to the attacker the current user secrets.

## 3.3 Tamarin Formal Verification

In Tamarin, a protocol is seen as a state machine. A state is a multiset of facts, and rules are transitions which shift the state when some conditions are fulfilled. A rule consists of three sets of facts: *premise*, *action facts*, and *conclusion*. If all the premise

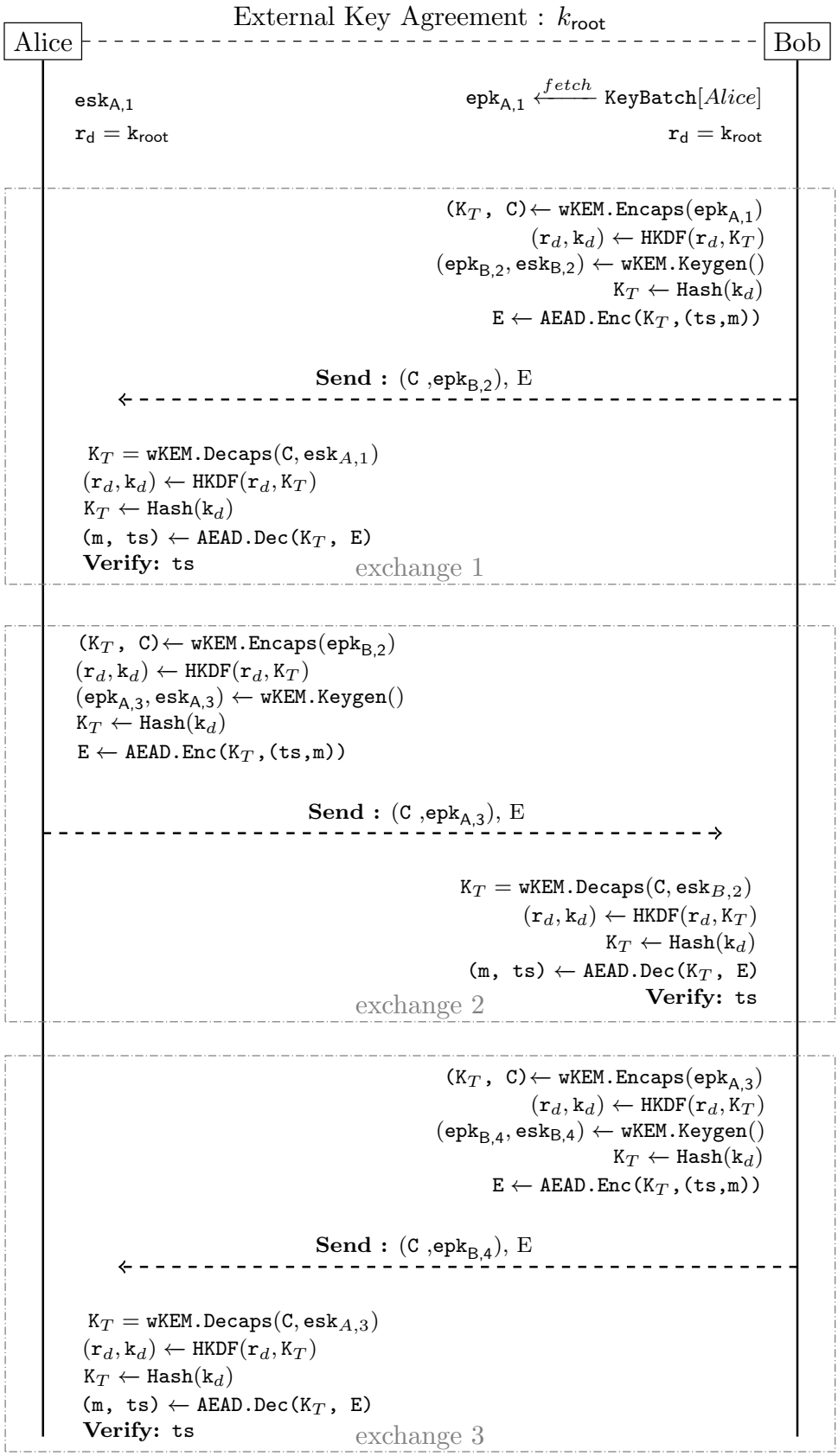



**Figure 3.3** – The KEM-double-ratchet protocol.

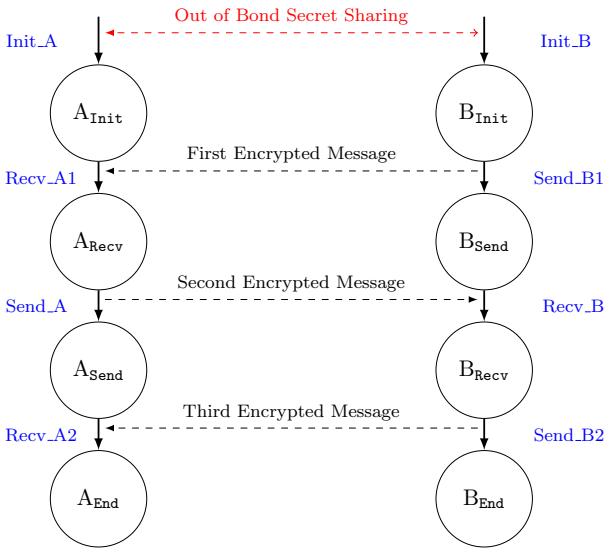



**Figure 3.4** – Graph of the Tamarin KEM-Double-Ratchet model.

facts exist, then the rule is applied. Applying a rule means *consuming* premise facts to produce conclusion facts while recording action facts in the protocol *trace*.

Some facts are native in Tamarin such as `In()` and `Out()` to model inputs and outputs of the protocol following the Dolev-Yao model [DY83]. Moreover, the `Fr()` fact is used to produce fresh or unique variables.

Tamarin proposes a set of *built-ins* cryptographic primitives to model protocols, including symmetric and asymmetric encryption, hash function, and signature. It also allows to define new primitives, via `functions` and `equations` commands. In the context of this work, we define a KEM as an asymmetric encryption scheme encrypting a fresh random key, and we consider the following AEAD Tamarin formalization from [HNS+21].

In some cases, we need to restrict some transitions in the protocol, e.g., to check the equality of two terms as shown below. Hence, when a rule has the restriction `Eq(x,y)` in its action facts, then the rule is applied if and only if $x = y$.

```
restriction Eq: "All x y #i. Eq(x,y) @ #i ==> x = y"
```

Tamarin uses the logic of first order to formalize security properties as *lemmas*. The keyword `All` stands for $\forall$, `Ex` for $\exists$ and `@` represents a marker for chronological events. Lemmas use *action facts* produced by the rules to prove or disprove properties. A trivial lemma is given below, it means that if `Action1()` happened then `Action2()` happened too.

```
lemma example:
   " All x #i. Action1(x) @ #i ==> Ex y #k. Action2(y) @ #k "
```

In order to verify that a protocol has given security property, Tamarin takes as input the protocol model, with all its possible transitions as rules, and a lemma corresponding to this security property. Then, if Tamarin completes its verification process, it will either output a proof of the property or an attack trace which falsifies it.

