Lattice-based Signatures in the Fiat-Shamir Paradigm

Julien Devevey


THESE

en vue de l'obtention du grade de Docteur, délivré par

l'ECOLE NORMALE SUPERIEURE DE LYON

Ecole Doctorale N° 512
École Doctorale en informatique et mathématiques de Lyon

Discipline : Informatique

Soutenue publiquement le 18/09/2023, par :

Julien Devevey

Signatures Fondées sur les Réseaux Euclidiens dans le Paradigme de Fiat-Shamir

---

Devant le jury composé de :

ALBRECHT, Martin, Professeur, King's College London	Rapporteur
LYUBASHEVSKY, Vadim, Professeur, IBM Research Zürich	Rapporteur
FOUQUE, Pierre-Alain, Professeur des universités, Université Rennes 1	Examinateur
KIRSHANOVA, Elena, Chercheur, Technology Innovation Institute	Examinatrice
PASSELÈGUE, Alain, Chargé de recherche, ENS de Lyon	Examinateur
ROSSI, Melissa, Chercheur, ANSSI	Examinatrice
STEHLÉ, Damien, Professeur des universités, ENS de Lyon	Directeur de thèse

Résumé

Le paradigme de Fiat-Shamir permet la création systématique de schémas de signature à partir de protocoles d'identification. Ces derniers sont des protocoles interactifs en trois messages, entre un prouveur possédant une information secrète et un vérificateur avec une information publique corrélée. La transformation de Fiat-Shamir est simple et flexible, ce qui explique sa popularité, notamment lorsqu'elle est combinée avec des protocoles d'identification efficaces. Malheureusement, son utilisation dans le cadre de la cryptographie fondée sur les réseaux Euclidiens se révèle plus compliquée, et on lui préfère sa variante dite avec rejet. Cette thèse propose une étude de cette variante et de son utilisation en conjonction avec les réseaux Euclidiens. Nous commençons par recouvrer sa sécurité après avoir identifié des erreurs dans les précédentes preuves de sécurité proposées. On étudie aussi le temps d'exécution d'une signature ainsi que les propriétés de non-divulgation de connaissance du protocole de Lyubahsevsky, qui est le plus célèbre à reposer sur les réseaux Euclidiens.

Nous l'étudions ensuite plus en détail. Ce protocole repose sur l'échantillonnage par rejet, qui peut être utilisée avec une très grande variété de paires "proches" de distributions : on échantillonne depuis l'une pour rejeter vers l'autre. Étant donné un nombre moyen d'itérations du protocole, nous minimisons la norme Euclidienne moyenne de la signature résultante, en choisissant soigneusement la paire de distributions source et cible. Cela diminue la taille de la signature, et la rend plus résistante aux attaques, ce qui permet de réduire les paramètres pour la même sécurité.

Nous proposons ensuite HAETAE, une implémentation de la version bimodale avec des distributions uniformes sur des boules Euclidiennes de la signature de Lyubashevsky. Si nous la comparons rapidement avec les signatures réseaux sélectionnées par le NIST pour être standardisées, on s'aperçoit que nous avons des tailles de signatures près de 40% plus petite que Dilithium, même si nous signons jusqu'à 8 fois plus lentement. Cela reste pourtant plus grand en taille mais plus rapide en vitesse de signature que Falcon. De plus, notre implémentation est en temps constant et n'utilise que de l'arithmétique à virgule fixe.

Enfin, nous proposons d'éviter l'échantillonnage par rejet en utilisant à la place des convolutions de Gaussiennes discrètes. Contrairement aux autres techniques connues, cette solution semble préserver les tailles de signatures et de clés de vérification du protocole pour des paramètres concrets, et elles sont même meilleures asymptotiquement. Bien que cette technique requiert d'échantillonner à partir d'une Gaussienne discrète elliptique, celle-ci est indépendante du message. Pour corriger

cette distribution du premier message, un centre aléatoire est choisi et le résultat sera indépendant du secret, ce qui est le problème critique qui avait mené à l'introduction du rejet en premier lieu.

Abstract

The Fiat-Shamir paradigm enables the systematic design of signature schemes from identification protocols. The latter are three-moves interactive protocols between a prover holding onto some secret information and a verifier with a correlated public information. There are efficient and elegant designs for them and as the Fiat-Shamir transform is simple and flexible, its popularity is easily explained. Unfortunately, its use in the lattice setting turns out to be more difficult and researchers usually rely on a variant, the Fiat-Shamir with aborts transform. This thesis proposes a study of this paradigm and its use in conjunction with Euclidean lattices. First, we recover the security of this transform after identifying flaws in the previous security reductions. This is also a pretense to study the runtime of the signature as well as discuss the zero-knowledge flavor satisfied by Lyubashevsky's scheme, the most famous aborting identification protocols based on Euclidean lattices.

We then move on to study it in more details. This scheme relies on a generic technique called rejection sampling, and can be adapted to work with a wide range of pairs of distributions to sample from and to reject to, as long as these two distributions are "close". Under a target average number of iterations of the scheme, we minimize the average Euclidean norm of the final signature, by wisely choosing the pair of source and target distributions. This directly decreases the size of the signature and also makes forgery attacks harder, allowing for smaller parameter choices.

Next we propose HAETAE, an implementation of the bimodal, uniform over Euclidean balls, version of Lyubashevsky's signature scheme. Let us quickly compare it with the two lattice-based signature schemes selected by NIST for standardisation. Contrary to Dilithium, which aimed for easy implementation, we aimed for low signature sizes. This results in a scheme with up to 40% smaller signature sizes than Dilithium but up to 8 times signing runtime. When compared to Falcon, however, we still have bigger signature sizes but lower signing runtime. Our implementation is constant-time and relies only on fixed-point arithmetic.

Finally, we propose a novel way to avoid rejection sampling by using discrete Gaussian convolutions. Contrary to flooding, this solution appears to preserve the signature and verification key size of the signature scheme and can even be proven to be asymptotically smaller. While it requires sampling from an elliptical discrete Gaussian, this distribution can be made independent from the message. Our technique can be seen as a generalisation of the bimodal technique, where instead of relying on two centers, we randomly pick one over a line with a discrete Gaussian distribution. This center corrects the distribution of the first message, making the last message independent from the secret, which was the critical problem that led to the introduction of rejection sampling in the first place.

Remerciements
Acknowledgements

Pour commencer ces remerciements, quoi de plus naturel que de me tourner vers Damien, qui m'a encadré pendant ces trois années. Merci pour ton temps, ta patience et ton énergie, qui m'ont toujours poussé en avant pour donner le meilleur de moi-même. Merci aussi à Alain, qui s'est énormément investi et a apporté de nombreux conseils et idées. Merci d'avoir relu mon introduction, tâche ingrate s'il en est.

I am deeply grateful to Martin and Vadim for accepting the task of reviewing this manuscript, which is a lot of work. I am fortunate that they agreed to it. I also extend my thanks to Pierre-Alain, Elena and Mélissa for agreeing to be in my jury.

Je tiens aussi à remercier Benoît pour m'avoir ouvert les portes du LIP en M2, ce qui déboucha sur cette thèse. Un grand merci à Omar, qui outre son rôle de tuteur pendant mes années de normalien, nous apporté du recul lorsque nous en manquions.

I am thankful to the whole Haetae team. Collaborating with you has been a pleasure and I look forward to keep improving Haetae in the future with you.

Mes pensées se tournent ensuite vers tous les doctorants de l'équipe AriC : Calvin, Joël, Mahshid, Pouria, Alaa et Arthur (par ordre d'arrivée !). Votre humour, votre gentillesse et votre amitié ont rendu toutes les pauses, goûters et soirées très agréables. Je n'oublie pas non plus toutes les fois où vos conseils m'ont aiguillé. Je pense aussi aux autres membres du LIP que j'ai fréquentés, Emily, Chen, Paul et ses parties de Magic, et aux anciens du LIP, Fabrice, Octavie et Alice, qui m'a appris qu'après les deadlines, vient le temps de faire une review. Je ne saurais assez remercier Chiraz et les Malip, dont leurs grandes efficacité et réactivité sont à l'épreuve du doctorant le plus ignorant des règles et délais administratifs.

Je tiens à remercier mes amis, qu'ils soient de Montbrison, des Lazaristes ou de l'ENS, qui ont toujours cru en moi et m'ont supporté toutes ces années. J'en profite aussi pour remercier ma belle-famille de l'accueil chaleureux qu'elle m'a réservé. Et qui serais-je sans le soutien indéfectible de ma famille, proche et éloignée ? Les cousinades et autres événements familiaux de ces trois dernières années ont toujours été une bouffée d'oxygène pour moi. Je termine ces remerciements en évoquant ma femme, Héloïse, qui a toujours su trouver les mots, dans les moments difficiles comme dans les bons. Nos parcours très similaires vont pour la première fois significativement diverger. J'ai hâte de commencer cette nouvelle aventure à ses côtés.

Contents

Résumé substantiel en Français

La cryptographie moderne a pour but de fournir un ensemble de primitives et protocoles sécurisés correspondant à n'importe quelle situation où une forme de sécurité est requise lors de communications. Parmi eux, les protocoles de signature numérique permettent à un signataire, en possession d'une clé secrète, de produire une signature pour un message quelconque. N'importe qui en possession de la clé de vérification correspondante peut vérifier la cohérence de la signature produite avec le message. De plus, il est difficile pour quelqu'un sans la clé de signature de contrefaire une signature pour un autre message de son choix en un temps raisonnable. Depuis leur introduction par Diffie et Hellman en 1976 [DH76], de nombreuses applications reposent sur les signatures numériques et il est difficile de surestimer leur importance. Le premier cas d'usage, et peut-être le plus évident, est en tant que remplacement des signatures manuscrites. Plusieurs pays, notamment les États Unis et les membres de l'Union Européenne, accordent une valeur légale aux signatures numériques dans plusieurs situations, telles que la signature de contrat entre entreprises. Une seconde application est la prévention d'usurpation d'identité lors de la mise à jour d'un logiciel. Plus précisément, un développeur peut adjoindre une clé de vérification à un logiciel lors de sa sortie initiale. Lors de mises à jour futures, il signera celles-ci, ce qui permettra aux utilisateurs d'être convaincus quant à l'intégrité de la mise à jour qu'ils viennent de télécharger.

Plus généralement, dans le contexte du chiffrement à clé publique, les signatures numériques permettent de distribuer les clés publiques nécessaires pour mettre en place un canal de communication sécurisé. En effet, lors de la transmission d'une telle clé publique, on peut démontrer son authenticité en la signant, en supposant que le destinataire possède notre clé de vérification pour la signature. C'est peut-être une fuite en avant mais il faut pourtant reconnaître que ce problème de l'échange de clé n'apparaît maintenant plus qu'une seule fois, lors de l'échange des clés de vérification. Il devient ensuite possible d'utiliser autant de clés publiques que nécessaire, notamment en cas de compromission de session. C'est d'ailleurs le rôle des autorités numériques et des certificats numériques, qui peuvent être vus comme la "carte d'identité" d'un ordinateur. En signant une clé de vérification en plus d'une adresse IP ou même physique, ces autorités certifient l'origine d'une clé de vérification, qui peut ensuite être librement utilisée ailleurs.

Un exemple crucial est la poignée de main TLS. Pendant celle-ci, un client et un serveur tentent d'établir un canal de communication sécurisé (HTTPS) en générant

ensemble une clé symétrique. Pour commencer, ils échangent leurs certificats, avec les clés de vérifications correspondantes. Ils peuvent ensuite les utiliser pour échanger une clé publique, qui leur permettra enfin d'échanger secrètement une clé symétrique. Au total, trois signatures et deux clés de vérification sont échangées pendant ce protocole, ce qui représente de nos jours quelques kilooctets de données transmises. En perspective, le poids moyen d'une page web atteint 2,3 Mégaoctets en mai 2023[1]. Ainsi, le coût actuel de ce protocole est infime par rapport au coût du téléchargement d'une page web.

Cependant, la sécurité des standards actuels repose sur la difficulté à factoriser de très grands entiers, ou encore la difficulté à calculer le logarithme discret dans un groupe. Or, ces deux problèmes sont menacés par l'avènement du calcul quantique, car ils sont résolus en temps polynomial grâce à l'algorithme de Shor. L'institut national américain des standards et de la technologie (NIST) a lancé en 2016, comme mesure préventive, une compétition pour trouver de nouveaux standards, ceux-ci résistants aux attaques par ordinateur quantique. Ces nouveaux standards doivent allier sécurité sur le long terme, efficacité calculatoire et tailles de signature raisonnables.

Différentes familles d'hypothèses sont pour le moment supposées résister aux algorithmes quantiques. Parmi elles, les problèmes reposant sur les réseaux Euclidiens, ou sous-groupes discrets de $\mathbb{R}^n$, sont particulièrement remarquables par leur grande flexibilité et leurs bonnes performances, qui sont encore accrues quand on les considère dans la version "module". Les problèmes *Short Integer Solution* (SIS) et *Learning with Errors* (LWE) sont d'un intérêt particulier pour la cryptographie. Le premier consiste à trouver un petit vecteur (pour la norme Euclidienne) dans le noyau d'une matrice $\mathbf{A}$ sur un groupe quotient $\mathbb{Z}/q\mathbb{Z}$, tirée uniformément. Le second consiste, étant donné $\mathbf{A}$ et un vecteur $\mathbf{b}$, à retrouver si $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ pour de petits $\mathbf{s}$ et $\mathbf{e}$, ou si $\mathbf{b}$ a été tiré uniformément.

Certains choix de paramètres bénéficient de réductions pire cas-moyen cas, renforçant la confiance en la difficulté de ces problèmes, telles que prouvées pour la première fois dans [Ajt96, Reg09]. De plus, pour les paramètres considérés en pratique, malgré plusieurs décennies de cryptanalyse, aucun algorithme quantique n'arrive à les résoudre efficacement. Ces arguments, parmi d'autres, ont conduit le NIST à choisir deux protocoles de signature fondés sur les réseaux, parmi les trois protocoles qui seront standardisés. Le premier, Falcon [FHK+17], découle d'une branche dénommée "*Hash-and-Sign*" (hacher puis signer). Le second, Dilithium [BDK+20], est construit à partir du paradigme de Fiat-Shamir. C'est à celui-ci que nous allons nous intéresser dans cette thèse. Il a été introduit par Fiat et Shamir en 1987 [FS87] et offre de multiples avantages : il est flexible, efficace et part d'une primitive plus simple conceptuellement. Nous allons maintenant brosser un bref portrait des innovations qui ont abouti à Dilithium.

Une Brève Histoire de Fiat-Shamir et des Réseaux

Le principal ingrédient de la transformation de Fiat-Shamir est un protocole d'identification. C'est un protocole interactif entre un prouveur, qui possède une information

[1] `https://httparchive.org/reports/page-weight?start=2023_03_01&end=latest&view=list`

secrète, et un vérificateur détenant une information publique corrélée. L'objectif du prouveur est de convaincre le vérificateur qu'il est bel et bien en possession de l'information secrète, sans la révéler. Cet objectif peut être réalisé en trois étapes. De tels protocoles peuvent être réalisés de manière simple et pourtant efficace, ce qui permettra ensuite d'obtenir une signature hautement efficace. Le protocole de Schnorr pour le logarithme discret est particulièrement intéressant dans notre cas, et nous verrons comment il est adapté au cas des réseaux.

Le Protocole de Schnorr pour le Logarithme discret

Étant donné un groupe cyclique $\mathbb{G}$ d'ordre p très grand, un générateur g ainsi qu'un élément quelconque du groupe g^x, le problème du logarithme discret consiste à calculer x. En supposant que ce problème est difficile, on comprend qu'on peut alors cacher des informations à l'intérieur de l'exposant, tandis que la cohérence des calculs peut être vérifiée en manipulant les éléments du groupe.

Le protocole de Schnorr [Sch91] profite de cette flexibilité pour être efficace et sécurisé. La clé publique du vérificateur est un élément du groupe g^s, tandis que la clé secrète du prouveur est son logarithme discret s. L'interaction se déroule alors comme suit. D'abord, le prouveur échantillonne $y \hookleftarrow U(\mathbb{Z}_p)$ et envoie g^y comme engagement. Il reçoit ensuite un défi c échantillonné uniformément sur $\mathbb{Z}_p$. Sa réponse est enfin $z = y + sc \bmod p$. Notons ici qu'aucune information n'est révélée sur s. En effet, comme y est caché au vérificateur (à moins qu'il ne résolve une instance du problème du logarithme discret), la réponse z est a priori uniforme et indépendante du secret. Cependant, il est convaincu que le prouveur connaît s en vérifiant la véracité de l'équation $g^z = g^y(g^s)^c$.

La Transformée de Fiat-Shamir

Nous décrivons maintenant la technique utilisée par Fiat et Shamir [FS87] pour transformer un protocole d'identification en une véritable signature. L'algorithme de génération de clé est exactement le générateur d'instance du protocole d'identification. L'algorithme de signature va dérouler le protocole interactif en jouant le rôle à la fois du prouveur et du vérificateur. Pour éviter qu'un adversaire ne puisse contrefaire une signature, le défi est calculé de manière déterministe en hachant l'engagement et le message. La signature est alors la retranscription complète de l'échange. Enfin, l'algorithme de vérification n'accepte que si deux conditions sont réunies. D'abord, le vérificateur doit accepter la retranscription. Ensuite, le défi doit avoir été calculé honnêtement : l'algorithme de vérification re-hache le message et l'engagement et vérifie que le haché obtenu est le même que le défi.

Heuristiquement, utiliser une fonction de hachage "force" le signataire à utiliser un défi uniforme, qui peut être vérifié publiquement, afin de l'empêcher de le modifier selon ses besoins. On peut alors prouver la sécurité de la signature dans le modèle de l'oracle aléatoire (la fonction de hachage est remplacée par une fonction dont les sorties sont tirées uniformément et qui est gérée par le challenger, l'attaquant n'ayant qu'un accès à celle-ci par "oracle"). Pour cela, il est nécessaire que le protocole d'identification satisfasse les propriétés suivantes :

- Min-entropie de l'engagement : si le protocole est utilisé à plusieurs reprises, la probabilité d'utiliser plusieurs fois le même engagement doit être négligeable. Dans le cas de Schnorr, c'est le cas comme l'ordre du groupe est grand.

- Sans Divulgation de Connaissance pour un Vérificateur Honnête (HVZK) : un vérificateur honnête ne devrait rien apprendre à propos du secret détenu par le prouveur. Cela se traduit par l'existence d'un simulateur capable de générer une retranscription, étant donné un défi, tel que la distribution de cette retranscription est proche de celle d'une vraie retranscription, conditionnée sur ce défi. Le simulateur pour Schnorr commence par échantillonner uniformément $z \hookleftarrow U(\mathbb{Z}_p)$ et pose $g^y = g^z(g^s)^{-c}$ comme engagement. On remarque alors que la distribution de cette simulation est exactement celle d'une véritable retranscription.

- Sûreté : il doit être difficile de convaincre le vérificateur pour une personne n'ayant pas le secret. Grâce à une technique dite de rembobinage, on peut montrer qu'un attaquant pouvant convaincre le vérificateur une fois peut en fait le convaincre deux fois, pour le même engagement mais deux défis différents. On obtient alors (g^y, c, c', z, z') tel que $g^y = g^z(g^s)^{-c} = g^{z'}(g^s)^{-c'}$ et on retrouve $s = (z - z')/(c - c') \bmod p$.

La réduction de sécurité utilise un jeu hybride, où le challenger rend les signature indépendantes de la clé secrète en les simulant à l'aide du simulateur. Il doit aussi reprogrammer l'oracle aléatoire pour qu'il reste cohérent avec le défi qu'il aura échantillonné. Il devient donc impossible pour un adversaire d'apprendre quoi que ce soit au sujet de la clé secrète à partir de signatures. Enfin, on peut utiliser cet adversaire pour contredire la propriété de sûreté du protocole d'identification, en construisant un attaquant contre la sûreté qui joue le rôle du challenger pour le premier adversaire, sans avoir besoin de la clé secrète. Il doit par contre reprogrammer la fonction de hachage au bon endroit afin qu'il puisse utiliser la signature contrefaite pour convaincre le vérificateur.

Notons que le protocole de Schnorr satisfait une propriété supplémentaire : le recouvrement de l'engagement, ce qui sera le cas pour les autres protocoles considérés dans cette thèse. En effet, si l'on omet l'engagement de la signature, on peut le recalculer en sachant que $g^y = g^z(g^s)^{-c}$. Cela permet de réduire considérablement la taille de la signature. La vérification va alors commencer par recouvrer l'engagement, avant de vérifier la cohérence du haché, et enfin de vérifier que la retranscription est convaincante. Dans le cas de Schnorr, on peut omettre cette dernière étape car elle sera automatiquement valide, vu la manière dont on a récupéré l'engagement.

Du Log Discret aux Réseaux Euclidiens : Introduction de l'Échantillonnage par Rejet

Un but de longue date dans la cryptographie à base de réseaux Euclidiens est de proposer une adaptation efficace du protocole de Schnorr au contexte des réseaux. Lyubashevsky en a proposé la première [Lyu09,Lyu12]. Celle-ci implique une matrice publique $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, qui remplace le générateur g et définit une instance SIS. La clé secrète est une matrice $\mathbf{S} \in \mathbb{Z}^{m \times k}$, dont chaque colonne a une petite norme Euclidienne, par rapport à q. En remplacement de g^s, la clé de vérification est $\mathbf{T} = \mathbf{AS} \bmod q$.

Dans le protocole d'identification, le prouveur échantillonne un petit vecteur de masquage $\mathbf{y} \in \mathbb{Z}^m$ et calcule un engagement pseudo-uniforme $\mathbf{w} = \mathbf{A}\mathbf{y} \bmod q$. Il reçoit ensuite un défi $\mathbf{c} \in \mathbb{Z}^k$ choisi uniformément parmi les vecteurs de norme infinie bornée par un paramètre du protocole. Enfin, après un test, dont nous parlerons plus tard, il renvoie $\mathbf{z} = \mathbf{y} + \mathbf{S}\mathbf{c}$ si celui-ci est réussi, sinon, il faut recommencer. Étant donné une retranscription $\sigma = (\mathbf{w}, \mathbf{c}, \mathbf{z})$, le vérificateur accepte si $\mathbf{A}\mathbf{z} - \mathbf{T}\mathbf{c} = \mathbf{w} \bmod q$, et si $\mathbf{z}$ est petit, ce qui est spécifique aux réseaux. Le lecteur trouvera une description détaillée en Anglais du protocole en Figure 2.5.

Contrairement au protocole de Schnorr, la clé de signature et le masque n'appartiennent pas à un groupe fini, ce qui empêche $\mathbf{y}$ de complètement cacher le terme sensible $\mathbf{S}\mathbf{c}$. Au contraire, en fournissant un effort proportionnel à la valeur moyenne de $\|\mathbf{y}\|$, des attaques statistiques génériques permettent de récupérer la clé secrète. Une possibilité (voir par exemple [DPSZ12]) est de choisir $\mathbf{y}$ exponentiellement plus large que $\mathbf{S}\mathbf{c}$ en tant que fonction du paramètre de sécurité, ce qui contrecarre l'attaque précédente et permet même de prouver la sécurité. En effet, la distance statistique entre $\mathbf{y}$ et $\mathbf{y} + \mathbf{S}\mathbf{c}$ devient alors négligeable. Étant donné que q doit être plus grand que $\mathbf{y}$ et que la petitesse de $\mathbf{S}$ par rapport à q impacte la sécurité, cette approche par *flooding* (noyade) donne de très grands paramètres. À la place, Lyubashevsky propose une notion de Fiat-Shamir avec rejet. C'est ici qu'intervient le test que passe $\mathbf{z}$ dans le protocole. Il sert à faire en sorte que la distribution de $\mathbf{z}$ ne dépende plus de $\mathbf{S}\mathbf{c}$.

Une application classique du rejet (cf. [Dev86, Chapitre 2]) est l'échantillonnage d'une distribution P "difficile" à partir d'une distribution plus simple à échantillonner Q. Ici, le rejet est détourné de son utilité première : on part d'une distribution pré-source Q qu'on décale par $\mathbf{S}\mathbf{c}$, ce qui donne une distribution source $Q_{+\mathbf{S}\mathbf{c}}$. On la rejette ensuite vers une distribution P qui ne dépend pas de $\mathbf{S}\mathbf{c}$. L'objectif ici alors est de cacher $\mathbf{S}\mathbf{c}$. Des choix divers ont été faits jusqu'à présent pour P et Q : uniforme dans des hypercubes [Lyu09], gaussiennes discrètes de même écart type [Lyu12]. L'efficacité de ces choix est contrôlée par deux paramètres : le nombre d'itérations en moyenne M, et la distance statistique ε entre le résultat du rejet et la distribution cible P. Grâce à cette flexibilité, il existe donc une myriade de possibilités pour adapter le protocole de Schnorr aux réseaux Euclidiens. Cela nous mène donc à la question suivante :

Q1. *Étant donné un nombre moyen d'itérations, quelle stratégie de rejet permet d'obtenir les signatures les plus compactes ?*

L'équipe de Dilithium avait choisi de répondre à une autre question, leur but étant de produire une signature "facile à implémenter". Leur choix s'est alors porté sur P et Q uniformes dans des hypercubes, ce qui permet d'échantillonner chaque coordonnée indépendamment dans un intervalle. De plus, la condition de rejet consiste à simplement calculer la norme infinie de $\mathbf{z}$ et à le rejeter si elle est trop grande. La question **Q1** reste alors ouverte.

Nous considérons aussi le cas de BLISS [DDLL13], qui note que le rejet de gaussienne à gaussienne est inexact à cause des queues de celles-ci. Ses auteurs considèrent alors une Gaussienne *bimodale* comme distribution source, ce qui règle ce problème. C'est à dire qu'au lieu de calculer $\mathbf{z} = \mathbf{y} + \mathbf{S}\mathbf{c}$, ils calculent $\mathbf{z} = \mathbf{y} + (-1)^b\mathbf{S}\mathbf{c}$, où b est un bit uniforme. Non seulement le rejet devient exact, mais il devient aussi plus

efficace. Pour le même nombre moyen d'itérations, on peut prendre des écarts-type plus faibles. Malheureusement, le bit b ne peut être révélé à moins de retomber dans le cas *unimodal*. On change alors de module, de q à $2q$, et changer la génération de clé afin que $\mathbf{AS} = q\mathbf{I} \bmod 2q$, pour que la vérification puisse fonctionner quelle que soit la valeur de b.

Nos Contributions

Pour conclure cette introduction, nous allons brièvement présenter les différentes contributions de chaque chapitre de cette thèse.

Chapitre 3 : Une Analyse Détaillée de Fiat-Shamir avec Rejet

Dans un premier temps, nous étudions les preuves de sécurité de la transformation de Fiat-Shamir avec rejet. Les preuves de sécurité proposées, par exemple dans [Lyu12, KLS18, AFLT16], diffèrent des preuves standards pour la transformation de Fiat-Shamir. En effet, si le protocole d'identification doit être appelé à plusieurs reprises avant d'obtenir une signature, ce n'est plus la transformation de Fiat-Shamir mais une transformation différente, que nous dirons de Fiat-Shamir avec rejet. Les preuves doivent alors être modifiées pour prendre ce changement en compte. En particulier, la notion de HVZK peut être abordée de deux manières : doit-on pouvoir simuler les retranscriptions rejetées ? Les réductions précédentes considèrent que non, tandis que nous considérons que oui. Notons au passage qu'elles étudient une version modifiée de la transformation, où le protocole est appelé au plus B fois, quitte à ce que la signature échoue, ce qui ne correspond pas aux signatures utilisées en pratique.

Dans un premier temps, nous avons identifié une erreur commune à toutes les preuves précédentes. Toutes ces preuves considèrent un premier jeu hybride où un défi uniforme est échantillonné et une signature est générée, jusqu'à en avoir une non-rejetée (ou atteindre B itérations), puis l'oracle aléatoire est reprogrammé pour la signature acceptante *uniquement*. C'est un problème, car dans la signature originale, les itérations sont corrélées, à cause de la fonction de hachage, et ici elles ne le sont pas. Contrairement à ce qui est affirmé, on ne peut pas conclure que les deux jeux sont identiques.

Nos contributions portent alors sur trois fronts. Le premier est de corriger ces preuves en considérant que la propriété HVZK doit permettre la simulation de retranscriptions rejetées. Le second est d'étendre les réductions au cas où le nombre d'itérations n'est pas borné, comme c'est le cas en pratique. Nous identifions alors quelques difficultés dans les définitions car un contre-exemple montre les limites de celles-ci pour des questions de temps d'exécution. Enfin, nous montrons que le protocole de Lyubashevsky satisfait notre propriété plus forte de HVZK. Au final, les bornes de sécurité dans les preuves sont quasiment identiques à celles qui étaient mises en avant auparavant.

Chapitre 4 : Optimisation des Protocoles d'Identification fondés sur les Réseaux Euclidiens

Dans ce chapitre, nous étudions l'utilisation de l'échantillonnage par rejet dans les protocoles d'identification fondés sur les réseaux Euclidiens, et nous cherchons à mi-

nimiser la taille (ou compacité) d'une signature, mesurée par la norme Euclidienne moyenne d'un élément, étant donné un nombre d'itérations moyen M donné, ce qui répond à la question **Q1**. Dans un premier temps, nous nous intéressons à l'échantillonnage par rejet lui-même, et nous prouvons que cette procédure est optimale pour minimiser le nombre moyen d'itérations, étant donné un choix de distributions. Ensuite, nous prouvons une borne inférieure dans les cas unimodal et bimodal sur la compacité d'une signature. Pour cela, nous avons besoin de remarquer que la contrainte M sur le nombre moyen d'itérations se traduit par une condition $M \geq R_\infty(P\|Q_{+\mathbf{Sc}})$ sur la divergence de Rényi entre P et n'importe quelle translatée de Q. Cette divergence est définie comme étant $\max_{x\in\mathrm{Supp}(P)} P(x)/Q_{+\mathbf{Sc}}(x)$. En manipulant avec prudence les différentes équations que cela donne, nous trouvons ces bornes inférieures.

Ensuite, nous explorons des instanciations de P et Q afin de trouver des choix pouvant atteindre ces bornes. Nous en isolons deux. Le premier, les gaussiennes discrètes, est un choix standard en cryptographie fondée sur les réseaux. Le second, la distribution uniforme dans une boule Euclidienne, est nouveau. Bien que ces deux choix offrent la même compacité, le second choix est intéressant car la condition de rejet est plus simple : il s'agit de calculer une norme et de vérifier si la signature est suffisamment petite dans le cas unimodal. Enfin, pour les comparer plus en détail, nous proposons plusieurs jeux de paramètres concrets ainsi que des estimations de tailles, qui sont similaires pour ces deux choix de distributions. Elles sont en revanche bien meilleures que pour les distributions uniformes dans des hypercubes, comme considérées dans Dilithium : jusqu'à 40% de gains.

Chapitre 5 : HAETAE, une Nouvelle Implémentation de Fiat-Shamir avec Rejet Fondée sur les Réseaux

Afin de concrétiser les résultats des deux premiers chapitres, nous proposons une implémentation, dénommée HAETAE. Celle-ci implémente la variante bimodale de la signature de Lyubashevsky qui repose sur la distribution uniforme dans des boules Euclidiennes. Dans ce chapitre, nous détaillons principalement les optimisations, largement présentes déjà dans Dilithium [BDK+20], que nous adaptons au cas bimodal. Nous présentons aussi les différents choix qui permettent de passer d'une distribution théorique à une implémentation concrète d'un échantillonneur ainsi que d'une condition de rejet. En particulier, l'un des défis d'une telle implémentation est de permettre l'utilisation d'arithmétique à virgule fixe tout au long de la procédure de signature. Finalement, nous présentons brièvement les performances et paramètres de l'implémentation.

L'équipe de HAETAE est composée d'une dizaine de membres et les informations complémentaires du projet peuvent être trouvées sur la page web[2]. Notons que cette signature est soumise à deux concours de standardisation, l'un organisé par la Corée du Sud et l'additionnel quatrième tour des signatures post-quantiques du NIST.

Cette signature étant le résultat d'un travail d'équipe, ce chapitre se focalisera sur les parties où mon implication a été la plus forte, tout en n'en omettant aucun aspect. Il en résulte que, bien que HAETAE soit plus compliqué conceptuellement que Dilithium, le gain en terme de tailles de signature et de clé de vérification est

[2]https ://kpqc.cryptolab.co.kr/haetae

largement en notre faveur : jusqu'à 40% de gain pour les signatures. D'un autre côté, par rapport à l'autre finaliste du NIST reposant sur les réseaux, Falcon, HAETAE reste conceptuellement plus simple, est plus rapide, mais a des tailles de signature et clé de vérification plus grandes.

Chapitre 6 : G+G, un Premier Pas vers un Fiat-Shamir Efficace Fondé sur les Réseaux

Dans ce dernier chapitre, nous explorons une solution permettant d'éviter l'échantillonnage par rejet, qui consiste à utiliser une propriété de convolution des gaussiennes discrètes. On note que dans le mode bimodal, la relation vérifiée par les clés de vérification et secrète est $\mathbf{AS} = q\mathbf{J} \bmod 2q$. Or, le défi $\mathbf{c}$ est toujours multiplié par $\mathbf{S}$ à gauche dans la signature, ce qui signifie que lors de la vérification, le secret est réduit modulo 2. C'est cette remarque qui permet au mode bimodal de fonctionner, en prenant un représentant de $\mathbf{c}$ qui est soit $\mathbf{c}$ soit $\mathbf{c} - 2\mathbf{c}$. Si on cherche à aller encore plus loin, on peut tirer un représentant de $\mathbf{c}$ suivant une gaussienne discrète sur $2\mathbb{Z}^k + \mathbf{c}$ sans centre de matrice de covariance $s^2\mathbf{I}_k$. Maintenant, si $\mathbf{y}$ est échantillonné suivant une gaussienne discrète de covariance $\sigma^2\mathbf{I}_n - s^2\mathbf{S}\mathbf{S}^\top$, il résulte que $\mathbf{z} = \mathbf{y} + \mathbf{S}\mathbf{c}'$ suit une Gaussienne discrète sphérique sans centre.