### 3.3.1 Security Properties

The security properties verified in our symbolic analysis of PQ-X3DH and KEM-DR are the same properties as those considered in the formal verification of classical X3DH and Double-Ratchet protocols [KBB17].

**Integrity.** Integrity is an important property for key exchange protocols. It allows the receiver of a message to have the assurance that the message has come unaltered from the intended sender. We separate the integrity of a message that can be verified upon receipt, called instantaneous integrity, and that which can be verified upon receipt of a subsequent message, called delayed integrity.

**Authentication.** We consider two authentication notions: the *partial authentication* from definition 3.1 and the *full authentication* from defintion 3.2.

**Definition 3.1** (Partial Authentication)**.** *A user $U$ is partially authenticated to another user $V$ if $V$ can prove that the message she receives comes from the same user as the one with whom she has initialized the session.*

**Definition 3.2** (Full Authentication)**.** *A user $U$ is fully authenticated to another user $V$ if $U$ is partially authenticated to $V$ and $V$ can prove the identity of $U$.*

**Forward Secrecy.** Here again we consider two different notions: the *perfect Forward Secrecy* (FS) from definition 3.3 and the *weak Forward Secrecy* (wFS) from definition 3.4. In this work, we also consider a new notion called *weak state Forward Secrecy* (wsFS) that we define as the wFS property except that the leakage concerns states instead of long term keys.

**Definition 3.3** (perfect Forward Secrecy (FS) [MOV96])**.** *A protocol is said to have* perfect forward secrecy *if compromised long-term keys does not compromise past session keys.*

**Definition 3.4** (weak Forward Secrecy (wFS) [Kra05])**.** *Any session key established by uncorrupted parties without active intervention by the attacker is guaranteed to remain secrete even if the parties to the exchange are corrupted after the session key has been erased.*

**Key Compromise Impersonation Resistance.** The KCI resistance from definition 3.5 is related to the use of long term public/private keys. Since there is no use of long term public/private key in the KEM-DR protocol, the KCI property is only applicable to the PQ-X3DH protocol.

**Definition 3.5** (Key Compromise Impersonation (KCI) Resistance)**.** *Even if an adversary compromises the long term private key of a user $U$, this adversary can not use this key to impersonate (to $U$) another user $V$ that is communicating to $U$.*

**Unknown Key-Share Resistance.** We recall the definition of a UKS attack in definition 3.6. UKS attacks can be seen as implicit impersonation. Thus, in the same way as for the KCI resistance, this property is only applicable to the PQ-X3DH protocol.

**Definition 3.6** (Unknown Key-Share (UKS) attack [BM99])**.** *An unknown key-share attack on an AKE protocol is an attack whereby a user $U$ ends up believing she shares a key with $V$, and although this is in fact the case, $V$ mistakenly believes the key is instead shared with an entity $W \neq U$.*

**Post-Compromise Recovery.** We recall the post-compromise recovery property in definition 3.7. By definition, this property is only applicable when a protocol is iterated repeatedly between the parties, this is the case for KEM-DR but not for PQ-X3DH.

**Definition 3.7** (Post-Compromise Recovery (PCR) [ACD19])**.** *If the attacker remains passive, i.e., the attacker does not inject any messages, and if users have access to fresh randomness, then the users recover a secure state from a compromised state after a few communication rounds.*

### 3.3.2 Tamarin Formalization

Before describing in more detail the Tamarin formalization of the different security properties presented above, we need to define some notations that will be useful later. In the rest of the paper, we use the following notations:

- $\mathcal{L} = \{0, 1, 2\}$ is the set of iteration indices, i.e., the three first exchanges ;
- $\mathcal{M} \subset \{0, 1\}^*$ is the set of messages sent via the Signal protocol ;
- $\mathcal{K} \subset \{0, 1\}^k$ is the set of secret keys where $k$ is the key length ;
- $\mathcal{S} \subset \{0, 1\}^*$ is the set of message indices, i.e., the message numbers ;
- $\Sigma$ is the set of protocol states.
- $\Gamma$ is the set of user states.

Moreover, using the Tamarin formalism, we note:

- KU(x): The adversary sent x and therefore has knowledge of x ;
- K(x): The adversary has knowledge of x.

In Table 3.1, respectively Table 3.2, we introduce the Tamarin action facts and their abbreviations needed in our symbolic analysis of PQ-X3DH, respectively KEM-DR. These action facts are used to define the Tamarin lemmas corresponding to the security properties.

In Tamarin, the user state is the current set of the secrets of this user. In order to characterize respectively full and partial knowledge of the user's secrets by the attacker, we define the *revealed* state in definition 3.8 and the *compromised* state in definition 3.9.

**Definition 3.8** (Revealed State)**.** *A state is said to be* revealed *if the adversary has knowledge of every hidden elements of the state.*

**Table 3.1** – Tamarin action facts for PQ-X3DH, abbreviations and definitions

| Action fact | Abbreviation | Definition |
|---|---|---|
| SessA | $S_A$(A,B,k) | A accepts the key k as valid to communicate with B |
| ExplicitAuth | EA(B,A) | A explicitly authenticates B |
| RevealL | $R_L$(A) | The long-term key of A is revealed to the attacker |
| RevealE | $R_E$(A) | The ephemeral key of A is revealed to the attacker |
| SendConnect | SC(A) | A initiates the PQ-X3DH protocol |
| RecvConnect | RC(A,B) | B receives the initialization message from A |
| SendSign | SS(B,A) | B sends its signature to A |
| weakFS | $wFS_{A,B}$(k) | Saves k to check its resistance against future reveal |
| Send/Recv | $Send_{A,n}$(m)<br>$Recv_{B,A,n}$(m) | A sends the message m of index n<br>B checks the integrity of message m of index n from A |

**Table 3.2** – Tamarin action facts for KEM-Double-Ratchet, abbreviations and definitions

| Action fact | Abbreviation | Definition |
|---|---|---|
| IntegS/IntegR | $I_{S/R}$(n,m,s) | Sends or receives message m of index n associated with session key s and checks its integrity in reception |
| FS | FS(A,B,n,st) | Saves the current state st associated with the sending of the message of index n between A and B in order to check its resistance against future reveal |
| Healed | $H_{A,B}$(st) | Checks if the state st has recovered from a previous reveal in the communication between A and B |
| Reveal | R(A,n) | Reveals the secrets of A associated with the message of index n. |

**Definition 3.9** (Compromised State)**.** *A state is said to be* compromised *if the adversary has knowledge of any hidden element of the state.*

#### 3.3.2.1 PQ-X3DH Security Properties.

For the sake of clarity, we describe the security properties verified with Tamarin in the usual mathematical formalism. Let E = A $<_t$ B the notation $<_t$ means that E is true if and only if event A occurs before event B. The definitions of the corresponding Tamarin lemmas are presented in appendix 3.4.