Nous proposons alors $\mathsf{G} + \mathsf{G}$, un protocole d'identification fondé sur les réseaux Euclidiens, qui ne nécessite pas de rejet, et qui peut être transformé en une signature en utilisant la transformation de Fiat-Shamir (sans rejet). Le retrait du rejet est quelque chose qui avait déjà été exploré précédemment [DPSZ12], mais le résultat était loin d'être pratique au vu des tailles qui en résultent. Ici, nous montrons que les tailles sont non seulement similaires à HAETAE pour des jeux de paramètres concrets, mais aussi les tailles de signature asymptotiques sont meilleures d'un facteur $\sqrt{\lambda/\log(\lambda)}$ par rapport aux meilleurs signatures avec rejet.

Chapter 1

Introduction

Modern cryptography aims at providing users with a toolkit of secure primitives and protocols which suits any communication setting where some form of security is desired. Among them, digital signature protocols allow a signer in possession of a secret key to produce a signature for any input message. Anyone in possession of the corresponding verification key may check the consistency of the signature with the message. Moreover, security guarantees prevent anyone from forging another signature for any message of their choice without the signing key in a reasonable time. Since their introduction by Diffie and Hellman in 1976 [DH76], digital signatures have found applications in numerous domains and it is hard to overstate their importance. The most obvious use case is as replacement for written signatures. Multiple countries, including members of the European Union and the USA, give legal value to digital signatures for contracts or other business applications. Another application is to prevent spoofing for software updating. Namely, a company bundles a verification key with a software in its initial release, and signs further updates. Users are then assured of the integrity of the update they downloaded, i.e., it has not been tampered with.

More broadly, in the context of public key encryption, digital signatures allow for secure distribution of public keys. When broadcasting a public key to someone in possession of the verification key, signing it allows to demonstrate its authenticity. Granted, this pushes back the problem one step before, as the verification key of the signature needs to be distributed beforehand. However, this problem now only happens at most once: with one verification key, one may then broadcast as many public keys as necessary. Digital authorities also use signatures to issue digital certificates. By signing an e-mail or an address, digital certificates can be seen as "identity cards" of servers and computers across the internet. They also embed a verification key to enable the previous usage.

This is particularly visible during TLS handshakes. In those, a client and a server come together to exchange a symmetric key, thus enabling secure communication and HTTPS web browsing. They first exchange their certificates, with the corresponding verification keys. They use the latter to sign the further messages they send each other in order to agree on a symmetric encryption key. In total, three signatures and two verification keys are sent, which amounts to a few kilobytes of data transmission nowadays. To put it into perspective, the average website page weighs 2.3MB as of

May 2023[1]. Thus the cost of signing is but a negligible fraction of the total cost of connecting to a website.

However, current standards rely on the hardness of factoring large integers or the hardness of computing the discrete logarithm in a group. With the ever-rising threat of quantum computing, the long term security of current standards for digital signatures is at stake, as these two problems are broken in polynomial time with Shor's algorithm. As a preventive measure, the American National Institute of Standards and Technology (NIST) launched in 2016 a competition to look for new, quantum-resistant, standards that should ally long-term security, computing efficiency and reasonable signature size.

Different families of assumptions are currently assumed to resist to quantum attacks. Among them, problems based on Euclidean lattices (discrete subgroups of $\mathbb{R}^n$) are particularly remarkable for the great design flexibility and good performance they offer, even more so when considered in their so-called "module" version. Of particular interest are the Short Integer Solution (SIS) problem and the Learning with Errors (LWE) problem. The former asks, given a matrix $\mathbf{A}$ over a finite field, to find a short (for the Euclidean norm over representatives centered around 0) element in its kernel. The latter asks to distinguish between $\mathbf{As} + \mathbf{e}$ for short, random $\mathbf{s}$ and $\mathbf{e}$ and a uniform vector over the finite field, given $\mathbf{A}$.

Certain settings benefit from worst-case to average-case reductions, strengthening the belief that those problems are hard to solve, as first shown in [Ajt96, Reg09]. Moreover, for settings considered in practice, despite more than a decade and a half of research, no polynomial time quantum algorithm can solve the problems yet. These arguments, among others, led the NIST to elect, out of the three winners of the competition for standardization of post-quantum digital signatures, two lattice-based schemes. The first one, Falcon [FHK+17], results from a first line of signatures, following the "Hash-and-Sign" design. The second one, whose line of work we are interested in in this thesis, is called Dilithium [BDK+20]. It relies on a paradigm introduced by Fiat and Shamir in 1987 [FS87], which offers multiple advantages: it is flexible, efficient and starts from a conceptually easier building block. We now briefly recall the results that ultimately led to Dilithium.

1.1 Brief History of Fiat-Shamir and Lattices

The main building block of the Fiat-Shamir transform is called an identification protocol. It is an interactive protocol between a prover, which holds some secret information, and a verifier, holding some related public information, where the prover tries to convince the verifier that it knows the secret information, without revealing it. The interaction is 3-round: the prover first sends a first message, the "commitment", the verifier replies with a "challenge" and the prover answers with a "response". The verifier checks the whole transcript and decides if it is convinced or not. Moreover, there is an efficient instance generator, which samples a random instance of related secret and public information. Simple yet efficient designs for such protocols arose, which in turn led to highly efficient and versatile signature schemes. We are

[1] `https://httparchive.org/reports/page-weight?start=2023_03_01&end=latest&view=list`

particularly interested in Schnorr's protocol for the discrete logarithm, which is then adapted to the lattice setting.

1.1.1 Schnorr Protocol for Discrete Logarithm

The discrete logarithm problem asks one, given a cyclic group $\mathbb{G}$ of large order p, a generator g and an element of the group g^x, to compute x. Based on the difficulty of solving this problem, one can hide secret information in the exponent, while the consistency of said private information can be checked through various operations over the group.

Schnorr's identification protocol [Sch91] takes advantage of this flexibility to offer an efficient design. The prover's public verification key is simply a group element g^s, whose discrete logarithm s forms the prover's secret key. The identification protocol proceeds as follows: the prover first commits to some uniform $y \hookleftarrow U(\mathbb{Z}_p)$ by sending g^y to a verifier. The latter returns some uniform challenge $c \in \mathbb{Z}_p$, to which the prover replies with the response $z = y + cs \bmod p$. Here, no information about s is revealed as z is still uniform modulo p as y is hidden from the verifier. Indeed, it cannot compute the discrete logarithm of g^y by assumption. However, a verifier is convinced that the prover knows s as it can verify $g^z = g^y(g^s)^c$.

1.1.2 The Fiat-Shamir Transform

We now describe the technique Fiat and Shamir used [FS87] to turn an identification protocol into a full-fledge signature scheme. The key generation algorithm is the instance generator of the identification protocol. The signing algorithm runs the interactive protocol. Instead of sampling a uniform challenge, which an attacker could maliciously choose, it hashes the commitment along with the message and obtains the challenge. The signature is comprised of the whole transcript. The verification algorithm accepts if the challenge is consistent with the hash of the commitment and the message, and if the verifier of the identification protocol accepts the transcript.

Heuristically, relying on the hash function "forces" the signer to use a uniform challenge, which can be publicly verified, to prevent it from tampering with it. The resulting signature can be proven unforgeable in the random oracle model (where the hash function is replaced with a function with uniform outputs) under the assumption that the identification protocol satisfies a few properties. Namely:

- Commitment min-entropy: if the protocol is ran multiple times, the probability of using the same commitment twice should be negligible. In the case of Schnorr's protocol, since the group order is large, g^y has large min-entropy.

- Honest Verifier Zero-Knowledge (HVZK): a honest verifier should not learn anything about the secret held by prover. In particular, if the challenge is known in advance, transcripts can be simulated by someone who does not hold the secret key. The distribution of the simulation must be statistically close to the real one. Schnorr's simulator samples a uniform $z \hookleftarrow U(\mathbb{Z}_p)$ and sets $g^y = g^z(g^s)^{-c}$, which does not require knowing the secret key. Moreover, the distribution of the transcript is exactly the same as a real one.

- Soundness: it must be hard for someone who does not hold the secret key to convince the verifier. Convincing Schnorr's verifier without knowing s implies computing the discrete logarithm of g^s via rewinding techniques, which yields two accepting transcripts with the same commitment but different challenges. This gives (g^y, c, c', z, z') such that $g^y = g^{z'}(g^s)^{-c'} = g^z(g^s)^{-c}$. The discrete logarithm of g^s is then $s = (z - z')/(c - c') \bmod p$.

The security reduction uses an hybrid game, where the signatures are independent from the secret key, rendering impossible to learn the secret key by observing signatures. To do so, the challenger generates a signature using the HVZK simulator, and then reprograms the random oracle accordingly, as it manages it. From here, the unforgeability of the signature scheme is reduced to the soundness of the identification protocol.

Schnorr's identification protocol further satisfies the commitment-recoverability property, as do all other identification protocols discussed in this thesis. Given the challenge c and its answer z, one can recover the commitment $g^y = g^z(g^s)^{-c}$. This is used to reduce the signature size by omitting the commitment from the signature. Verification is achieved by recovering it during verification and checking the consistency of the challenge with the hash function.

1.1.3 From Discrete Logarithm to Lattices: Introduction of Rejection Sampling

A long-standing goal in lattice-based cryptography is proposing efficient adaptations of Schnorr's protocol to lattices. Lyubashevsky proposed famous adaptations [Lyu09, Lyu12] of this protocol for lattices. Lyubashevsky's scheme involves a publicly shared matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ (note that other algebraic setups are possible, but this is not relevant to the present discussion), which replaces the group generator g and defines a SIS instance. The secret key is a matrix $\mathbf{S} \in \mathbb{Z}^{m \times k}$. It is small in the sense that all its entries have absolute values significantly smaller than q. Instead of a group element whose discrete logarithm is s, the verification key associated to $\mathbf{S}$ is $\mathbf{T} = \mathbf{AS} \bmod q$. In the identification protocol, the prover samples a small masking vector $\mathbf{y} \in \mathbb{Z}^m$ and computes a random-looking commitment $\mathbf{w} = \mathbf{Ay} \bmod q$. It receives a uniform $\mathbf{c} \in \mathbb{Z}^k$ challenge with small values from the verifier. Finally, if some (possibly probabilistic, later defined) test passes, it outputs $\mathbf{z} = \mathbf{y} + \mathbf{Sc}$, and else it restarts from scratch. Given a transcript $\sigma = (\mathbf{w}, \mathbf{c}, \mathbf{z})$, the verifier accepts if and only if $\mathbf{Az} - \mathbf{Tc} = \mathbf{w} \bmod q$, as in Schnorr's protocol, and if $\mathbf{z}$ is small, which is particular to the lattice setting. We refer the reader to Figure 2.5 for a formal description.

Compared to Schnorr's signature scheme, the signing key and mask do not belong to a finite set, preventing the use of a uniform mask $\mathbf{y}$ to hide the sensitive term $\mathbf{Sc}$.[2] On the contrary, with an effort proportional to the expected value of $\|\mathbf{y}\|$, generic statistical attacks allow for key recovery. One possibility (see, e.g., [DPSZ12]) is to sample $\mathbf{y}$ exponentially larger than $\mathbf{Sc}$ as a function of the security parameter, so that the distributions of $\mathbf{y}$ and $\mathbf{y} + \mathbf{Sc}$ have exponentially small statistical distance.

[2] If we view $\mathbf{y}$ and $\mathbf{S}$ over $\mathbb{Z}_q$ rather than $\mathbb{Z}$, then they do belong to a finite set; but for security, the masking should preserve smallness relative to q, which the uniform distribution modulo q does not achieve.

As q must be larger than $\mathbf{y}$ and the smallness of $\mathbf{S}$ relative to q impacts security, this flooding approach leads to large parameters. Instead, Lyubashevsky [Lyu09, Lyu12] put forward the notion of Fiat-Shamir with aborts. This is the reason for the test concerning $\mathbf{z}$ in the signing algorithm: it is so that the output signature $(\mathbf{z}, \mathbf{c})$ follows a distribution that is independent of the sensitive term $\mathbf{Sc}$.

A classic application of rejection sampling (see, e.g., [Dev86, Chapter 2]) is to use a source distribution Q that is convenient to sample from, to create samples from a target distribution P. In Lyubashevsky's scheme, the purpose differs: we start from a pre-source distribution Q for $\mathbf{y}$; it is shifted by $\mathbf{Sc}$, leading to a distribution $Q_{+\mathbf{Sc}}$ for $\mathbf{y}+\mathbf{Sc}$; the latter is the source distribution; it is rejected to a target distribution P for $\mathbf{z}$ that does not depend on the signing key $\mathbf{S}$. The purpose of rejection sampling here is to hide the sensitive data $\mathbf{Sc}$. Diverse choices of pairs of distributions have been put forward in the literature: uniform in hypercubes [Lyu09], Gaussian with the same standard deviation while allowing for some small statistical inaccuracy in the target distribution [Lyu12]. The efficiency of these choices is constrained by two parameters: the expected number of iterations M and the statistical distance ε between the the resulting distribution and the target one. Due to its flexibility, the rejection sampling technique allows us to adapt the Schnorr protocol to the lattice setting in multiple ways. This gives rise to the following question:

Q1. *Given signing runtime requirements, which rejection sampling strategy leads to the most compact signatures?*

Dilithium chose to answer a different question: which rejection sampling strategy leads to the "easier to implement" signature? Their approach was to choose P and Q uniform in hypercubes, i.e. each coordinate is uniform over a centered range. This leads to a rejection condition which only depends on the infinite norm of $\mathbf{z}$. Thus, obtaining randomness for the scheme can be done very efficiently as well as computing the rejection condition. An interesting observation from [DDLL13] is that the statistical inaccuracy from Gaussian to Gaussian rejection sampling stems from the tails of the distributions. In particular, while one tail from the source (shifted) Gaussian behaves nicely, the other cannot be rejected correctly to the target distribution. Then, one can consider *bimodal* Gaussian as the source distribution. Namely, instead of setting $\mathbf{z} = \mathbf{y} + \mathbf{Sc}$, one sets $\mathbf{z} = \mathbf{y} + (-1)^b\mathbf{Sc}$ for some uniform bit b. Now, the two tails of the source distribution behave nicely, which leads to a smaller expected number of rejections for the same standard deviation. Equivalently, given a target expected number of iterations, it is achieved with a smaller standard deviation than in the *unimodal* setting. However, the bit b cannot be revealed or we fall back in the unimodal setting. The key generation algorithm is then tweaked in order to produce $\mathbf{A}$ and $\mathbf{S}$ such that $\mathbf{AS} = -\mathbf{AS} \bmod p$ for some integer p. In [DDLL13], the authors aim at setting $\mathbf{AS} = q\mathbf{I} \bmod 2q$, where $\mathbf{I}$ is the identity matrix. A careful analysis however reveals that setting $\mathbf{AS} = q\mathbf{J} \bmod 2q$ for any nonzero binary rectangular $\mathbf{J}$ is enough for the security reduction to go through.

1.2 Contributions

The findings of this thesis are part of the aforedescribed paradigm. We start by analyzing and fixing the security proofs of Fiat-Shamir with Aborts (Chapter 3) before

carefully choosing a new pair of source and target distributions in order to minimize the expected norm of the signature (Chapter 4). We then propose an implementation and analyze its performances (Chapter 5) before proposing a rejection-free yet efficient, new lattice-based adaptation of Schnorr protocol (Chapter 6).

1.2.1 Chapter 3: A Detailed Analysis of Fiat-Shamir with Aborts

This chapter focuses on proving the security of Fiat-Shamir with Aborts. Indeed, previous proofs suffer from a common flaw, which we now describe.

1.2.1.1 A Flaw in Previous Proofs

An important caveat of rejection sampling is that with an aborting identification protocol, simply applying the Fiat-Shamir transform does not yield a correct signature, because of the rejection probability. However, one can adapt the transform by rerunning the signing algorithm until a non-aborting transcript is found: this is the Fiat-Shamir with aborts transform. Of course, this change of setting renders the generic security reductions of Fiat-Shamir useless in this case. This raises the question of the HVZK property needed: do we need to simulate aborting transcripts? Previous reductions [Lyu12, KLS18, AFLT16] only required to simulate non-aborting transcripts. However, those reductions only consider a specific case, where the number of aborts is bounded, for simplicity. Indeed, up to outputting $\perp$, they assume that the signing algorithm is run at most B times: we call this Fiat-Shamir with Bounded Aborts (FSwBA).

An unsubstantiated intuition. We start by describing a first flaw appearing in all existing analyses. These analyses start as follows: in the genuine security experiment (denoted Game 0), all (successful or not) transcripts generated during a sign query use a challenge that is computed with the hash function. Then, a first hybrid (Game 1) changes the sign algorithm by sampling a uniformly random challenge and programming the hash function consistently with the successful proof transcript *only*. All proofs immediately conclude these two games are identical: the (unsubstantiated) intuition is that the adversary does not have access to the aborted transcripts, and hence programming these transcripts does not impact the adversary's view.

F1. Assume the challenger in the genuine CMA (or even CMA_1) security game answers a sign query μ using a sequence of commitments $w_1, w_2, \ldots$. Assume that rejecting is a deterministic function of w and c (this is for example the case for Lyubashevsky's signatures with the parameters considers in [AFLT16]). Then, as soon as w_1 fails to produce a valid transcript, the hash value $H(w_1 \| \mu)$ is fixed and the sign oracle can no longer return a valid signature which uses commitment w_1. This is not the case in Game 1, since the hash value $H(w_1 \| \mu)$ is not programmed by the failed attempt, and the sign query could return a signature (w_1, c', z') for $c' \neq c$.

FSwBA has been analyzed and used numerous times (we focus here on the most detailed analyses), yet the above flaw **F1** appears in [Lyu12, Lemma 5.3], [Lyu16,

Lemma 4.1], [KLS18, Theorem 3.2], and [Kat21, Lemma 4.6]. Moreover it appears in [AFLT16] though not in Game 1 but in Game 0: in the proof of [AFLT16, Theorem 1], the authors directly start with the above Game 1 rather than with the correct Game 0. Finally, the difficulty with the hash function inconsistencies seems identified in [ABB+17, Appendix B.4], but the authors do not handle the case of inconsistencies between different sign queries for the same message.

The fact that the adversary can make hash queries on superpositions of all inputs in the Quantum Random Oracle Model (QROM) makes it even more difficult to argue that the adversary cannot detect random oracle programmings, which induces additional errors.

1.2.1.2 A New, Complete Analysis of Fiat-Shamir with Aborts

Our first set of results concerns FSwBA. We assume that the identification protocol satisfies a stronger notion of HVZK, where one can simulate any transcript, even aborting ones. This notion of HVZK is closer to the standard definition in the case of non-aborting identification protocols. In particular, with this notion of HVZK, we can show that if we apply the Fiat-Shamir transform on the identification protocol, then the resulting signature may not be correct, but is at least unforgeable according to standard reductions such as [GHHM21]. We can then consider the reduction which on FSwBA signature queries forwards these queries at most B times to the Fiat-Shamir signature oracle, which gives rise to a tight reduction. This is in essence what we do, except that we need to do it from scratch as the details of the reduction will help to prove the expected polynomial runtime of the signing algorithm in the case of Fiat-Shamir with Unbounded Aborts (FSwUA).

Our second set of results concerns FSwUA, i.e. the flavor of Fiat-Shamir with Aborts without an artificial bound on the number of iterations. On the negative side, we exhibit an interactive proof system such that applying FSwUA to it leads to a signature scheme such that:

- for all signing keys, with non-zero probability over the random oracle randomness, signing loops forever for all messages; in particular, the expected signing runtime is infinite;
- with overwhelming probability over the random oracle randomness, for all messages and all signing keys, the expected runtime of signing over its own randomness is below a fixed polynomial.

This suggests a modification of the signing efficiency requirement, in which the runtime expectation is not taken over the randomness of the random oracle, but should be bounded by a polynomial with overwhelming probability over the randomness of the random oracle. On the positive side, we give analyses of correctness, signing efficiency (with respect to the modified definition) and security for FSwUA in the ROM.

The main applications of these results are the Lyubashevsky and BLISS identification protocols. We show that they satisfy the stronger notion of HVZK we consider. Simulating a non-aborting transcript is done as usual: first sample $\mathbf{z}$ according to the target distribution, and then compute $\mathbf{w} = \mathbf{Az} - \mathbf{Tc} \bmod p$, as $\mathbf{c}$ is given as an input of the simulator. In the case of aborting transcripts, we have to

compute $\mathbf{w} = \mathbf{A}\mathbf{y} \bmod p$ with $\mathbf{y}$ sampled from the source distribution conditioned on $\mathbf{z}$ being rejected. We show that in usual cases, the entropy of $\mathbf{y}$ is sufficiently large, even with the conditioning, and the leftover hash lemma allows us to replace $\mathbf{w}$ with some uniform vector modulo p.

Throughout the chapter, we give an adaptation of the setting for another flavor of HVZK. We note that the simulator we exposed above can be decomposed in two parts: one part for non-aborting transcripts, and one part for aborting ones. We consider a specific case, where the simulator for non-aborting transcripts is close to real non-aborting transcripts in the sense of the *Rényi divergence* (aborted transcripts are still measured using the statistical distance). Given two distributions P and Q, the Rényi divergence of infinite order is the maximal ratio of their probability mass function. We recall its properties in Section 2.2.2.1. Sections pertaining to this specific case have a name starting with "Rényi Divergence Approach".

To understand its usefulness, we go back to the rejection sampling technique. We notice that for a given pair of source and target distribution, the proof that shows that the statistical distance between the rejection sampling output and the target distribution is $\leq \varepsilon$ also shows that their Rényi divergence of infinite order is (roughly) $\leq 1 + \varepsilon$. Hence, both approaches can be applied to the Lyubashevsky identification protocol[3]. It turns out that by comparing them, we notice that the Rényi Divergence allows for larger ε, i.e. for a broader class of pairs of source and target distributions. Namely, the Rényi divergence approach gives useful reductions up to $\varepsilon = O(1/Q_s)$, the number of signature queries, while the statistical distance approach is only meaningful up to $\varepsilon = \mathsf{negl}(\lambda)$.

While our results can be extended to the QROM, this chapter focuses on the ROM and classical adversaries, as the quantum part of this joint work was handled by a coauthor.

1.2.2 Chapter 4: Optimizing Lattice-based Identification Protocols

Equipped with the previous result, which gives us a class of admissible pairs of source and target distributions, we aim in this chapter at answering the question **Q1**. This chapter addresses it with a particular goal in mind: minimizing the expected norm of $\mathbf{z}$. Indeed, as $\mathbf{z}$ makes up for (almost all of) the signature, minimizing its norm minimizes the signature size. Moreover, this also minimizes the verification bound, which in turns makes the signature harder to forge. To further formalize the question, we need to address how we model the expected signing time. We make the following assumptions in this chapter:

- We have a bound $M \geq 1$ on the expected number of iterations under which we want to minimize the above.
- The runtime of one iteration is independent from the number of previous iterations and the choice of source and target distribution. Namely, the runtime is a linear function of M.
- The set of shifts $\{\mathbf{Sc}\}$ is modelled as a set of the form $\{\mathbf{v} | \|\mathbf{v}\| \leq t\}$, i.e. an hyperball.

[3] We could also apply it for BLISS, but it turns out that $\varepsilon = 0$ in this case.

On Alternative Rejection Sampling Strategies. In Section 4.2, we first investigate rejection sampling itself. Indeed, the classical rejection sampling algorithm has an expected runtime of $R_\infty(P\|Q)$ when rejecting from distribution Q to distribution P. If there is a way to make it faster, then we could look for even smaller distributions under the same time constraints. We consider the following setting. The goal is to sample from a distribution P given access to a sampler from a distribution Q, and we consider a sequence of samples $(X_i)_{i\geq 1}$ from distribution Q. Any strategy is allowed as long as we output one of the X_i's. A strategy is given by a sequence of algorithms $(A_i)_{i\geq 1}$ that take samples $(X_j)_{j\leq i}$ as input and return either an index $j \in [i]$, which corresponds to halting with output X_j, or a special symbol r which corresponds to rejecting and moving to A_{i+1}. We restrict ourselves to the case of procedures that terminate with probability 1. Considering i^* the random variable denoting the number of samples observed in a strategy, our objective is then to measure how small $\mathbb{E}(i^*)$ can be. We prove that for any P, Q, we have $\mathbb{E}(i^*) \geq R_\infty(P\|Q)$. This result is obtained by proving that for any x, we have $P(x) \leq \mathbb{E}(i^*) \cdot Q(x)$, leading to the former inequality by definition of R_∞.

Lower Bounds. In Section 4.3, we prove lower bounds on achievable compactness in the case of exact rejection sampling in both unimodal and bimodal settings. These lower bounds are obtained following a similar path. In what follows, we focus on the unimodal setting.

Our lower bounds are obtained in three steps: (1) considering the same setting with continuous distributions, we first prove that we can restrict ourselves to the case of isotropic distributions over $\mathbb{R}^m$, where isotropic means that their densities only depend on the norm. (2) Starting with f and g isotropic, we show that $\mathbb{E}_{\mathbf{x}\hookleftarrow f}(\|\mathbf{x}\|) = \mu_m/\mu_{m-1}$ where $\mu_k = \int_0^\infty r^k f(r)\,\mathrm{d}r$. Our lower bound is then obtained by applying the Cauchy-Schwarz inequality $|\mathbb{E}(XY)|^2 \leq \mathbb{E}(X^2)\mathbb{E}(Y^2)$ to random variables $X = \|\mathbf{x}\|^{m/2}$ and $Y = \|\mathbf{x}\|^{(m-2)/2}$, where $\mathbf{x} \hookleftarrow f$. Indeed, it immediately leads to inequality $\mu_m \cdot \mu_{m-2} \geq (\mu_{m-1})^2$, which results in $\mu_m/\mu_{m-1} \geq \mu_{m-1}/\mu_{m-2} \geq (t/M^{1/(m-1)} - 1)$. (3) A similar lower bound in the discrete setting is obtained by considering the continuous density $p(\mathbf{x}) = P(\lceil \mathbf{x} \rfloor)$ with P being a discrete probability. These lower bounds provide us with a target to reach, and we can compare them with the signature size obtained when instantiating the above scheme with various distributions.

- Considering Lyubashevsky's scheme with perfect rejection sampling to the target distribution P (as in [Lyu09]), the relevant quantity measuring the signing runtime is then given by $M = \max_{\mathbf{S},\mathbf{c}} R_\infty(P\|Q_{+\mathbf{Sc}})$. We show (under a mild assumption discussed below) that for all P and Q such that M is finite, the expected norm $\mathbb{E}_{\mathbf{x}\hookleftarrow P}(\|\mathbf{x}\|)$ is $\Omega((m/\log M) \cdot \max_{\mathbf{S},\mathbf{c}} \|\mathbf{Sc}\|)$.

- In the case of perfect rejection with the accommodating arithmetic modification from [DDLL13], then the relevant quantity for measuring the signing runtime is $M = \max_{\mathbf{S},\mathbf{c}} R_\infty(P\|Q_{\pm\mathbf{Sc}})$, where $Q_{\pm\mathbf{Sc}}$ denotes the balanced mixture of $Q_{+\mathbf{Sc}}$ and $Q_{-\mathbf{Sc}}$. In this case, we show (under the same mild assumption) that for all P and Q such that M is finite, the expected norm $\mathbb{E}_{\mathbf{x}\hookleftarrow P}(\|\mathbf{x}\|)$ is $\Omega(\sqrt{m/\log M} \cdot \max_{\mathbf{S},\mathbf{c}} \|\mathbf{Sc}\|)$.

Usual Choices. We recall in Section 4.4.3 results on two usual choices for source and target distributions. Namely, the uniform distribution in hypercubes and the

discrete Gaussian distribution. Gaussian distributions provide better signature compactness in the bimodal and imperfect unimodal regimes, than uniforms in hypercubes in the perfect unimodal regime. In the perfect unimodal setting, our lower bound is a factor $\sqrt{m}$ lower than the hypercube result, while Gaussian distributions are incompatible with this setting. In the bimodal setting, our lower bound is actually reached (up to a constant factor) by discrete Gaussian distributions as in [DDLL13].

However, uniforms in hypercubes are sometimes preferred (see, e.g., Dilithium), because they lead to a simpler implementation, which in turn makes protection against timing attacks easier.

Hyperball Uniforms. We show that (continuous) uniform distributions over hyperballs reach the signature compactness lower bound (up to constant factors) in both unimodal and bimodal settings, as shown in Section 4.4.1. We also show that they are experimentally as good as Gaussians for imperfect rejection sampling. These results reduce to Rényi divergence computations, which involve geometric properties of hyperballs. We emphasize that while Gaussian distributions also achieve similar signature size in both unimodal and bimodal settings (but only in the case of imperfect rejection sampling with polynomial loss for the unimodal case), using uniform distributions over hyperballs makes the rejection test as simple as computing $\|\mathbf{z}\|$ in the unimodal case since it consists only in checking that $\mathbf{z}$ is in the hyperball of the target distribution P. In the bimodal case, the rejection test involves computing two norms and flipping a coin. In order to use this distribution in a signature, we propose a generalization of Lyubashevsky's signature that allows for continuous source and target distributions, by adding a rounding step after accepting a sample. Its security relies on the same mechanisms as the discrete case. This strategy could also benefit to Gaussian distributions, by allowing to replace discrete Gaussian sampling with possibly simpler continuous Gaussian sampling. To assess the practicality of this new choice of distributions, we propose parameters for a variant of Dilithium with uniform distributions in hyperballs. If considering the sum of bitsizes of a verification key and a signature, the gains range from $\sim 25\%$ to $\sim 30\%$, depending on the security level, just like for the Gaussian variant, whose parameters we also update.

The results concerning signature compactness for unbounded (perfect and imperfect) rejection sampling are summarized in Table 1.1. For practical estimations, we give in Table 1.2 the estimated sizes for the signatures based on unimodal Gaussian rejection sampling and unimodal hyperball rejection sampling, assuming that they integrate various standard optimizations, such as relying on polynomial modules instead of $\mathbb{Z}^n$, and considering compression of the signature via truncation of its lower bits.

1.2.3 Chapter 5: HAETAE, A New Implementation of Lattice-based Fiat-Shamir with Aborts

As a natural follow-up to the two previous chapters, we propose HAETAE[4], an implementation of the hyperball-uniform bimodal variant of Lyubashevsky's sig-

[4]The haetae is a mythical Korean lion-like creature with the innate ability to distinguish right from wrong.

	Unimodal ($\varepsilon = 0$)	Unimodal ($\varepsilon \geq 2^{-o(m)}$ and $\varepsilon = o(1/m)$)	Bimodal ($\varepsilon = 0$)
Hypercube	$\frac{tm^{3/2}}{\log M}$	$\frac{tm^{3/2}}{\log M}$	$\frac{tm^{3/2}}{\log M}$
Gaussian	∞	$\frac{t\sqrt{m}\sqrt{\log\frac{1}{\varepsilon}+\log M}}{\log M}$	$\frac{t\sqrt{m}}{\sqrt{\log M}}$
Hyperball	$\frac{tm}{\log M}$ (Lemma 4.10)	$\frac{t\sqrt{m}\sqrt{\log\frac{1}{\varepsilon}+\log M}}{\log M}$ (Lemma 4.10)	$\frac{t\sqrt{m}}{\sqrt{\log M}}$ (Lemma 4.11)
Lower bound	$\frac{tm}{\log M}$ (Corollary 4.6)	?	$\frac{t\sqrt{m}}{\sqrt{\log M}}$ (Corollary 4.9)

Table 1.1: Optimal asymptotic expected Euclidean norm of $\mathbf{z}$ for different pairs of source and target distributions. Parameter M quantifies the expected number of iterations, ε quantifies the accuracy of the rejection sampling, m is the dimension of $\mathbf{z}$ and t is an upper bound on $\|\mathbf{Sc}\|$. Multiplicative constants are omitted and we assume that $\log M \leq m$. The last row corresponds to the lower bounds we computed.

nature. This implementation features various optimizations, already present in Dilithium [BDK+20] and which we adapt to the bimodal setting.

The HAETAE team is comprised of ten members and informations about the project can be found on the website[5]. It is a submission to the Korean post-quantum competition[6] and to the fourth round of the NIST post-quantum signature competition[7].

It implements the *bimodal hyperball* setting from Chapter 4. We adapted the following optimizations, which were already considered in multiple schemes, e.g., Dilithium, Kyber, Saber, to the bimodal setting for our implementation. We also adapted continuous hyperball sampling to discrete hyperball sampling in order to enable fixed-point arithmetic in the signature algorithm.

Polynomial modules. Instead of working over $\mathbb{Z}^m$, we rely on polynomial modules over a power-of-two cyclotomic ring, thus speeding up computations and decreasing the sizes.

Modulus and truncating the verification key. First, as we noted in the previous chapter, the bimodal hyperball setting leads to the smallest for the Euclidean norm signatures. For the same SIS security than Dilithium, we are thus able to aggressively decrease the modulus, from 23 to 16 bits. As the verification key bit-size is proportional to the logarithm of the modulus, this helps reducing it. Moreover, we adapt Dilithium's verification key truncation to our setting. However, as we already enjoy the modulus reduction, our adaptation turns out to be less efficient than Dilithium's as we only cut at most 1 bit per coordinate.