**Integrity.** The integrity is checked on both messages transmitted through the PQ-X3DH protocol. Nothing in this protocol allows an immediate integrity check of the first transmitted message. However, if both parties share the same key at the end of the protocol, then the integrity of this message is ensured in a delayed manner. For this reason, we define the following condition under which the delayed integrity of the

first message is assured:

$$\forall \mathrm{A}, \mathrm{B} \in \mathcal{U}, \forall \mathrm{m}_1, \mathrm{m}_2 \in \mathcal{M}, \forall \mathrm{k} \in \mathcal{K}.$$
$$\mathrm{S_A(A,B,k)} \wedge \mathrm{S_B(B,A,k)} \wedge \mathrm{Send(A,1,m_1)} \wedge \mathrm{Recv(B,A,1,m_2)} \implies \mathrm{m_1} = \mathrm{m_2} \ .$$

The integrity of the second message can be immediately verified thanks to the signature. Thus the condition to check the immediate integrity of this message is as follows:

$$\forall \mathrm{A}, \mathrm{B} \in \mathcal{U}, \ \forall \mathrm{m}_1, \mathrm{m}_2 \in \mathcal{M}. \ \mathrm{Send_{B,2}(m_1)} \wedge \mathrm{Recv_{A,B,2}(m_2)} \wedge \neg \mathrm{R_E(A)} \wedge \neg \mathrm{R_L(B)}$$
$$\wedge \ \mathrm{R_E(A)} \leq_t \mathrm{Recv_{A,B,2}(m_2)} \wedge \mathrm{R_L(B)} \leq_t \mathrm{Recv_{A,B,2}(m_2)} \implies \mathrm{m_1} = \mathrm{m_2} \ .$$

**Authentication.** We consider two different notions of authentication depending on the role of the user in the PQ-X3DH protocol. Indeed, the initiator, Alice, does not sign any message and her KEM long term key is provided by a server without guaranteeing its authenticity. In these conditions, Alice can only be partially authenticated according to definition 3.1. In the case where the equivalent of a certificate of the Alice's KEM long term key was added, then she could be fully authenticated at the end of the PQ-X3DH protocol. The second user, Bob, signs the message which allows Alice to explicitly authenticates him under the classical conditions of a public key infrastructure. The following condition is for the full authentication of Bob by Alice:

$$\forall \mathrm{A}, \mathrm{B} \in \mathcal{U}. \ \mathrm{EA(B,A)} \wedge \neg \mathrm{R_L(B)} \wedge \mathrm{EA(B,A)} \leq_t \mathrm{R_L(B)} \implies \mathrm{SS(B,A)} \wedge \mathrm{RC(A,B)}$$
$$\wedge \ \mathrm{SC(A)} \wedge \mathrm{SS(B,A)} \leq_t \mathrm{EA(B,A)} \wedge \mathrm{SC(A)} \leq_t \mathrm{SS(B,A)} \wedge \mathrm{RC(A,B)} =_t \mathrm{SS(B,A)} \ .$$

**weak Forward Secrecy.** The following condition verifies the wFS property on $k_{\text{root}}$ and $\tilde{k}$ keys of the PQ-X3DH protocol in case of future compromise of the initiator's short-term and responder's signing keys.

$$\forall \mathrm{A}, \mathrm{B} \in \mathcal{U}, \forall \mathrm{k}_1, \mathrm{k}_2 \in \mathcal{K}. \ \mathrm{wFS_{A,B}(k_1, k_2)} \wedge \neg \mathrm{R_L(B)}$$
$$\wedge \ \mathrm{R_L(B)} \leq_t \mathrm{wFS_{A,B}(k_1, k_2)} \wedge \mathrm{R_E(A)} \implies \neg \mathrm{KU(k_1)} \wedge \neg \mathrm{KU(k_2)} \ .$$

**KCI resistance.** The only possible scenario in which a KCI attack occurs is when the signing long-term key of the responder is compromised, in this case it must be guaranteed that an attacker cannot use this key to impersonate any of the users. Thus we have the following condition for KCI resistance:

$$(\forall \mathrm{A}, \mathrm{B}, \mathrm{S} \in \mathcal{U}, \forall \mathrm{k} \in \mathcal{K}. \mathrm{S_A(A,S,k)} \wedge \mathrm{R_L(A)} \wedge \mathrm{S_B(B,A,k)} \implies \mathrm{S} = \mathrm{B}) \wedge$$
$$(\forall \mathrm{A}, \mathrm{B}, \mathrm{S} \in \mathcal{U}, \forall \mathrm{k} \in \mathcal{K}. \mathrm{S_A(A,B,k)} \wedge \mathrm{R_L(A)} \wedge \mathrm{S_B(B,S,k)} \implies \mathrm{S} = \mathrm{A}) \ .$$

**UKS resistance.** The UKS resistance consists of ensuring that if two users have agreed on a common session key, then they have the assurance that if neither key is compromised then no other user can impersonate either of them.

$$\forall \mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{S} \in \mathcal{U}, \forall \mathrm{k} \in \mathcal{K}. \ \mathrm{S_A(A,S,k)} \wedge \mathrm{S_B(B,C,k)} \implies \mathrm{S} = \mathrm{B} \wedge \mathrm{C} = \mathrm{A} \ .$$

#### 3.3.2.2 Double-Ratchet Security Properties.

Regarding the KEM-DR protocol, we verify the classical security properties as well as the post-compromise recovery from [ACD19].

**Integrity.** For each exchange, we verify that the message sent is indeed the message received thanks to the integrity provided by the AEAD scheme.

$$\forall n \in \mathcal{S}, \forall m_1, m_2 \in \mathcal{M}, \forall k \in \mathcal{K}.\ I_S(n,m_1,s) \wedge I_R(n,m_2,s) \wedge \neg K(k) \implies m_1 = m_2 \ .$$

**PCR.** As this property ensures that for a corruption during a given exchange it is enough to wait for two exchanges before the session key is secret again, it is necessary to check this condition on three consecutive exchanges. Then this base case allows to prove theorem 3.1 by induction.

$$\begin{aligned}&\forall A, B \in \mathcal{U}, \forall st \in \Sigma.\\ &H_{A,B}(st) \wedge R(B,0) \wedge H_{A,B}(st) <_t R(B,0) \implies \neg K(st) \vee R(B,2) \vee R(A,1) \ .\end{aligned}$$

**weak state Forward Secrecy.** Similarly, two consecutive exchanges of the KEM-DR protocol are sufficient to prove by induction the wsFS property in theorem 3.2.

$$\begin{aligned}&\forall A, B \in \mathcal{U}, \forall st \in \Sigma.\\ &FS(A,B,0,st) \wedge R(A,1) \wedge FS(A,B,0,st) <_t R(A,1) \implies \neg K(st) \vee R(B,0) \ .\end{aligned}$$

### 3.3.3 Formal Verification Results

In Table 3.3 we present the results obtained from the automatic verification with Tamarin of the security properties considered for the PQ-X3DH and KEM-DR protocols.