Compression techniques to lower the signature size. We use two techniques to compress the signatures. First, as the verification key $\mathbf{A}$ is in (almost)-HNF, we rely on the Bai-Galbraith technique [BG14]. Namely, the second part of the signature, which is multiplied by $2\mathbf{Id}$ in the challenge computation and verification algorithm, can be aggressively compressed by cutting its low bits. This requires in turn modifying the computation of the challenge c and the verification algorithm, in order

[5] `https://kpqc.cryptolab.co.kr/haetae`

[6] `https://www.kpqc.or.kr/`

[7] `https://csrc.nist.gov/projects/pqc-dig-sig/`

to account for this precision loss. Usually, this is done by keeping only the high bits of $\mathbf{Ay}$ in the computation of the challenge. However, as we multiply everything by 2, we do not keep the lowest bit of those high bits and keep the (overall) least significant bit instead. As in Dilithium, our decomposition of bits technique is a Euclidean division with a centered remainder, and we choose a representative range for modular integers that starts slightly below zero to further reduce the support of the high bits. The second compression technique, suggested in [ETWY22] in the context of lattice-based hash-and-sign signatures, concerns the choice of the binary representation of the signature. As the largest part of it consists in a vector that is far from being uniform, we can choose some entropic coding to obtain a signature size close to its entropy. In particular, as in [ETWY22], we choose the efficient range Asymmetric Numeral System to encode our signature, as it allows us to encode the whole signature and not lose a fraction of a bit per vector coordinate, like with Huffman coding. We can further apply the two techniques to the hint vector $\mathbf{h}$, which is also a part of the signature, to reduce the signature sizes.

Fixed-point algorithm for hyperball sampling. Unlike uniform Gaussian sampling or uniform hypercube sampling, uniform hyperball sampling has not been considered in the cryptographic protocols before the suggestion of [DFPS22]. To narrow the gap between the hyperball uniforms sampled in the real and the ideal world, we discretize the hyperball and bound the numerical error and their effect by analyzing their propagation. This leads to a fixed-point hyperball sampling algorithm and, therefore, the fixed-point implementation of the whole signing process.

Let us briefly compare the resulting scheme with Dilithium and Falcon.

Comparison with Dilithium. Once the design rationale was set, I was mainly involved with the parameter computations. We chose the *core-SVP* methodology, a conservative approach to security estimation. Our signature size is 25% to 40% smaller than Dilithium's while the verification key is 20% to 25% smaller as seen in Table 1.2. This comes at the price of having a less efficient signature algorithm. However, we note that the hyperball sampling takes up to 80% of the total runtime, and thus any improvement on this sampler would lead to massive gains in the signing time. Finally, we note that our design is based on Dilithium-G, which appears in a preliminary version of [DKL$^+$18], and its design has a few subtle differences with Dilithium when it comes to the role of the hint and in the definition of the commitment, where both are optimized for hypercubes in Dilithium.

Comparison with Hash and Sign lattice signatures. In terms of ease of implementation, our scheme favorably compares to lattice signatures based on the hash and sign paradigm such as Falcon [FHK$^+$17] and Mitaka [EFG$^+$22]. HAETAE, Falcon and Mitaka all three rely on some form of Gaussian sampling, which are typically difficult to implement and protect against side-channel attacks. Falcon makes sequential calls to a Gaussian sampler over $\mathbb{Z}$ with arbitrary centers. Mitaka also relies on an integer Gaussian sampler with arbitrary centers, but the calls to it can be massively parallelized. It also uses a continuous Gaussian sampler, which is arguably simpler. HAETAE, however, only relies on a (zero-centered) continuous Gaussian sampler, used to sample uniformly in hyperballs. The calls to it can also be massively parallelized. This difference makes HAETAE possible to have a fixed-point signing algorithm and easier maskings. Further, in the randomized version of the signature scheme, these samples can be computed off-line as they are independent from the message to be

signed. The on-line tasks are far simpler than those of Falcon and Mitaka. Finally, we note that key-generation is much simpler for HAETAE than in Falcon and Mitaka.

While HAETAE is simpler from an implementation perspective, its verification key and signature sizes are larger than Falcon's and Mitaka's.

1.2.4 Chapter 6: Towards Efficient Lattice-based Fiat-Shamir

In this last chapter, we introduce a new paradigm for adapting Schnorr's identification protocol to the lattice setting. It relies on Gaussian convolution, rather than flooding or rejection sampling. Our $\mathsf{G}+\mathsf{G}$ (Gaussian Plus Gaussian) identification protocol can be compiled into a signature using the Fiat-Shamir heuristic (without aborts), in the QROM.

$\mathsf{G}+\mathsf{G}$ involves two Gaussian samples that are being summed. The first one is $\mathbf{y}$ and the second one corresponds to $\mathbf{Sc}$. The first difficulty that we face is that $\mathbf{S}$ is fixed and $\mathbf{c}$ is publicly known as part of the resulting signature and hence cannot be assumed random for the sake of studying the distribution of $\mathbf{z}$.

To introduce the required new randomness, we start from BLISS [DDLL13]. Recall that the verification key $\mathbf{A} \in \mathbb{Z}_{2q}^{m\times k}$ and the signing key $\mathbf{S} \in \mathbb{Z}^{k\times m}$ satisfy the relation $\mathbf{AS} = q\mathbf{I}_m \bmod 2q$. Among the variants of Lyubashevsky's signature, it is a specificity of BLISS to work modulo $2q$, which is particularly useful in our case. The commitment of the prover is $\mathbf{w} = \mathbf{Ay} \bmod 2q$, and upon receiving $\mathbf{c} \in \{0,1\}^m$, the prover replies with either $\mathbf{z} = \mathbf{y} + \mathbf{Sc}$ or $\mathbf{z} = \mathbf{y} - \mathbf{Sc}$ with probability $1/2$ each. The verifier checks that $\mathbf{z}$ is short and $\mathbf{Az} = \mathbf{w} + q\mathbf{c} \bmod 2q$. This check works for both values of $\mathbf{z}$ that the prover chose from. This can be explained by observing that the verification views $\mathbf{c}$ modulo 2, i.e., as a coset of $\mathbb{Z}^m/2\mathbb{Z}^m$, and negating it does not change the coset. This observation was used in [Duc14] to take negations of individual coordinates of $\mathbf{c}$ to minimize the Euclidean norm of $\mathbf{Sc}$ and hence decrease the standard deviation of $\mathbf{y}$ necessary to hide $\mathbf{Sc}$ via rejection sampling. We go further and let the prover extend the coset $\mathbf{c}$ sent by the verifier to a Gaussian sample with support $2\mathbb{Z}^m + \mathbf{c}$. The verification equation above still holds, and we now have our second Gaussian.

At this stage, the prover samples a Gaussian $\mathbf{y}$ over $\mathbb{Z}^k$, transmits $\mathbf{w} = \mathbf{Ay} \bmod 2q$, receives a uniform coset $\mathbf{c} \in \mathbb{Z}^m/2\mathbb{Z}^m$ from the verifier, produces a Gaussian sample $\mathbf{x}$ with support $2\mathbb{Z}^m + \mathbf{c}$ and computes $\mathbf{z} = \mathbf{y} + \mathbf{Sx}$. Equivalently, it samples $\mathbf{k}$ Gaussian with support $2\mathbf{S}\mathbb{Z}^m + \mathbf{Sc}$ and returns $\mathbf{z} = \mathbf{y} + \mathbf{k}$. In order to obtain the zero-knowledge property (i.e., be able to simulate signatures without knowing the signing key), we aim to prove that the distribution of the Gaussian convolution $\mathbf{z}$ can be sampled from publicly. If $\mathbf{y}$ and $\mathbf{k}$ were continuous Gaussians, we would set their covariance matrices $\mathbf{\Sigma_y}$ and $\mathbf{\Sigma_k}$ such that $\mathbf{\Sigma_y} + \mathbf{\Sigma_k} = \mathbf{\Sigma_z}$ for a known covariance matrix $\mathbf{\Sigma_z}$ for $\mathbf{z}$. To fix the ideas, we could set $\mathbf{\Sigma_z} = \sigma^2\mathbf{I}$ for some $\sigma > 0$, i.e., the distribution of $\mathbf{z}$ is a spherical Gaussian, and set $\mathbf{\Sigma_y} = \sigma^2\mathbf{I} - \mathbf{\Sigma_k}$. If we sample $\mathbf{x}$ from a spherical Gaussian with standard deviation $s > 0$, then $\mathbf{\Sigma_k} = s^2\mathbf{S}\mathbf{S}^\top$ and $\mathbf{\Sigma_y} = \sigma^2\mathbf{I} - s^2\mathbf{S}\mathbf{S}^\top$ (by taking σ sufficiently large, the latter is indeed definite positive). This is the choice we actually make for $\mathsf{G}+\mathsf{G}$, but there is flexibility.

The above over-simplifies the situation as the Gaussians we manipulate are discrete rather than continuous. Further, their supports do not have the same dimensions. Indeed, the support of $\mathbf{y}$ is $\mathbb{Z}^k$ whereas the support of $\mathbf{k}$ is exactly $2\mathbf{S}\mathbb{Z}^m + \mathbf{Sc}$ whose span has dimension $m < k$: the second Gaussian lives in a smaller dimension

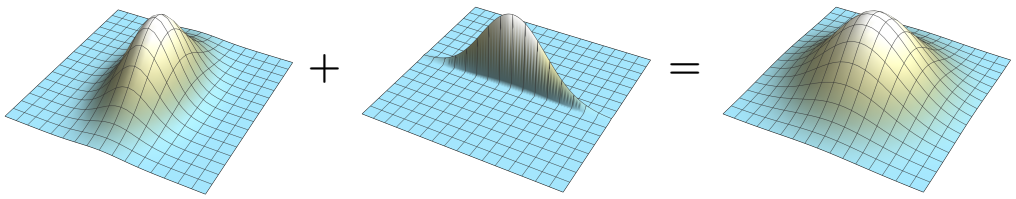

Figure 1.1: The sum of two Gaussians with compensating covariance matrices is a spherical Gaussian, even when the second Gaussian is rank-deficient. In the $\mathsf{G}+\mathsf{G}$ identification protocol and signature, the first Gaussian corresponds to $\mathbf{y}$, the second Gaussian is associated to $\mathbf{Sc}$ and the resulting one corresponds to $\mathbf{z}$.

and its support is sparser. This is illustrated in Figure 1.1. Convolution of discrete Gaussians has been studied in [BF11], but restricted to the case of full-dimensional supports. As a technical contribution, we extend their result to rank-deficient Gaussians with co-diagonalizable covariance matrices.

Thanks to the above, if the covariance matrices are set appropriately, then $\mathsf{G}+\mathsf{G}$ is honest-verifier zero-knowledge (HVZK). The proofs of completeness and soundness are adapted from [DDLL13]. To apply the Fiat-Shamir heuristic, we also need the commitment $\mathbf{Ay}$ to have sufficiently high min-entropy. This is technically more complex than for Lyubashevsky's signatures as $\mathbf{y}$ is distributed from a skewed Gaussian. The properties satisfied by $\mathsf{G}+\mathsf{G}$ allow a conversion to a secure signature scheme, using completeness and commitment-recoverability to obtain the correctness of the signature, HVZK and commitment-min-entropy to reduce security against chosen-message attacks to security against no-message attacks, and computational soundness (resp. lossy-soundness) which implies security against no-message attacks for different parametrizations.

Comparison with BLISS. Among variants of Lyubashevsky's signatures, BLISS provides the smallest $\mathbf{z}$: its expected norm can be as small as $\sigma_1(\mathbf{S})m/\sqrt{\log M}$ (up to a constant factor), where $\sigma_1(\mathbf{S})$ is the largest singular value of $\mathbf{S}$ and M is the expected number of repetitions (see Lemma 4.16). Further, following Corollary 4.9, this is essentially optimal for Lyubashevsky's signatures, even if we allow to optimize over the choice of source and target distributions. In the case of $\mathsf{G}+\mathsf{G}$, the strongest constraint on parameters is essentially that the standard deviation σ of $\mathbf{z}$ be sufficiently large to "smooth out" the lattice $2\mathbf{S}\mathbb{Z}^m$. By using the variant of the HVZK property based on the Rényi divergence rather than the statistical distance defined in Definition 3.3, where we further set $p = 0$ as no rejection is involved, it suffices that σ be above $\sigma_1(\mathbf{S})\sqrt{\log Q_S}$, up to a constant factor, where Q_S is the maximum number of signature queries that the adversary is allowed to make. As a result, the expected norm of $\mathbf{z}$ in $\mathsf{G}+\mathsf{G}$ is $\sigma_1(\mathbf{S})\sqrt{m \log Q_S}$. We conclude by observing that $\log Q_S$ is typically much smaller than m, and that the $\sqrt{\log M}$ term from BLISS cannot grow sufficiently to compensate for the difference. More concretely, if we set $M = \lambda^{\Theta(1)}$, $Q_S = \lambda^{\Theta(1)}$ and $m = \Theta(\lambda)$, where λ is the security parameter, then the expected norms of $\mathbf{z}$ in BLISS and $\mathsf{G}+\mathsf{G}$ respectively grow as $\sigma_1(\mathbf{S}) \cdot \lambda/\sqrt{\log \lambda}$ and $\sigma_1(\mathbf{S}) \cdot \sqrt{\lambda \log \lambda}$.

Optimization and concrete parameters. While all key generation techniques pre-

Bit Security	120		160		260	
NIST Level	II		III		V	
Size	sig.	vk	sig.	vk	sig.	vk
Dilithium	2420	1312	3293	1952	4595	2592
Chapter 4 (Unimodal Gaussian)	1921	800	2462	1056	3553	1760
Chapter 4 (Unimodal Hyperball)	1903	800	2473	1056	3461	1760
Chapter 5 (Bimodal Hyperball)	1463	992	2337	1472	2908	2080
Chapter 6 (Rejection-free)	1542	1120	2033	1568	2518	2336

Table 1.2: Signature sizes in bytes from across this thesis and Dilithium for comparison. All results can be reproduced using the scripts found in their respective folder at `https://github.com/jdevevey/thesis` and are all modifications of `https://github.com/pq-crystals/security-estimates`.

sented in [DDLL13] can be used with our $\mathsf{G}+\mathsf{G}$ protocol, we present alternative versions which offer more flexibility. A first improvement is that we can set $\mathbf{AS} = q\mathbf{J} \bmod 2q$, where $\mathbf{J} \in \mathbb{Z}_q^{m\times\ell}$ is only rectangular and full column-rank rather than set to the identity. Departing from the BLISS setting, when instantiating $\mathsf{G}+\mathsf{G}$ with the MLWE and MSIS hardness assumptions [BGV12,LS15] over a ring $\mathcal{R} = \mathbb{Z}[x]/(x^n+1)$ with n a power of 2, we take $\mathbf{j} = (x^{n/2}+1, 0, \ldots, 0)$. This allows us to replace the lattice $2\mathbf{s}\mathcal{R}$ with $(x^{n/2}-1)\mathbf{s}\mathcal{R}$, and to decrease the standard deviation of $\mathbf{z}$ by a factor $\sqrt{2}$. Overall, we obtain signature sizes that are between 20% and 30% smaller than those from Chapter 4, or 35% to 45% smaller than Dilithium [DKL$^+$18]. We recall that Table 1.2 summarizes all the different sizes for the different signatures presented in this thesis.

1.3 Other Contributions and Publications

The various results from this thesis and complimentary findings resulted in the following publications.

[DFPS23] A detailed Analysis of Fiat-Shamir with Aborts.
J. Devevey, P. Fallahpour, A. Passelègue, D. Stehlé. *Crypto 2023.*
This work contains additional results for Chapter 3, which includes QROM proofs as well as a fix for the [KLS18] proof.

[DFPS22] On Rejection Sampling in Lyubashevsky's Signature Scheme.
J. Devevey, O. Fawzi, A. Passelègue, D. Stehlé. *Asiacrypt 2022.*
This work contains additional results for Chapter 4. It includes more details as well as a section on bounded rejection sampling strategies, which are a hybrid between rejection-free signatures and rejection sampling.

[CCD$^+$23] HAETAE: Shorter Lattice-based Fiat-Shamir Signatures.
J. H. Cheon, H. Choe, J. Devevey, T. Güneysu, D. Hong, M. Krausz, G. Land, M. Möller, D. Stehlé, and M. Yi. *Preprint.*
This work contains additional results for Chapter 5. I played two main roles in the project. First, as I brought the bimodal hyperball idea, I worked on adapting the optimizations we will discuss next to the bimodal setting. Second,

I was in charge of computing the parameter sets. The tools I used can be found on the project website or on the dedicated git repository for this thesis[8]. The full version includes the whole documentation of HAETAE, including parts where I was less involved, such as practical security assessment, longer report on performances and implementation techniques. It also includes reminders which were already present in Chapter 3 and Chapter 4.

In preparation G + G: A Fiat-Shamir Lattice Signature Based on Convolved Gaussians.
J. Devevey, A. Passelègue, D. Stehlé. *Submitted.*
The last chapter of this thesis, Chapter 6, is currently in review process for publication. The version presented in this thesis is already the full version and no content was omitted.

Additionally, three more publications resulted from these PhD years. They are omitted from this manuscript for consistency as they approach different topics.

[DSSS21] On the Integer Polynomial Learning with Errors Problem.
J. Devevey, A. Sakzad, D. Stehlé, R. Steinfeld. *PKC 2021.*
Consider a polynomial ring $\mathbb{Z}[x]/f(x)$ and a prime q. We compare the hardness of the LWE problem over this ring with the hardness of the problem where each polynomial is evaluated at q and reduced modulo $f(q)$ before doing any operation. This allowed to prove the hardness of a variant of ThreeBears, a round 2 key-exchange mechanism submission to the NIST post-quantum competition.

[DLN+21] Non-interactive CCA2-secure Threshold Cryptosystems: Achieving Adaptive Security in the Standard Model without Pairings.
J. Devevey, B. Libert, K. Nguyen, T. Peters, M. Yung. *PKC 2021.*
This work proposes two constructions of threshold public key encryption, where the decryption key is split among multiple parties. One is based on the DCR assumption and the second one on lattice assumptions. The adaptive IND-CCA2 security model allows an adversary to adaptively corrupt parties as well as query partial decryption for non-corrupted ones. Before this work, this was not achieved in the standard model except for pairing-based constructions.

[DLP22] Rational Modular Encoding in the DCR setting: Non-interactive Range Proofs and Paillier-based Naor-Yung in the Standard Model.
J. Devevey, B. Libert, T. Peters. *PKC 2022.*
This work proposes a new construction of range proofs in the standard model under the DCR assumption. As an application, we recover the security of a Naor-Yung construction of threshold encryption under the DCR setting. The previous proof contained a flaw which was fixed using our new construction of range proofs.

[8] `https://github.com/jdevevey/thesis`

Chapter 2

Preliminaries

We start by introducing all the notations we use and we recall the previous results we rely on and give some extensions.

2.1 Notations

Matrices are denoted in bold font and upper case letters (e.g., $\mathbf{A}$), while vectors are denoted in bold font and lowercase letters (e.g., $\mathbf{y}$ or $\mathbf{z}_1$). The i-th component of a vector is denoted with subscript i (e.g., y_i for the i-th component of $\mathbf{y}$).

Vectors are column vectors. We let $(\mathbf{u}, \mathbf{v})$ denote concatenation between matrices and/or vectors by putting the rows below and $(\mathbf{u}|\mathbf{v})$ the columns on the right.

When we consider a probability density, this is with respect to the canonical (i.e., Lebesgue or counting) measure μ over their support. We may identify the notion of probability distribution and probability density in the discrete case. Given two probability distributions F and G with densities f and g, we let $x \hookleftarrow F$ denote the sampling of x according to F. We let $\mathrm{Supp}(F)$ denote the smallest (for inclusion) set such that for any set X, we have $F(X \cap \mathrm{Supp}(F)) = F(X)$. In particular, in the case where F is discrete, then we have $\mathrm{Supp}(F) = \{x | F(x) \neq 0\}$. Given a set $S \subseteq \mathrm{Supp}(F)$, we let F^S denote the distribution F cut to S, i.e., the measure $F/F(S)$ restricted to S. We let $F \otimes G$ denote the distribution of (x, y) where $x \hookleftarrow F$ and $y \hookleftarrow G$ are independent and $f \otimes g$ one of its density. Moreover given an element $\mathbf{x}$, we let $F_{+\mathbf{x}}$ (respectively $F_{\pm\mathbf{x}}$) denote the distribution with density $f_{+\mathbf{x}} : \mathbf{y} \mapsto f(\mathbf{y}-\mathbf{x})$ (respectively $f_{\pm\mathbf{x}} : \mathbf{y} \mapsto (f(\mathbf{y}-\mathbf{x})+f(\mathbf{y}+\mathbf{x}))/2$). Given a finite or measurable set S, we let $U(S)$ denote the uniform distribution over S. We let $\mathcal{N}(\mu, \sigma)$ denote the normal distribution centered at μ with standard deviation σ. By notation abuse, we use algorithm names to denote the random variable associated to their output.

Given a dimension $m \geq 1$, a center $\mathbf{c} \in \mathbb{R}^m$ and a radius $r > 0$, we let $\mathcal{B}_m^p(r, \mathbf{c})$ (resp. $\mathcal{S}_m^p(r, \mathbf{c})$) denote the p-norm ball (resp. sphere) of radius r and center $\mathbf{c}$ for any $p \in [1, +\infty]$. When $p = 2$ (resp. $\mathbf{c} = 0$), we omit it. Let $V_m(r) := \frac{\pi^{m/2}}{\Gamma(m/2+1)} r^m$ denote its volume as well as $S_m = m \cdot V_m(1)$ denote the surface of the unit sphere.

Given a set and a subset $S \subseteq Y$ we let $\chi_S : x \mapsto \{1 \text{ if } x \in S,\ 0 \text{ if } x \in Y \setminus S\}$ denote the indicator function of S. Let $\lceil \cdot \rfloor : \mathbb{R} \to \mathbb{Z}$ be the rounding operator that maps x to the nearest integer (in case of a tie, it is rounded downwards). It is

naturally extended to $\mathbb{R}^n$ by coordinate-wise application. The notation log refers to the natural logarithm, except in Chapter 3.

For some λ going to infinity, we let $\mathsf{negl}(\lambda) = \lambda^{-\omega(1)}$ as well as $\mathsf{poly}(\lambda) = \lambda^{O(1)}$. Let $H : \mathcal{D} \to \mathcal{R}$ be a function with finite $\mathcal{R}$. It is modeled as a random oracle by replacing it by a uniformly sampled function among those from $\mathcal{D}$ to $\mathcal{R}$. We call this the random oracle model (ROM). To denote that it is reprogrammed at input x to the value y we use the notation $H^{x \mapsto y}$.

For a positive integer α, we let $r \bmod^{\pm} \alpha$ denote the unique integer r' satisfying the relation $r = r' \bmod \alpha$ in the range $[-\alpha/2, \alpha/2)$. We also define $r \bmod^{+} \alpha$ as the unique integer r' in the range $[0, \alpha)$ satisfying $r = r' \bmod \alpha$. We let $\mathsf{LSB}(r)$ denote the least significant bit of an integer r. We naturally extend this to integer polynomials and vectors, by applying it component-wise.

2.2 Probabilities

This section introduces commonly-used probability tools.

2.2.1 Min-Entropy and Statistical Distance

We recall the definition of min-entropy and conditional min-entropy.

Definition 2.1 (Min-Entropy)**.** *Let $X = (Y, Z)$ be a random variable. Let p_X, p_Z be the densities of X and Z, and $p_{Y|Z=z}$ the density of Y conditioned on $Z = z$. The min-entropy of Y is $H_\infty(Y) = \max_{y \in \mathrm{Supp}(p_Y)} p_Y(y)$. The conditional min-entropy of Y on Z is:*

$$H_\infty(Y|Z)_{p_X} = -\log \left(\int_{\mathrm{Supp}(p_Z)} p_Z(z) \max_{y \in \mathrm{Supp}(p_{Y|Z=z})} p_{Y|Z=z}(y) \, \mathrm{d}\mu(z) \right).$$

To quantify similarities between distributions, we consider the statistical distance.

Definition 2.2 (Statistical Distance)**.** *Let P, Q be two probability distributions with respective densities p, q. Their statistical distance is*

$$\Delta(P, Q) := \frac{1}{2} \int_{\mathrm{Supp}(P) \cup \mathrm{Supp}(Q)} |p(x) - q(x)| \, \mathrm{d}\mu(x).$$

We recall the Leftover Hash Lemma, which we use in Chapter 3.

Lemma 2.1 (Leftover Hash Lemma)**.** *Let $n, m > 0$ and q be a prime and define the set $\mathcal{H} = \{\mathbf{A} \in \mathbb{Z}_q^{n \times m} : \mathbf{y} \mapsto \mathbf{A}\mathbf{y}\}$. Then for any random variable X over $\mathbb{Z}_q^m$ and $\varepsilon > 0$ such that $H_\infty(X) \geq \log q^n + 2\log(1/\varepsilon)$, the distributions $(h, h(X))$ and $(h, U(\mathbb{Z}_q^n))$ are within statistical distance ε.*

2.2.2 Rényi Divergence and Extension

The Rényi Divergences form a class of divergences which play a central role in this work. Indeed, in the case of lattice-based signatures, it often gives tighter security bounds than what is obtained through the use of the statistical distance. Moreover, the Rényi divergence of infinite order shares a strong link with rejection sampling, which we explicit in Section 2.2.3.

2.2.2.1 Rényi Divergence

We start by recalling the definition of the Rényi divergence of order $a \in (1, +\infty]$.

Definition 2.3 (Rényi divergence)**.** *Let $a \in (1, +\infty)$. Let P, Q be two probability distributions with P absolutely continuous with respect to Q. Their Rényi divergence of order a, assuming that it exists, is*

$$R_a(P\|Q) := \left(\int_{\mathrm{Supp}(P)} \left(\frac{\mathrm{d}P}{\mathrm{d}Q}(x) \right)^{a-1} \mathrm{d}P(x) \right)^{\frac{1}{a-1}}.$$

Their Rényi divergence of infinite order is

$$R_\infty(P\|Q) := \operatorname*{ess\,sup}_{x \in \mathrm{Supp}(P)} \frac{\mathrm{d}P}{\mathrm{d}Q}(x).$$

By notation abuse, we may use random variables instead of probability densities as arguments, for the notions defined above.

The following lemma lists standard properties of the Rényi divergence.

Lemma 2.2 ([vEH14])**.** *Let X and Y be two random variables with probability distributions P_X and P_Y such that $\mathrm{Supp}(P_X) \subseteq \mathrm{Supp}(P_Y)$. The following holds for any order $a \in (1, +\infty]$.*

- ***Log. Positivity:*** *$R_a(P_X\|P_Y) \geq R_a(P_X\|P_X) = 1$.*
- ***Data Processing Inequality:***

  $$R_a(P_{f(X)}\|P_{f(Y)}) \leq R_a(P\|Q) \tag{2.1}$$

  for any map f, where $P_{f(Z)}$ denotes the distribution of $f(Z)$ for $Z = X$ or Y.
- ***Multiplicativity:*** *Let $X = (X_1, X_2)$ and $Y = (Y_1, Y_2)$. Let P_{X_1} and P_{Y_1} denote the probability distribution of X_1 and Y_1. Let $P_{X_2|X_1=x}$ and $P_{Y_2|Y_1=x}$ denote the distribution of X_2 conditioned on $X_1 = x$ as well as Y_2 conditioned on $Y_1 = x$. If X_1 and X_2 are independent and Y_1 and Y_2 are independent, then*

  $$R_a(P_X\|P_Y) = R_a(P_{X_1}\|P_{Y_1}) R_a(P_{X_2}\|P_{Y_2}).$$

  Otherwise

  $$R_a(P_X\|P_Y) \leq R_\infty(P_{X_1}\|P_{Y_1}) \cdot \max_{x \in \mathrm{Supp}(P_{X_1})} R_a(P_{X_2|X_1=x}\|P_{Y_2|Y_1=x}).$$

- ***Probability Preservation:*** *For any event $E \subseteq \mathrm{Supp}(P_Y)$,*

  $$P_Y(E) \geq \frac{P_X(E)^{\frac{a}{a-1}}}{R_a(P_X\|P_Y)}. \tag{2.2}$$

The Rényi divergence is non-decreasing and continuous as a function of $a \in [1, +\infty]$, as long as it is finite.

The following result is used in Chapter 6.

Lemma 2.3. *Let $\varepsilon < 1$. Let P and Q be two random variables taking values in some countable set Ω. Let $c \in \mathbb{R}$ be a constant such that*

$$\forall a \in \Omega : \ \Pr[Q = a] = c(1 - \delta(a)) \Pr[P = a] \ ,$$

for some function $\delta : \Omega \to [0, \varepsilon]$. Then it holds that:

$$R_\infty(P\|Q) \leq \frac{1}{1-\varepsilon} \ , \quad R_\infty(Q\|P) \leq \frac{1}{1-\varepsilon} \quad \textit{and} \quad \Delta(P,Q) \leq \frac{\varepsilon}{1-\varepsilon} \ .$$

Proof. Let us first note that $(1-\varepsilon)c \leq 1 \leq c$, by summing the above equality over all $a \in \Omega$ and applying the bounds on $\delta(a)$. Then we have

$$R_\infty(P\|Q) = \sup_{a\in\Omega} \frac{\Pr[P=a]}{\Pr[Q=a]} = \sup_{a\in\Omega} \frac{1}{c(1-\delta(a))} \leq \frac{1}{1-\varepsilon} \ .$$

We also have

$$R_\infty(Q\|P) = \sup_{a\in\Omega} \frac{\Pr[Q=a]}{\Pr[P=a]} = \sup_{a\in\Omega} c(1-\delta(a)) \leq c \leq \frac{1}{1-\varepsilon} \ .$$

Finally, we refer to [BF11, Lemma A.2] for the third bound. □

2.2.2.2 Smooth Rényi Divergence

We introduce a relaxed version of the Rényi divergence, termed the smooth Rényi divergence, where one is able to remove a few problematic points from the support, including those that may lie in $\mathrm{Supp}(p) \setminus \mathrm{Supp}(q)$. Doing so, we can compare a wider set of probability distributions. For instance, while the Rényi divergence of infinite order between the discrete Gaussian distributions $D_{\mathbb{Z}^m,\sigma}$ and $D_{\mathbb{Z}^m,\sigma,\mathbf{v}}$ (as defined in Section 2.3) is infinite when $\mathbf{v} \neq \mathbf{0}$, their smooth divergence is finite, as we show in Lemma 4.15 and is implicit in [Lyu12]. We could give this definition for any order $a \in [1, +\infty]$. However, only the case $a = +\infty$ is relevant for this work.

This definition is useful to link previous works on rejection sampling and the Rényi divergence. A similar quantity has been previously defined in the quantum information literature [Ren05, Dat09], though the specific notion of smoothing we consider here is slightly different.

Definition 2.4 (Smooth Rényi Divergence)**.** *Let $\varepsilon \geq 0$. Let p, q be two probability densities such that $\int_{\mathrm{Supp}(q)} p(x)\,\mathrm{d}\mu(x) \geq 1-\varepsilon$. Their ε-smooth Rényi divergence of infinite order is*

$$R^\varepsilon_\infty(p\|q) := \inf_{\substack{S\subseteq\mathrm{Supp}(q) \\ \int_S p(x)\,\mathrm{d}\mu(x)\geq 1-\varepsilon}} \operatorname*{ess\,sup}_{x\in S} \frac{p(x)}{q(x)}.$$

This definition is equivalent to

$$R^\varepsilon_\infty(p\|q) := \inf\{M > 0 \mid \Pr_{x\hookleftarrow p}(p(x) \leq Mq(x)) \geq 1-\varepsilon\}.$$

By convention, if $\int_{\mathrm{Supp}(q)} p(x)\,\mathrm{d}\mu(x) < 1-\varepsilon$, we define $R^\varepsilon_\infty(p\|q) = +\infty$.

We first prove that the two definitions of Definition 2.4 are indeed equivalent.

Proof. Let $R^{\varepsilon}_{\infty}(p\|q)$ be the first quantity and $\mathcal{R}^{\varepsilon}_{\infty}(p\|q)$ be the second one.

Let $S \subseteq \mathrm{Supp}(q)$ such that $\int_S p(x)\,\mathrm{d}\mu(x) \geq 1-\varepsilon$. Let $M = \sup_{x\in S} \frac{p(x)}{q(x)}$, i.e.

$$\Pr_{x\hookleftarrow p}[p(x) \leq Mq(x)] \geq \int_S p(x)\,\mathrm{d}\mu(x) \geq 1-\varepsilon.$$

Then $\mathcal{R}^{\varepsilon}_{\infty}(p\|q) \leq M$. By definition of $R^{\varepsilon}_{\infty}(p\|q)$, this implies the first inequality

$$\mathcal{R}^{\varepsilon}_{\infty}(p\|q) \leq R^{\varepsilon}_{\infty}(p\|q).$$

Now let $M > 0$ such that $\Pr_{x\hookleftarrow p}(p(x) \leq Mq(x)) \geq 1-\varepsilon$. Define

$$S := \big\{x \in \mathrm{Supp}(p) \cup \mathrm{Supp}(q) \mid p(x) \leq Mq(x)\big\}.$$

Then $\int_S p(x)\,\mathrm{d}\mu(x) \geq 1-\varepsilon$ by definition. Note that if we choose $S' = S \cap \mathrm{Supp}(p)$ we have $\int_{S'} p(x)\,\mathrm{d}\mu(x) = \int_S p(x)\,\mathrm{d}\mu(x)$ as we only removed elements that were not in the support of p. Moreover, assume that there exists $x \in S'$ such that $x \notin \mathrm{Supp}(q)$. We would have $p(x) \leq M \cdot 0 = 0$, contradicting the fact that $x \in \mathrm{Supp}(p)$. Then it holds that $S' = \{x \in \mathrm{Supp}(q) | p(x) \leq Mq(x)\}$. We have

$$M \geq \sup_{x\in S'} \frac{p(x)}{q(x)}, \int_{S'} p(x)\,\mathrm{d}\mu(x) \geq 1-\varepsilon \quad \text{and} \quad S' \subseteq \mathrm{Supp}(q).$$

This implies, by definition of $R^{\varepsilon}_{\infty}(p\|q)$ that $M \geq R^{\varepsilon}_{\infty}(p\|q)$. By definition of the second quantity, we have the inequality $R^{\varepsilon}_{\infty}(p\|q) \leq \mathcal{R}^{\varepsilon}_{\infty}(p\|q)$, thus completing the proof of the equality. □

We now give a few properties of the smooth Rényi divergence. The probability preservation and multiplicativity properties are used in the security proof of our signature variant with a bounded number of rejection steps as we define in Chapter 4. The comparison to the Rényi divergence is used to bound the smooth Rényi divergence between Gaussian distributions.