**Table 3.3** – Results of Tamarin verification for PQ-X3DH and KEM-Double-Ratchet protocols.

| Protocols | Integrity | | Auth. | Imp. resistance | | Forward secrecy | | | PCR |
|---|---|---|---|---|---|---|---|---|---|
| | Instant | Delayed | | KCI | UKS | FS | wFS | wsFS | |
| PQ-X3DH | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | NA | NA |
| KEM-Double-Ratchet | ✓ | NA | NA | NA | NA | ✓ | NA | ✓ | ✓ |

Since the KEM-DR protocol admits an arbitrary number of interactions, properties impacting previous or future states of the protocol require an additional proof in order to holds for any exchange of the protocol. Only the PCR and wsFS properties fall in this case, the others are trivially proven.

**Theorem 3.1** (KEM-Double-Ratchet Post Compromise Security)**.** *For all user state* $State_n$ *with* $n > 0$*:*

$$\begin{aligned} &\textit{Compromised}(State_n) \land \neg\ \textit{Revealed}(State_{n+1}) \land \neg\ \textit{Revealed}(State_{n+2}) \\ &\implies \textit{Healed}_n(State_{n+2})\ . \end{aligned}$$

*Proof.* We prove theorem 3.1 by induction for all integer $n > 0$. The base case has been proven using Tamarin. Suppose that the theorem is true for all integer $k < n$, and that:

$$\neg\ \texttt{Healed}_{n+1}(\text{State}_{n+3}) \quad \text{with} \quad \neg\ \texttt{Compromised}(\text{State}_{n+3})$$

By definition of a healed state, for all $n > 2$ we have:

$$\texttt{Healed}(\text{State}_n) \iff \exists k < n,\ \ \texttt{Revealed}(\text{State}_k)\ \land \neg\ \texttt{Compromised}(\text{State}_n)$$

And thus:

$$\begin{aligned} &\neg\ \texttt{Healed}(\text{State}_{n+3})) \implies \forall k < n+3, \\ &\texttt{Revealed}(\text{State}_k) \lor \texttt{Compromised}(\text{State}_{n+3}) \end{aligned}$$

Since the knowledge of the $\text{State}_k$ decapsulation secret key implies the knowledge of the $\text{State}_{k+1}$ exchanged key, we have:

$$\texttt{Revealed}(\text{State}_k) \implies \texttt{Compromised}(\text{State}_k)\ \land\ \texttt{Compromised}(\text{State}_{k+1})$$

And then:

$$\begin{aligned} &\neg\ \texttt{Healed}(\text{State}_{n+3}) \implies \forall k < n+3, \\ &\texttt{Compromised}(\text{State}_k)\ \land\ \texttt{Compromised}(\text{State}_{k+1})\ \lor\ \texttt{Compromised}(\text{State}_{n+3}) \end{aligned}$$

Finally, by induction hypothesis:

$$\begin{aligned} &\neg\ \texttt{Healed}(\text{State}_{n+1})) \\ &\implies \neg\ \texttt{Compromised}(\text{State}_{n+1}) \lor\ \texttt{Revealed}(\text{State}_{n+2}) \lor \texttt{Revealed}(\text{State}_{n+3}) \end{aligned}$$

□

We introduce the notion of *healing ball* in definition 3.10 to prove the wsFS property in theorem 3.2.

**Definition 3.10** (Healing Ball)**.** *We define the* healing ball $B_h$ *for all user state* $S \in \Gamma$*, as* $B_h(S) = \{\gamma \in \Gamma \mid \textit{Revealed}(\gamma) \implies \neg\ \textit{Healed}(S)\}$*.*

**Theorem 3.2** (KEM-Double-Ratchet weak state Forward Secrecy)**.** *For all user state* $State_n$ *with* $n \geq 2$*:*

$$\begin{aligned} &\textit{Compromised}(State_n) \land (\forall k < n,\ S \in B_h(State_k), \neg\ \textit{Revealed}(S)) \\ &\implies \neg\ \textit{Compromised}(State_k)\ . \end{aligned}$$

*Proof.* We prove theorem 3.2 by induction for all integer $n > 1$. The base case has been proven using Tamarin. Suppose that the theorem is true for all integer $\ell \leq n$, and that $\text{State}_{n+1}$ has been compromised. Then, we have:

$$\texttt{Compromised}(\text{State}_{n+1}) \implies \texttt{Revealed}(\text{State}_n) \vee \texttt{Revealed}(\text{State}_{n+1})$$

and by definition, for all $\ell$:

$$\texttt{Revealed}(\text{State}_\ell) \implies \texttt{Compromise}(\text{State}_\ell)$$

If $\text{State}_n$ has been revealed and not $\text{State}_{n+1}$, we apply the induction hypothesis. Now suppose that $\text{State}_{n+1}$ has been revealed but not $\text{State}_n$, we then use the fact that KEM.decaps is supposed ideal by Tamarin and then deterministic, so regarding the backward analysis $\text{State}_{n+1}$ is a deterministic function of $\text{State}_n$. Finally, if both states have been revealed, then we apply the induction hypothesis. □



## 3.4 Tamarin Security Lemmas

In this section we present the Tamarin lemmas corresponding to security properties presented in section 3.3.2 and defined in our formal verification of the PQ-X3DH and KEM-Double-Ratchet protocols.

### 3.4.1 PQ-X3DH Security Properties

```
lemma integrity_first_exchange:
    "
    All A B exch1 exch2 sess #i #j #l1 #l2.
    SessA(A, B, sess)@#i & SessB(B,A,sess)@#j
    & Send(A, '1', exch1)@#l1
    & Recv(B,A, '1', exch2)@#l2
    ==>
    exch1 = exch2
    "

lemma integrity_second_exchange:
    "
    All A B exch1 exch2 #i #j.
    Send(B, '2', exch1)@#i & Recv(A,B, '2', exch2)@#j
    & not(Ex #k. RevealE(A)@#k & #k<#j)
    & not(Ex #k. RevealL(B)@#k & #k<#j)
     ==> exch1 = exch2
    "

lemma explicit_authentication:
    "
```

```
    All A B #i.
    ExplicitAuth(B,A)@#i & not(Ex #r. RevealL(B)@#r & #r<#i)
    ==>
    (Ex #k #l. SendSign(B,A)@#k & RecvConnect(A,B)@#k
    & SendConnect(A)@#l &#k<#i & #l<#k)
    "

lemma weak_forward_secrecy:
    "
    All A B ss1 ss2 #i.
    WeakFS(A, B, ss1, ss2)@#i
    & not(Ex #k. RevealL(B)@#k & #k<#i)
    & not(Ex #k. RevealE(A)@#k)
    ==> not(Ex #r. KU(ss1)@#r) & not(Ex #r. KU(ss2)@#r)
    "

lemma key_compromise_attack_resistance:
    "
    (
    All A S B sess #i #j #k.
    SessA(A,S, sess)@#i & RevealL(A)@#k & SessB(B, A, sess)@#j
    ==> S=B
    )
    &
     (
    All A S B sess #i #j #k.
    SessA(A,B, sess)@#i & RevealL(B)@#k & SessB(B, S, sess)@#j
    ==> S=A
    )
    "

lemma unknown_keyshare_resistance:
    "
    All A C S B sess #i #j.
    SessA(A,S, sess)@#i & SessB(B,C, sess)@#j ==> S=B & C=A
    "
```

### 3.4.2 KEM-Double-Ratchet Security Properties

```
lemma Integrity:
            "
            All l x y s #i #j.
            IntegS(l, x, s)@#i & IntegR(l, y, s)@#j
            & not(Ex #k. K(s)@#k)
```

```
            ==>
            x=y
            "

lemma Post_Compromise_Security:
            "
            All A B state #i #j.
            Healed(A, B, state)@#i & Reveal(B, '0')@#j & #i<#j
            ==>
            not(Ex #k. K(state)@#k) |
            (Ex #k. Reveal(B, '2')@#k) | (Ex #k. Reveal(A, '1')@#k)
            "

lemma Forward_Secrecy:
            "
            All A B state #i #j.
            FS(A, B, '0', state)@#i & Reveal(A, '1')@#j & #i<#j
            ==>
            not(Ex #k. K(state)@#k) | (Ex #k. Reveal(B, '0')@#k)
            "
```

# Références

[ABD+21] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler et Damien Stehlé. *CRYSTALS-Kyber – Submission to round 3 of the NIST post-quantum project*. In : 2021. (Cf. p. 40).

[ACD19] Joël Alwen, Sandro Coretti et Yevgeniy Dodis. *The Double Ratchet : Security Notions, Proofs, and Modularization for the Signal Protocol*. In : *EUROCRYPT 2019, Part I*. Sous la dir. d'Yuval Ishai et Vincent Rijmen. T. 11476. LNCS. Springer, Heidelberg, mai 2019, p. 129-158 (cf. p. 11, 40, 41, 48, 51).

[BBK17] Karthikeyan Bhargavan, Bruno Blanchet et Nadim Kobeissi. *Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate*. In : *IEEE Symposium on Security and Privacy, SP*. 2017, p. 483-502 (cf. p. 40).

[BDH+18] David A. Basin, Jannik Dreier, Lucca Hirschi, Sasa Radomirovic, Ralf Sasse et Vincent Stettler. *A Formal Analysis of 5G Authentication*. In : *CCS*. Sous la dir. de David Lie, Mohammad Mannan, Michael Backes et XiaoFeng Wang. 2018, p. 1383-1396 (cf. p. 40).

[BFG+20] Jacqueline Brendel, Marc Fischlin, Felix Günther, Christian Janson et Douglas Stebila. *Towards Post-Quantum Security for Signal's X3DH Handshake*. In : *Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers*. Sous la dir. d'Orr Dunkelman, Michael J. Jacobson Jr. et Colin O'Flynn. T. 12804. Lecture Notes in Computer Science. Springer, 2020, p. 404-430 (cf. p. 40).

[Bla16] Bruno Blanchet. *Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif*. In : *Found. Trends Priv. Secur.* 1.1-2 (2016), p. 1-135 (cf. p. 40).

[BM99] Simon Blake-Wilson et Alfred Menezes. *Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol*. In : *Public Key Cryptography, Second International Workshop on Practice and Theory in Public Key Cryptography, PKC*. T. 1560. 1999, p. 154-170 (cf. p. 48).

[CCD+20] Katriel Cohn-Gordon, Cas Cremers, Benjamin Dowling, Luke Garratt et Douglas Stebila. *A Formal Security Analysis of the Signal Messaging Protocol*. In : *J. Cryptol.* 33.4 (2020), p. 1914-1983 (cf. p. 40).

[CHH+17] Cas Cremers, Marko Horvat, Jonathan Hoyland, Sam Scott et Thyla van der Merwe. *A Comprehensive Symbolic Analysis of TLS 1.3*. In : *CCS*. 2017, p. 1773-1788 (cf. p. 40).

[CHSW22] Sofía Celi, Jonathan Hoyland, Douglas Stebila et Thom Wiggers. *A Tale of Two Models : Formal Verification of KEMTLS via Tamarin*. In : *ESORICS 2022, Part III*. Sous la dir. de Vijayalakshmi Atluri, Roberto Di Pietro, Christian Damsgaard Jensen et Weizhi Meng. T. 13556. LNCS. Springer, Heidelberg, sept. 2022, p. 63-83 (cf. p. 41).

[DY83] Danny Dolev et Andrew Chi-Chih Yao. *On the security of public key protocols*. In : *IEEE Trans. Inf. Theory* 29.2 (1983), p. 198-207 (cf. p. 46).

[HKKP21b] Keitaro Hashimoto, Shuichi Katsumata, Kris Kwiatkowski et Thomas Prest. *An Efficient and Generic Construction for Signal's Handshake (X3DH) : Post-Quantum, State Leakage Secure, and Deniable*. In : *PKC*. T. 12711. 2021, p. 410-440 (cf. p. 40-42).

[HNS+21] Andreas Hülsing, Kai-Chun Ning, Peter Schwabe, Florian Weber et Philip R. Zimmermann. *Post-quantum WireGuard*. In : *2021 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, mai 2021, p. 304-321 (cf. p. 41, 46).

[KBB17] Nadim Kobeissi, Karthikeyan Bhargavan et Bruno Blanchet. *Automated Verification for Secure Messaging Protocols and Their Implementations : A Symbolic and Computational Approach*. In : *EuroS&P*. 2017, p. 435-450 (cf. p. 40, 47).

[Kra05] Hugo Krawczyk. *HMQV : A High-Performance Secure Diffie-Hellman Protocol*. In : *CRYPTO*. Sous la dir. de Victor Shoup. T. 3621. 2005, p. 546-566 (cf. p. 41, 47).

[Low96] Gavin Lowe. *Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR*. In : *TACAS*. T. 1055. 1996, p. 147-166 (cf. p. 40).