We start by proving a probability preservation property.

Lemma 2.4 (Probability Preservation)**.** *Let P, Q be two distributions. For any $\varepsilon \geq 0$ such that $R^{\varepsilon}_{\infty}(P\|Q)$ is finite, the following holds.*

$$\forall E \subseteq \mathrm{Supp}(P), P(E) \leq R^{\varepsilon}_{\infty}(P\|Q) \cdot Q(E) + \varepsilon.$$

Proof. Let $S := \{x \in \mathrm{Supp}(P) | P(x) \leq R^{\varepsilon}_{\infty}(P\|Q) \cdot Q(x)\}$. We decompose the event E into the disjoint union $(E \cap S) \cup (E \setminus S)$. The following holds:

- $P(E \cap S) \leq R^{\varepsilon}_{\infty}(P\|Q) \cdot Q(E \cap S) \leq R^{\varepsilon}_{\infty}(P\|Q) \cdot Q(E)$, by definition of S,
- $P(E \setminus S) \leq \varepsilon$, by definition of $R^{\varepsilon}_{\infty}(P\|Q)$.

Combining both inequalities yields the result. □

In the case of Fiat-Shamir signatures, the security loss bound may depend on the number of signing queries. The following result then proves useful.

Lemma 2.5 (Multiplicativity)**.** *Let (X_1, X_2) (resp. (Y_1, Y_2)) be a random variable with probability density $p_{X_1X_2}$ (resp. $p_{Y_1Y_2}$). Let p_{X_1} (resp. p_{Y_1}) be the density of X_1 (resp. Y_1). For any x, let $p_{X_2|X_1=x}$ (resp. $p_{Y_2|Y_1=x}$) denote the probability density of X_2 (resp. Y_2) conditioned on $X_1 = x$ (resp. $Y_1 = x$). For any $\varepsilon_1, \varepsilon_2 \geq 0$ it then holds that*

$$R_\infty^{\varepsilon_1+\varepsilon_2}(p_{X_1X_2}\|p_{Y_1Y_2}) \leq R_\infty^{\varepsilon_1}(p_{X_1}\|p_{Y_1}) \cdot \sup_{\substack{x\in\mathrm{Supp}(p_{X_1})\\ \cap\mathrm{Supp}(p_{Y_1})}} R_\infty^{\varepsilon_2}(p_{X_2|X_1=x}\|p_{Y_2|Y_1=x}).$$

Proof. Let $R_1 = R_\infty^{\varepsilon_1}(p_{X_1}\|p_{Y_1})$ and $R_2 = \sup_{x\in S_1} R_\infty^{\varepsilon_2}(p_{X_2|X_1=x}\|p_{Y_2|Y_1=x})$, where we let $S_1 = \mathrm{Supp}(p_{X_1}) \cap \mathrm{Supp}(p_{Y_1})$. If $R_1 = +\infty$, the statement is vacuously true. Assuming that this is not the case, we now define $R = R_1 \cdot R_2$ and

$$S = \big\{(x, y) \in \mathrm{Supp}(p_{X_1X_2}) \mid p_{X_1X_2}(x, y) > R \cdot p_{Y_1Y_2}(x, y)\big\}.$$

Any pair $(x, y) \in S$ satisfies $p_{X_1}(x)p_{X_2|X_1=x}(y) > Rp_{Y_1}(x)p_{Y_2|Y_1=x}(y)$ or it holds that $x \notin \mathrm{Supp}(P_{Y_1})$. This implies that it either holds that $p_{X_1}(x) > R_1 \cdot p_{Y_1}(x)$ or $p_{X_2|X_1=x}(y) > R_2 p_{Y_2|Y_1=x}(y)$. We then have, using the union bound,

$$\begin{aligned}\int_S p_{X_1X_2}(\mathbf{x})\,\mathrm{d}\mathbf{x} &\leq \Pr_{x\leftarrow p_{X_1}}[p_{X_1}(x) > R_1 \cdot p_{Y_1}(x)]\\ &\quad + \sum_{x\in S_1} p_{X_1}(x) \cdot \Pr_{y\leftarrow p_{X_2|X_1=x}}\big[p_{X_2|X_1=x}(y) > R_2 \cdot p_{Y_2|Y_1=x}(y)\big]\\ &\leq \varepsilon_1 + \sum_{x\in S_1} p_{X_1}(x)\varepsilon_2\\ &\leq \varepsilon_1 + \varepsilon_2.\end{aligned}$$

Define the set

$$\overline{S} := \mathrm{Supp}(p_{Y_1Y_2}) \setminus S = \{(x, y) \in \mathrm{Supp}(p_{Y_1Y_2}) | p_{X_1X_2}(x, y) \leq R \cdot p_{Y_1Y_2}(x, y)\}.$$

We have $\overline{S} = (\mathrm{Supp}(p_{X_1X_2}) \cup \mathrm{Supp}(p_{Y_1Y_2})) \setminus S$, as $\mathrm{Supp}(p_{X_1X_2}) \setminus \mathrm{Supp}(p_{Y_1Y_2}) \subseteq S$. Then it satisfies $\int_{\overline{S}} p_{X_1X_2}(\mathbf{x})\,\mathrm{d}\mathbf{x} \geq 1 - \varepsilon_1 - \varepsilon_2$. The first definition of the smooth divergence provides the result. □

Noticing that $R_a(P\|Q)^{a-1} = \mathbb{E}_{x\leftarrow P}((p(x)/q(x))^{a-1})$, we can apply concentration inequalities to compare the smooth divergence and the Rényi divergence, as was done in [RW04] for entropies. We however recall that the smooth Rényi divergence may be finite for pairs of random variables for which the Rényi divergence is infinite, in which case our bound is trivial.

Lemma 2.6. *Let X, Y be two discrete random variable with probability distributions P_X and P_Y. For any $\varepsilon \geq 0$ and order $a \in (1, +\infty)$ it holds*

$$R_\infty^\varepsilon(P_X\|P_Y) \leq \frac{R_a(P_X\|P_Y)}{\varepsilon^{1/(a-1)}} \quad \text{and} \quad R_\infty^\varepsilon(P_X\|P_Y) \leq R_\infty(P_X\|P_Y).$$

Proof. Markov's inequality gives that for any $t > 0$,

$$\Pr_{x\leftarrow P_X}\left(\left(\frac{P_X(x)}{P_Y(x)}\right)^{a-1} \geq t\right) \leq \frac{R_a(P_X\|P_Y)^{a-1}}{t}.$$

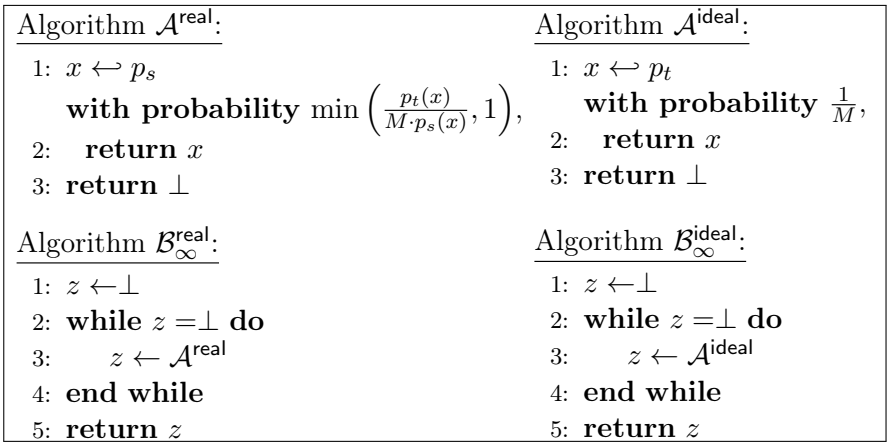

Figure 2.1: Rejection sampling algorithms.

Setting t_0 such that $R_a(P_X\|P_Y)^{a-1}/t_0 = \varepsilon$, we have:

$$\Pr_{x \hookleftarrow P_X}[P_X(x) \geq t_0^{1/(a-1)} \cdot P_Y(x)] \leq \varepsilon.$$

By the second definition of $R_\infty^\varepsilon(P_X\|P_Y)$, this shows

$$R_\infty^\varepsilon(P_X\|P_Y) \leq t_0^{1/(a-1)} = \frac{R_a(P_X\|P_Y)}{\varepsilon^{\frac{1}{a-1}}}.$$

To conclude the proof, recall that the Rényi divergence is continuous as a function of a. Taking the limit of this upper bound when a tends to $+\infty$ gives the second result. □

2.2.3 Rejection Sampling

Given two close enough densities p_t and p_s, either both continuous or both discrete, rejection sampling is a way to generate samples from p_t given access to samples from p_s, as explained for instance in [Dev86]. It was used mainly to generate samples from complex distributions that were "close" to easier-to-sample distributions. However, in cryptography and particularly in the line of works started with [Lyu09], it found a peculiar use that diverged from its primary use. Given a family of densities $(p_s^{(v)})$, rejection sampling can be used to hide the parameter v given a density p_t that is close to every density in this family. It was later observed in [Lyu12] that an "imperfect" rejection procedure is sufficient for this use and leads to smaller parameters, notably standard deviation of p_s.

We consider the following algorithms from Figure 2.1, which take some $M \geq 1$ as a parameter. Algorithm $\mathcal{B}_\infty^{\mathsf{real}}$ is the rejection sampling algorithm and $\mathcal{A}^{\mathsf{real}}$ is one iteration. Algorithm $\mathcal{B}_\infty^{\mathsf{ideal}}$ is the target distribution, which we break down for analysis. The following lemma, phrased here in full genericity and in terms of smooth Rényi divergence, expresses the closeness of the outputs of the algorithms both in terms of statistical distance and Rényi divergence.

Lemma 2.7 (Adapted from [Lyu12, Lemma 4.7])**.** *Let $M \geq 1$ and $\varepsilon \in [0, 1/2]$ such that*

$$\Pr_{z \hookleftarrow p_t}(p_t(z) \leq M \cdot p_s(z)) \geq 1 - \varepsilon,$$

which can be rewritten in terms of smooth Rényi divergence as $R^{\varepsilon}_{\infty}(p_t \| p_s) \leq M$. Then the probability $\mathcal{A}^{\mathsf{real}}(\perp)$ that $\mathcal{A}^{\mathsf{real}}$ aborts is such that

$$\frac{M-1}{M} \leq \mathcal{A}^{\mathsf{real}}(\perp) \leq \frac{M-1+\varepsilon}{M}.$$

We have $\Delta(\mathcal{A}^{\mathsf{real}}, \mathcal{A}^{\mathsf{ideal}}) \leq \varepsilon/M$ and $\Delta(\mathcal{B}^{\mathsf{real}}_{\infty}, \mathcal{B}^{\mathsf{ideal}}_{\infty}) \leq \varepsilon$ as well as

$$R_{\infty}(\mathcal{A}^{\mathsf{real}} \| \mathcal{A}^{\mathsf{ideal}}) \leq 1 + \frac{\varepsilon}{M-1} \quad \text{and} \quad R_{\infty}(\mathcal{B}^{\mathsf{real}}_{\infty} \| \mathcal{B}^{\mathsf{ideal}}_{\infty}) \leq \frac{1}{1-\varepsilon}.$$

Proof. Let $S = \mathrm{Supp}(p_t) \cup \mathrm{Supp}(p_s)$. Let us write for any $x \in \mathrm{Supp}(p_s)$:

$$\min\left(\frac{p_t(x)}{M p_s(x)}, 1\right) = \frac{\frac{1}{C} \cdot \min\left(p_t(x), M \cdot p_s(x)\right)}{\frac{M}{C} \cdot p_s(x)},$$

where C is normalization constant defined as

$$C = \int_{\mathrm{Supp}(p_s)} \min\left(p_t(x), M \cdot p_s(x)\right) \, \mathrm{d}x.$$

Notably, we have $1 \geq C \geq 1 - \varepsilon$, by giving $p_t(x)$ as an upper bound of the integrand in the first inequality, and by keeping only the set of x's such that $p_t(x) \leq M p_s(x)$ in the second inequality. We have that the function $p'_t : x \mapsto \min(p_t(x), M p_s(x))/C$ is a probability density satisfying $R_{\infty}(p'_t \| p_s) \leq M/C$. Then algorithm $\mathcal{A}^{\mathsf{real}}$ is a perfect rejection sampling algorithm with target density p'_t and source density p_s. The output density of $\mathcal{B}^{\mathsf{real}}$ is exactly p'_t, as explained in [Dev86, Chapter II.3] (see in particular [Dev86, Theorems 3.1 and 3.2]). Moreover, the probability that $\mathcal{A}^{\mathsf{real}}$ outputs nothing is

$$\mathcal{A}^{\mathsf{real}}(\perp) = 1 - \frac{C}{M} \in \left[\frac{M-1}{M}, \frac{M-1+\varepsilon}{M}\right],$$

and the density of $\mathcal{A}^{\mathsf{real}}$ is $x \mapsto (1 - \mathcal{A}^{\mathsf{real}}(\perp)) \cdot p'_t(x) = \min(p_t(x)/M, p_s(x))$.

Let us then bound the statistical distance.

$$\begin{aligned}
\Delta(\mathcal{A}^{\mathsf{real}}, \mathcal{A}^{\mathsf{ideal}}) &= \frac{1}{2}\int_S \left|\frac{p_t(x)}{M} - \min\left(\frac{p_t(x)}{M}, p_s(x)\right)\right| \mathrm{d}x + \frac{1}{2}\left|\mathcal{A}^{\mathsf{real}}(\perp) - \frac{M-1}{M}\right| \\
&\leq \frac{1}{2}\int_S \left|\max\left(0, \frac{p_t(x)}{M} - p_s(x)\right)\right| \mathrm{d}x + \frac{\varepsilon}{2M} \\
&\leq \frac{1}{2}\int_{\{x \in S | p_s(x) \leq p_t(x)/M\}} \left(\frac{p_t(x)}{M} - p_s(x)\right) \mathrm{d}x + \frac{\varepsilon}{2M} \\
&\leq \frac{\varepsilon}{2M} + \frac{\varepsilon}{2M},
\end{aligned}$$

by assumption on p_s and p_t.