[MOV96] Alfred Menezes, Paul C. van Oorschot et Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996 (cf. p. 47).

[MSCB13] Simon Meier, Benedikt Schmidt, Cas Cremers et David A. Basin. *The TAMARIN Prover for the Symbolic Analysis of Security Protocols*. In : *Computer Aided Verification CAV*. T. 8044. 2013, p. 696-701 (cf. p. 40).

[SSW20] Peter Schwabe, Douglas Stebila et Thom Wiggers. *Post-Quantum TLS Without Handshake Signatures*. In : *CCS '20 : 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event*. 2020, p. 1461-1480 (cf. p. 41).

[Trea] Moxie Marlinspike Trevor Perrin. *The Double Ratchet Algorithm*. (Cf. p. 39, 44).

[Treb] Moxie Marlinspike Trevor Perrin. *The X3DH Key Agreement Protocol*. (Cf. p. 39, 42).

# Bibliographie


[ABD+21] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler et Damien Stehlé. *CRYSTALS-Kyber – Submission to round 3 of the NIST post-quantum project*. In : 2021. (Cf. p. 40).

[ACD19] Joël Alwen, Sandro Coretti et Yevgeniy Dodis. *The Double Ratchet : Security Notions, Proofs, and Modularization for the Signal Protocol*. In : *EUROCRYPT 2019, Part I*. Sous la dir. d'Yuval Ishai et Vincent Rijmen. T. 11476. LNCS. Springer, Heidelberg, mai 2019, p. 129-158 (cf. p. 11, 40, 41, 48, 51).

[Ajt96] Miklós Ajtai. *Generating Hard Instances of Lattice Problems (Extended Abstract)*. In : *28th ACM STOC*. ACM Press, mai 1996, p. 99-108 (cf. p. 7).

[ALS20] Thomas Attema, Vadim Lyubashevsky et Gregor Seiler. *Practical Product Proofs for Lattice Commitments*. In : *CRYPTO 2020, Part II*. Sous la dir. de Daniele Micciancio et Thomas Ristenpart. T. 12171. LNCS. Springer, Heidelberg, août 2020, p. 470-499 (cf. p. 17-21, 23, 26, 28-30).

[APS15] Martin R. Albrecht, Rachel Player et Sam Scott. *On the concrete hardness of Learning with Errors*. In : *J. Math. Cryptol.* 9.3 (2015), p. 169-203. (Cf. p. 17, 30).

[AW18] Martin R. Albrecht et Michael Walter. "dgs, Discrete Gaussians over the Integers". Available at `https://bitbucket.org/malb/dgs`. 2018. (Cf. p. 31).

[BBK17] Karthikeyan Bhargavan, Bruno Blanchet et Nadim Kobeissi. *Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate*. In : *IEEE Symposium on Security and Privacy, SP*. 2017, p. 483-502 (cf. p. 40).

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss et Anna Lysyanskaya. *P-signatures and Noninteractive Anonymous Credentials*. In : *TCC 2008*. Sous la dir. de Ran Canetti. T. 4948. LNCS. Springer, Heidelberg, mars 2008, p. 356-374 (cf. p. 16).

[BCN18] Cecilia Boschini, Jan Camenisch et Gregory Neven. *Relaxed Lattice-Based Signatures with Short Zero-Knowledge Proofs*. In : *ISC 2018*. Sous la dir. de Liqun Chen, Mark Manulis et Steve Schneider. T. 11060. LNCS. Springer, Heidelberg, sept. 2018, p. 3-22 (cf. p. 16, 17, 21-23, 25).

[BCR+23] Olivier Blazy, Céline Chevalier, Thomas Ricosset, Guillaume Renaut, Éric Sageloli et Hugo Senet. *Efficient Implementation of a Post-Quantum Anonymous Credential Protocol*. In : *ARES'23 – Proceedings of the 18th International Conference on Availability, Reliability and Security*. Association for Computing Machinery, 2023 (cf. p. 10).

[BCRS23] Hugo Beguinet, Céline Chevalier, Thomas Ricosset et Hugo Senet. *Formal Verification of a Post-Quantum Signal Protocol with Tamarin*. In : *VECoS'23 – Proceedings of the 16th International Conference on Verification and Evaluation of Computer and Communication Systems*. LNCS. Springer, 2023 (cf. p. 10, 11).

[BDH+18] David A. Basin, Jannik Dreier, Lucca Hirschi, Sasa Radomirovic, Ralf Sasse et Vincent Stettler. *A Formal Analysis of 5G Authentication*. In : *CCS*. Sous la dir. de David Lie, Mohammad Mannan, Michael Backes et XiaoFeng Wang. 2018, p. 1383-1396 (cf. p. 40).

[BDL+18] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner et Chris Peikert. *More Efficient Commitments from Structured Lattice Assumptions*. In : *SCN 18*. Sous la dir. de Dario Catalano et Roberto De Prisco. T. 11035. LNCS. Springer, Heidelberg, sept. 2018, p. 368-385 (cf. p. 18).

[BFG+20] Jacqueline Brendel, Marc Fischlin, Felix Günther, Christian Janson et Douglas Stebila. *Towards Post-Quantum Security for Signal's X3DH Handshake*. In : *Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers*. Sous la dir. d'Orr Dunkelman, Michael J. Jacobson Jr. et Colin O'Flynn. T. 12804. Lecture Notes in Computer Science. Springer, 2020, p. 404-430 (cf. p. 40).

[Bla16] Bruno Blanchet. *Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif*. In : *Found. Trends Priv. Secur.* 1.1-2 (2016), p. 1-135 (cf. p. 40).

[BM99] Simon Blake-Wilson et Alfred Menezes. *Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol*. In : *Public Key Cryptography, Second International Workshop on Practice and Theory in Public Key Cryptography, PKC*. T. 1560. 1999, p. 154-170 (cf. p. 48).

[Bra00] Stefan A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates : Building in Privacy*. Cambridge, MA, USA : MIT Press, 2000. isbn : 0262024918 (cf. p. 16).

[CCD+20] Katriel Cohn-Gordon, Cas Cremers, Benjamin Dowling, Luke Garratt et Douglas Stebila. *A Formal Security Analysis of the Signal Messaging Protocol*. In : *J. Cryptol.* 33.4 (2020), p. 1914-1983 (cf. p. 40).

[CDHK15] Jan Camenisch, Maria Dubovitskaya, Kristiyan Haralambiev et Markulf Kohlweiss. *Composable and Modular Anonymous Credentials : Definitions and Practical Constructions*. In : *ASIACRYPT 2015, Part II*. Sous la dir. de Tetsu Iwata et Jung Hee Cheon. T. 9453. LNCS. Springer, Heidelberg, nov. 2015, p. 262-288 (cf. p. 16).

[Cha85] David Chaum. *Security Without Identification : Transaction Systems to Make Big Brother Obsolete*. In : *Commun. ACM* 28.10 (1985), p. 1030-1044. (Cf. p. 15).

[CHH+17] Cas Cremers, Marko Horvat, Jonathan Hoyland, Sam Scott et Thyla van der Merwe. *A Comprehensive Symbolic Analysis of TLS 1.3*. In : *CCS*. 2017, p. 1773-1788 (cf. p. 40).

[CHSW22] Sofía Celi, Jonathan Hoyland, Douglas Stebila et Thom Wiggers. *A Tale of Two Models : Formal Verification of KEMTLS via Tamarin*. In : *ESORICS 2022, Part III*. Sous la dir. de Vijayalakshmi Atluri, Roberto Di Pietro, Christian Damsgaard Jensen et Weizhi Meng. T. 13556. LNCS. Springer, Heidelberg, sept. 2022, p. 63-83 (cf. p. 41).

[CL01] Jan Camenisch et Anna Lysyanskaya. *An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation*. In : *EUROCRYPT 2001*. Sous la dir. de Birgit Pfitzmann. T. 2045. LNCS. Springer, Heidelberg, mai 2001, p. 93-118 (cf. p. 16).

[CL02] Jan Camenisch et Anna Lysyanskaya. *Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials*. In : *CRYPTO 2002*. Sous la dir. de Moti Yung. T. 2442. LNCS. Springer, Heidelberg, août 2002, p. 61-76 (cf. p. 16).

[CL03] Jan Camenisch et Anna Lysyanskaya. *A Signature Scheme with Efficient Protocols*. In : *SCN 02*. Sous la dir. de Stelvio Cimato, Clemente Galdi et Giuseppe Persiano. T. 2576. LNCS. Springer, Heidelberg, sept. 2003, p. 268-289 (cf. p. 16).

[CL04] Jan Camenisch et Anna Lysyanskaya. *Signature Schemes and Anonymous Credentials from Bilinear Maps*. In : *CRYPTO 2004*. Sous la dir. de Matthew Franklin. T. 3152. LNCS. Springer, Heidelberg, août 2004, p. 56-72 (cf. p. 16).

[Cv91] David Chaum et Eugène van Heyst. *Group Signatures*. In : *EUROCRYPT'91*. Sous la dir. de Donald W. Davies. T. 547. LNCS. Springer, Heidelberg, avr. 1991, p. 257-265 (cf. p. 16).

[DDLL13] Léo Ducas, Alain Durmus, Tancrède Lepoint et Vadim Lyubashevsky. *Lattice Signatures and Bimodal Gaussians*. In : *CRYPTO 2013, Part I*. Sous la dir. de Ran Canetti et Juan A. Garay. T. 8042. LNCS. Springer, Heidelberg, août 2013, p. 40-56 (cf. p. 31).

[dLS18] Rafaël del Pino, Vadim Lyubashevsky et Gregor Seiler. *Lattice-Based Group Signatures and Zero-Knowledge Proofs of Automorphism Stability*. In : *ACM CCS 2018*. Sous la dir. de David Lie, Mohammad Mannan, Michael Backes et XiaoFeng Wang. ACM Press, oct. 2018, p. 574-591 (cf. p. 17-19, 26, 29, 30).

[DY83] Danny Dolev et Andrew Chi-Chih Yao. *On the security of public key protocols*. In : *IEEE Trans. Inf. Theory* 29.2 (1983), p. 198-207 (cf. p. 46).

[ETWY22] Thomas Espitau, Mehdi Tibouchi, Alexandre Wallet et Yang Yu. *Shorter Hash-and-Sign Lattice-Based Signatures*. In : *CRYPTO 2022, Part II*. Sous la dir. d'Yevgeniy Dodis et Thomas Shrimpton. T. 13508. LNCS. Springer, Heidelberg, août 2022, p. 245-275 (cf. p. 33).

[Gal46] Évariste Galois. *Mémoire sur les conditions de résolubilité des équations par radicaux*. In : *Journal de mathématiques pures et appliquées* XI (1846), p. 417-433 (cf. p. 4).

[Gau01] Johann Carl Friedrich Gauss. *Disquisitiones arithmeticae*. Gerhard Fleischer, 1801 (cf. p. 4).

[GGM14] Christina Garman, Matthew Green et Ian Miers. *Decentralized Anonymous Credentials*. In : *NDSS 2014*. The Internet Society, fév. 2014 (cf. p. 16).

[Gt20] Torbjörn Granlund et the GMP development team. *GNU MP : The GNU Multiple Precision Arithmetic Library*. 6.2.1. 2020. (Cf. p. 31).

[HKKP21a] Keitaro Hashimoto, Shuichi Katsumata, Kris Kwiatkowski et Thomas Prest. *An Efficient and Generic Construction for Signal's Handshake (X3DH) : Post-Quantum, State Leakage Secure, and Deniable*. In : *PKC 2021, Part II*. Sous la dir. de Juan Garay. T. 12711. LNCS. Springer, Heidelberg, mai 2021, p. 410-440 (cf. p. 11).

[HKKP21b] Keitaro Hashimoto, Shuichi Katsumata, Kris Kwiatkowski et Thomas Prest. *An Efficient and Generic Construction for Signal's Handshake (X3DH) : Post-Quantum, State Leakage Secure, and Deniable*. In : *PKC*. T. 12711. 2021, p. 410-440 (cf. p. 40-42).

[HNS+21] Andreas Hülsing, Kai-Chun Ning, Peter Schwabe, Florian Weber et Philip R. Zimmermann. *Post-quantum WireGuard*. In : *2021 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, mai 2021, p. 304-321 (cf. p. 41, 46).

[ILV11] Malika Izabachène, Benoît Libert et Damien Vergnaud. *Block-Wise P-Signatures and Non-interactive Anonymous Credentials with Efficient Attributes*. In : *Cryptography and Coding - 13th IMA International Conference, IMACC 2011, Oxford, UK, December 12-15, 2011. Proceedings*. Sous la dir. de Liqun Chen. T. 7089. Lecture Notes in Computer Science. Springer, 2011, p. 431-450. (Cf. p. 16).

[Jos22] Josef Weidendorfer. *Callgrind : a call-graph generating cache and branch prediction profiler*. 3.20.0. 2022. (Cf. p. 32).

[KBB17] Nadim Kobeissi, Karthikeyan Bhargavan et Bruno Blanchet. *Automated Verification for Secure Messaging Protocols and Their Implementations : A Symbolic and Computational Approach*. In : *EuroS&P*. 2017, p. 435-450 (cf. p. 40, 47).

[Ker83a] Auguste Kerckhoffs. *La cryptographie militaire*. In : *Journal des sciences militaires* IX (jan. 1883), p. 5-38 (cf. p. 4).

[Ker83b] Auguste Kerckhoffs. *La cryptographie militaire*. In : *Journal des sciences militaires* IX (février 1883), p. 161-191 (cf. p. 4).

[Kra05] Hugo Krawczyk. *HMQV : A High-Performance Secure Diffie-Hellman Protocol*. In : *CRYPTO*. Sous la dir. de Victor Shoup. T. 3621. 2005, p. 546-566 (cf. p. 41, 47).

[KTB+21] Katarzyna Kapusta, Vincent Thouvenot, Olivier Bettan, Hugo Beguinet et Hugo Senet. *A Protocol for Secure Verification of Watermarks Embedded into Machine Learning Models*. In : *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security*. IH&MMSec '21. Virtual Event, Belgium : Association for Computing Machinery, 2021, p. 171-176. isbn : 9781450382953. (Cf. p. 10).

[LN17] Vadim Lyubashevsky et Gregory Neven. *One-Shot Verifiable Encryption from Lattices*. In : *EUROCRYPT 2017, Part I*. Sous la dir. de Jean-Sébastien Coron et Jesper Buus Nielsen. T. 10210. LNCS. Springer, Heidelberg, avr. 2017, p. 293-323 (cf. p. 19, 23, 26, 28, 30).

[LNP22] Vadim Lyubashevsky, Ngoc Khanh Nguyen et Maxime Plançon. *Lattice-Based Zero-Knowledge Proofs and Applications : Shorter, Simpler, and More General*. In : *CRYPTO 2022, Part II*. Sous la dir. d'Yevgeniy Dodis et Thomas Shrimpton. T. 13508. LNCS. Springer, Heidelberg, août 2022, p. 71-101 (cf. p. 17).

[Low96] Gavin Lowe. *Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR*. In : *TACAS*. T. 1055. 1996, p. 147-166 (cf. p. 40).

[MOV96] Alfred Menezes, Paul C. van Oorschot et Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996 (cf. p. 47).

[MSCB13] Simon Meier, Benedikt Schmidt, Cas Cremers et David A. Basin. *The TAMARIN Prover for the Symbolic Analysis of Security Protocols*. In : *Computer Aided Verification CAV*. T. 8044. 2013, p. 696-701 (cf. p. 40).

[Reg05] Oded Regev. *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*. In : *37th ACM STOC*. Sous la dir. d'Harold N. Gabow et Ronald Fagin. ACM Press, mai 2005, p. 84-93 (cf. p. 8).

[SSW20] Peter Schwabe, Douglas Stebila et Thom Wiggers. *Post-Quantum TLS Without Handshake Signatures*. In : *CCS '20 : 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event*. 2020, p. 1461-1480 (cf. p. 41).

[The23] The MPFR team. *GNU MPFR : The Multiple Precision Floating-Point Reliable Library*. 4.2.0. 2023. (Cf. p. 31).

[Trea] Moxie Marlinspike Trevor Perrin. *The Double Ratchet Algorithm*. (Cf. p. 39, 44).

[Treb] Moxie Marlinspike Trevor Perrin. *The X3DH Key Agreement Protocol*. (Cf. p. 39, 42).

# Table des figures

# Liste des tableaux

## RÉSUMÉ

Afin de parer à la menace que ferait peser l'essor des calculateurs quantiques sur nombre de systèmes cryptographiques (et non des moindres) en usage actuellement, d'importants efforts sont faits pour que voient le jour et se développent des systèmes dits « post-quantiques » de chiffrement, d'authentification et d'identification, analogues aux systèmes classiques mais qui résisteraient à des attaques rendues possibles par de tels calculateurs, y compris si ces derniers venaient à devenir particulièrement puissants.
C'est avec l'intention de m'associer à ces efforts que je me suis penché sur les systèmes d'accréditations anonymes. Ces systèmes permettent à un fournisseur de services de vérifier les droits d'un utilisateur à bénéficier de ces services en vertu de la garantie – donnée antérieurement par une autorité de confiance – du fait que cet utilisateur possède bien les attributs requis, cela en préservant le secret des autres attributs, tels que l'identité de l'utilisateur, dont l'autorité aurait eu à prendre connaissance, et cela en permettant la levée de l'anonymat de l'utilisateur par un autre agent, disposant d'une clef secrète spéciale. Je me suis employé à étudier les systèmes d'accréditations anonymes décrits dans la littérature et j'ai participé à l'élaboration d'un schéma théorique d'accréditations anonymes post-quantiques optimisé, fondé sur les problèmes difficiles relatifs aux réseaux euclidiens, pour ensuite contribuer au développement du premier programme informatique à sources ouvertes d'accréditations anonymes post-quantiques qui soit une démonstration directe de la compatibilité d'un tel schéma avec des cas d'usages concrets.
Le protocole Signal, sur lequel se fondent de nombreux systèmes de messagerie instantanée actuels, est lui aussi appelé à connaître une adaptation post-quantique. Dans un article, nous en proposons une vérification formelle réalisée avec le prouveur Tamarin en mettant l'accent sur ses deux principales composantes, le protocole d'échange de clefs X3DH et le protocole du « double cliquet » de mise à jour des clefs de session.

## MOTS-CLEFS

Cryptographie post-quantique, accréditations anonymes, cryptographie asymétrique, chiffrement vérifiable, preuves à divulgation nulle de connaissance, réseaux euclidiens.

## ABSTRACT

In order to mitigate the potential threat that the rise of quantum computers could pose to many of the cryptographic systems currently in use, significant efforts are being made to develop so-called "post-quantum" encryption, authentication, and identification systems. These systems are analogous to classical ones but would withstand attacks made possible by such quantum computers, even if these were to become particularly powerful.
It was with the intention of contributing to these efforts that I delved into anonymous credential systems. These systems allow a service provider to verify a user's rights to access these services, based on a guarantee previously given by a trusted authority, that the user indeed possesses the required attributes, while preserving the secrecy of other attributes such as the user's identity which the authority would have had to acknowledge, and also allowing for the user's anonymity to be lifted by another actor, possessing a special secret key. I have endeavoured to study anonymous credential systems described in the literature and participated in the development of an optimized post-quantum anonymous credential scheme, based on hard problems related to lattices, and then contributed to the first open-source implementation of a post-quantum anonymous credential system that is a direct demonstration of the compatibility of such a scheme with real use cases.
The Signal protocol, upon which many current instant messaging systems are based, is also expected to undergo a post-quantum adaptation. In a paper, we propose a formal verification carried out with the Tamarin prover, placing emphasis on its two main components, the X3DH key agreement protocol and the "double ratchet" session key management protocol.

## KEYWORDS

Post-quantum cryptography, anonymous credentials, public-key cryptography, verifiable encryption, zero-knowledge proofs, lattices.