To bound the statistical distance between $\mathcal{B}_\infty^{\mathsf{real}}$ and $\mathcal{B}_\infty^{\mathsf{ideal}}$, we first note that their distributions actually correspond to the distributions of $\mathcal{A}^{\mathsf{real}}$ and $\mathcal{A}^{\mathsf{ideal}}$ conditioned on the fact that they do not abort. Let $a = 2(1 - \mathcal{A}^{\mathsf{real}}(\perp))$. We have

$$\begin{aligned}
\Delta(\mathcal{B}_\infty^{\mathsf{real}}, \mathcal{B}_\infty^{\mathsf{ideal}}) &= \frac{1}{2}\int_{\mathrm{Supp}(p_t)} \left| p_t(x) - \frac{1}{1-\mathcal{A}^{\mathsf{real}}(\perp)} \min\left(\frac{p_t(x)}{M}, p_s(x)\right)\right| \mathrm{d}x \\
&= \frac{1}{a}\int_{\mathrm{Supp}(p_t)} \left| (1-\mathcal{A}^{\mathsf{real}}(\perp))p_t(x) - \min\left(\frac{p_t(x)}{M}, p_s(x)\right)\right| \mathrm{d}x \\
&= \frac{1}{a}\left[\int_{\substack{\mathrm{Supp}(p_t)\\ p_s(x)\geq p_t(x)/M}} \left|1-\mathcal{A}^{\mathsf{real}}(\perp) - \frac{1}{M}\right| p_t(x)\,\mathrm{d}x \right. \\
&\qquad \left. + \int_{\substack{\mathrm{Supp}(p_t)\\ p_s(x)< p_t(x)/M}} \left|(1-\mathcal{A}^{\mathsf{real}}(\perp))P_t(x) - P_s(x)\right| \mathrm{d}x \right].
\end{aligned}$$

Given the upper and lower bounds on $\mathcal{A}^{\mathsf{real}}(\perp)$, the first integral is bounded with:

$$\int_{\substack{x\in\mathrm{Supp}(p_t)\\ p_s(x)\geq p_t(x)/M}} \left|1-\mathcal{A}^{\mathsf{real}}(\perp) - \frac{1}{M}\right| p_t(x)\,\mathrm{d}x \;\leq\; \frac{\varepsilon}{M}\int_{\substack{x\in\mathrm{Supp}(p_t)\\ p_s(x)\geq p_t(x)/M}} p_t(x)\,\mathrm{d}x.$$

We now observe that:

$$1-\mathcal{A}^{\mathsf{real}}(\perp) \geq \frac{\int_{\{x\in\mathrm{Supp}(P_t)\mid P_s(x)\geq P_t(x)/M\}} p_t(x)\,\mathrm{d}x}{M}.$$

Then, when we multiply the left integral by $1/a$, we obtain:

$$\frac{1}{a}\int_{\substack{x\in\mathrm{Supp}(p_t)\\ p_s(x)\geq p_t(x)/M}} \left|1-\mathcal{A}^{\mathsf{real}}(\perp) - \frac{1}{M}\right| p_t(x)\,\mathrm{d}x \;\leq\; \frac{\varepsilon}{2}.$$

Next, we study the right integral. Note that since $\varepsilon \leq 1/2$, it holds that

$$0 \leq p_s(x) \leq \frac{p_t(x)}{M} \leq 2(1-\mathcal{A}^{\mathsf{real}}(\perp))p_t(x),$$

as $1-\mathcal{A}^{\mathsf{real}}(\perp) \geq (1-\varepsilon)/M \geq 1/(2M)$. Hence the right integral satisfies

$$\int_{\substack{x\in\mathrm{Supp}(p_t)\\ p_s(x)< p_t(x)/M}} \left|(1-\mathcal{A}^{\mathsf{real}}(\perp))p_t(x) - p_s(x)\right| \mathrm{d}x \leq (1-\mathcal{A}^{\mathsf{real}}(\perp))\varepsilon.$$

Finally, when divided by $2(1-\mathcal{A}^{\mathsf{real}}(\perp))$, we get $\varepsilon/2$ as an upper bound.

We move on to studying the divergences.

$$\begin{aligned}
R_a(\mathcal{A}^{\mathsf{real}}\|\mathcal{A}^{\mathsf{ideal}})^{a-1} &= \left[\int_{\mathrm{Supp}(p_s)} \frac{\left(p_s(x)\min\left(\frac{p_t(x)}{M\cdot p_s(x)}, 1\right)\right)^a}{(p_t(x)/M)^{a-1}}\,\mathrm{d}x\right] + \frac{(\mathcal{A}^{\mathsf{real}}(\perp))^a}{(\mathcal{A}^{\mathsf{ideal}}(\perp))^{a-1}} \\
&\leq \int_{\mathrm{Supp}(p_s)} \frac{\left(p_s(x)\frac{p_t(x)}{M\cdot p_s(x)}\right)^a}{(p_t(x)/M)^{a-1}}\,\mathrm{d}x + \frac{(1-(1-\varepsilon)/M)^a}{(1-1/M)^{a-1}} \\
&= \int_{\mathrm{Supp}(p_s)} \frac{p_t(x)}{M}\,\mathrm{d}x + \frac{M-1+\varepsilon}{M}\cdot\left(\frac{M-1+\varepsilon}{M-1}\right)^{a-1} \\
&\leq \frac{1}{M} + \frac{M-1+\varepsilon}{M}\cdot\left(1+\frac{\varepsilon}{M-1}\right)^{a-1}.
\end{aligned}$$

$\underline{\mathsf{Game\ Reprogram}_b:}$	$\underline{\mathsf{Reprogram}(x_2):}$
1: $H_0 \leftarrow U(Y^{X_1 \times X_2})$	1: $(x_1, x') \leftarrow D$
2: $H_1 := H_0$	2: $y \leftarrow U(Y)$
3: $b' \leftarrow \mathcal{A}^{H_b,\ \mathsf{Reprogram}(\cdot)}$ **return** b'	3: $H_1 := H_1^{(x_1,x_2)\mapsto y}$ **return** (x_1, x')

Figure 2.2: The reprogramming game.

We move on to bounding the second divergence. For any $x \in \mathrm{Supp}(p_s)$:

$$\mathcal{B}_\infty^{\mathsf{real}}(x) = \frac{\mathcal{A}^{\mathsf{real}}(x)}{1 - \mathcal{A}^{\mathsf{real}}(\perp)}.$$

This also holds for $\mathcal{B}_\infty^{\mathsf{ideal}}$ with $\mathcal{A}^{\mathsf{ideal}}$ instead of $\mathcal{A}^{\mathsf{real}}$. We obtain:

$$\begin{aligned} R_a(\mathcal{B}_\infty^{\mathsf{real}} \| \mathcal{B}_\infty^{\mathsf{ideal}})^{a-1} &= \int_{\mathrm{Supp}(p_s)} \frac{1}{M^{a-1}} \cdot \frac{\left(p_s(x) \min\left(\frac{p_t(x)}{M \cdot p_s(x)}, 1\right)\right)^a}{(\mathcal{A}^{\mathsf{real}}(\perp))^a (p_t(x)/M)^{a-1}} \\ &\leq \frac{M}{(1-\varepsilon)^a} \int_{\mathrm{Supp}(p_s)} \frac{\left(p_s(x) \min\left(\frac{p_t(x)}{M \cdot p_s(x)}, 1\right)\right)^a}{(p_t(x)/M)^{a-1}}. \end{aligned}$$

This sum was already computed just above and is at most $1/M$.

The continuity of $a \mapsto R_a(P_t \| P_s)$ at $a = +\infty$ gives the last bounds. □

2.2.4 Reprogramming the Random Oracle

We state the classical variant of [GHHM21, Proposition 2], used in Chapter 3.

Lemma 2.8 (Classical Adaptive Reprogramming)**.** *Let X_1, X_2, X' and Y be finite sets, and let D be a distribution on $X_1 \times X'$. Let $\mathcal{A}$ be a distinguisher playing in the reprogramming game in Figure 2.2 and making q classical queries to the random oracle and r classical queries to the* Reprogram *function. Then*

$$\left| \Pr[1 \Leftarrow \mathsf{Reprogram}_0^{\mathcal{A}}] - \Pr[1 \Leftarrow \mathsf{Reprogram}_1^{\mathcal{A}}] \right| \leq rq \cdot 2^{-\alpha},$$

where α is the min-entropy of the first component of D.

Proof. Note that the adversary makes q random oracle queries, implying that at most q input-output pairs of the random oracle are being revealed. If a reprogramming query does not coincide with these values, then the view of the adversary is identical for $b = 0$ and $b = 1$. For each reprogramming query, the probability of having a collision with the known random oracle values is at most $q \cdot 2^{-\alpha}$ since the input min-entropy of each reprogramming call is α. One can complete the proof by using the union bound. □

2.3 Gaussian Distributions and Smoothing Parameter

We recall properties of the discrete Gaussian distribution, a widely used distribution in lattice-based cryptography.

Definition 2.5 (Gaussian Distribution)**.** *Let $m \geq 1, \sigma > 0$, $\mathbf{v} \in \mathbb{R}^m$ and $C = 1/2$ in Chapter 4 or $C = \pi$ everywhere else. Define*

$$\rho_\sigma : \mathbf{x} \mapsto \exp\left(-C\frac{\|\mathbf{x}\|^2}{\sigma^2}\right).$$

The discrete Gaussian distribution with standard deviation parameter σ and center parameter $\mathbf{v}$ is defined as

$$D_{\mathbb{Z}^m,\sigma,\mathbf{v}} : \mathbf{z} \mapsto \frac{\rho_\sigma(\mathbf{z}-\mathbf{v})}{\rho_{\sigma,\mathbf{v}}(\mathbb{Z}^m)},$$

where we let $\rho_{\sigma,\mathbf{v}}(\mathbb{Z}^m)$ denote $\sum_{\mathbf{x}\in\mathbb{Z}^m}\rho_\sigma(\mathbf{x}-\mathbf{v})$. If $\mathbf{v} = 0$, we omit it in the subscript.

The divergence between two discrete Gaussian distributions is well-known (see, e.g., [LSS14, Lemma 4.2] for $a = 2$). We give a formulation that includes every order ≥ 1, while restricting our case to the $\mathbb{Z}^m$ lattice. It is proven by adapting their proof.

Lemma 2.9. *Let $m \geq 1$, $\sigma > 0$ and $\mathbf{v} \in \mathbb{Z}^m$. Then for any $a \in [1, +\infty)$:*

$$R_a(D_{\mathbb{Z}^m,\sigma}\|D_{\mathbb{Z}^m,\sigma,\mathbf{v}}) = \exp\left(aC\frac{\|\mathbf{v}\|^2}{\sigma^2}\right).$$

We also have $R_\infty(D_{\mathbb{Z}^m,\sigma}\|D_{\mathbb{Z}^m,\sigma,\mathbf{v}}) = +\infty$ if $\mathbf{v} \neq \mathbf{0}$.

We also consider bimodal Gaussian distributions in Chapter 4.

Definition 2.6 (Bimodal Gaussian Distribution)**.** *Let $m \geq 1$. The bimodal Gaussian distribution $BD_{\mathbb{Z}^m,\sigma,\mathbf{v}}$ with parameters $\sigma > 0$ and $\mathbf{v} \in \mathbb{R}^m$ is the distribution obtained by sampling $b \hookleftarrow U(\{-1, 1\})$, and returning $\mathbf{x} \hookleftarrow D_{\mathbb{Z}^m,\sigma,b\mathbf{v}}$. It can be expressed as*

$$BD_{\mathbb{Z}^m,\sigma,\mathbf{v}} : \mathbf{z} \mapsto \frac{1}{2}\left(D_{\mathbb{Z}^m,\sigma,\mathbf{v}}(\mathbf{z}) + D_{\mathbb{Z}^m,\sigma,-\mathbf{v}}(\mathbf{z})\right).$$

In particular, since $\rho_{\sigma,\mathbf{v}}(\mathbb{Z}^m) = \rho_{\sigma,-\mathbf{v}}(\mathbb{Z}^m)$ (which can be seen by reordering the sum), we can write

$$BD_{\mathbb{Z}^m,\sigma,\mathbf{v}}(\mathbf{z}) = \frac{1}{\rho_{\sigma,\mathbf{v}}(\mathbb{Z}^m)}\exp\left(\frac{-\|\mathbf{z}\|^2 - \|\mathbf{v}\|^2}{2\sigma^2}\right)\cosh\left(\frac{|\langle\mathbf{z},\mathbf{v}\rangle|}{\sigma^2}\right).$$

For spherical Gaussians, the upper and lower part of a vector are statistically independent. This is not the case anymore for general covariance matrices. The following lemma give the conditional distribution of the lower part of a Gaussian vector, given the upper part. The proof is adapted from the continuous setting.

Lemma 2.10 (Conditional distribution)**.** *Let $k \geq m > 0$, $\mathbf{\Sigma} \in \mathbb{R}^{k\times k}$ be a symmetric positive-definite matrix and $\mathbf{c} \in \mathbb{R}^k$. Write*

$$\mathbf{c} = \begin{pmatrix}\mathbf{c}_1\\ \mathbf{c}_2\end{pmatrix} \quad \text{and} \quad \mathbf{\Sigma} = \begin{pmatrix}\mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12}\\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22}\end{pmatrix},$$

where $\mathbf{c}_1 \in \mathbb{R}^{k-m}$ and $\mathbf{\Sigma}_{11} \in \mathbb{R}^{(k-m)\times(k-m)}$. Let $(Y_1^\top|Y_2^\top) \hookleftarrow D_{\mathbb{Z}^k,\mathbf{\Sigma},\mathbf{c}}$, where Y_1 takes values in $\mathbb{Z}^{k-m}$. Given any $\mathbf{y}_1 \in \mathbb{Z}^{k-m}$, the conditional distribution of Y_2 conditioned on $Y_1 = \mathbf{y}_1$ is $D_{\mathbb{Z}^m,\overline{\mathbf{\Sigma}},\overline{\mathbf{c}}}$, where

$$\overline{\mathbf{c}} = \mathbf{c}_2 + \mathbf{\Sigma}_{21}\,\mathbf{\Sigma}_{11}^{-1}(\mathbf{y}_1 - \mathbf{c}_1) \quad \text{and} \quad \overline{\mathbf{\Sigma}} = \mathbf{\Sigma}_{22} - \mathbf{\Sigma}_{21}\,\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}.$$

Proof. As $\mathbf{\Sigma}$ is symmetric and positive-definite, both $\mathbf{\Sigma}_{11}$ and $\mathbf{\Sigma}_{22}$ are also symmetric and positive-definite and thus invertible. This is shown by considering vectors of the form $(\mathbf{x}^\top|(\mathbf{0}^m)^\top)^\top$ or $((\mathbf{0}^{k-m})^\top|\mathbf{y}^\top)^\top$. Let us write the block inverse of $\mathbf{\Sigma}$ as follows:

$$\mathbf{\Sigma}^{-1} = \left(\begin{array}{c|c} \mathbf{\Sigma}_{11}^{-1} + \mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}^{-1}\overline{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}_{21}\,\mathbf{\Sigma}_{11}^{-1} & -\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}^{-1}\overline{\mathbf{\Sigma}}^{-1} \\ \hline -\overline{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}_{21}\,\mathbf{\Sigma}_{11}^{-1} & \overline{\mathbf{\Sigma}}^{-1} \end{array}\right) = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{pmatrix}.$$

This formula also ensures that $\overline{\mathbf{\Sigma}}$ is invertible, as it is a diagonal block of the positive definite symmetric matrix $\mathbf{\Sigma}^{-1}$.

Let $\mathbf{y}_2 \in \mathbb{Z}^m$. The probability that $Y_2 = \mathbf{y}_2$ conditioned on $Y_1 = \mathbf{y}_1$ is

$$\rho_{\mathbf{\Sigma},\mathbf{c}}\begin{pmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{pmatrix} \Big/ \sum_{\mathbf{y}\in\mathbb{Z}^m} \rho_{\mathbf{\Sigma},\mathbf{c}}\begin{pmatrix}\mathbf{y}_1\\ \mathbf{y}\end{pmatrix}.$$

Let us then study $\rho_{\mathbf{\Sigma},\mathbf{c}}((\mathbf{y}_1^\top|\mathbf{y}^\top)^\top)$ by expanding it and completing the square.

$$\rho_{\mathbf{\Sigma},\mathbf{c}}\begin{pmatrix}\mathbf{y}_1\\ \mathbf{y}\end{pmatrix} \sim \exp\left(-\pi\left((\mathbf{y}-\overline{\mathbf{c}})^\top\mathbf{S}_{22}(\mathbf{y}-\overline{\mathbf{c}})\right)\right) = \rho_{\overline{\mathbf{\Sigma}},\overline{\mathbf{c}}}(\mathbf{y}),$$

where the notation $\sim$ hides terms that do not depend on $\mathbf{y}$. Using the fact that the probability mass sums to 1, we obtain that the distribution of Y_2 conditioned on $Y_1 = \mathbf{y}_1$ is $D_{\mathbb{Z}^m,\overline{\mathbf{\Sigma}},\overline{\mathbf{c}}}$. □

As showed in [GPV08], Gaussian distributions can be sampled from by using Klein's algorithm [Kle00]. We will rely on the following variant.

Lemma 2.11 (Adapted from [BLP+13, Lemma 2.3])**.** *There is a* ppt *algorithm that, given a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_\ell)$ of a full-rank ℓ-dimensional lattice Λ, a positive definite symmetric matrix $\mathbf{\Sigma}$ and $\mathbf{c} \in \mathbb{R}^\ell$ returns a sample from $D_{\Lambda,\mathbf{\Sigma},\mathbf{c}}$, assuming that $\sqrt{\ln(2\ell+4)/\pi} \cdot \max_i \|\mathbf{\Sigma}^{-1/2}\mathbf{b}_i\| \leq 1$.*

Given a lattice $\Lambda \subseteq \mathbb{R}^k$ and $\varepsilon > 0$, the smoothing parameter $\eta_\varepsilon(\Lambda)$ of the lattice Λ is defined as the smallest σ such that $\rho_{1/\sigma}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \varepsilon$. The smoothing parameter satisfies the following two properties.

Lemma 2.12 ([ZXZ18, Theorem 2])**.** *Let $k > 1$ and $\varepsilon < 0.086k$. Let $\Lambda \subseteq \mathbb{R}^k$ be a full-rank lattice with basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_k)$. It holds that*

$$\eta_\varepsilon(\Lambda) \leq \sqrt{\frac{\ln(k-1+2k/\varepsilon)}{\pi}} \cdot \max_{i\leq k}\|\mathbf{b}_i\| \ .$$

Lemma 2.13 ([MR07])**.** *Let Λ be a k-dimensional full-rank lattice. Let $\varepsilon > 0$ and $\mathbf{\Sigma} \in \mathbb{R}^{k\times k}$ be a definite positive symmetric matrix with all singular values larger than $\eta_\varepsilon(\Lambda)$ and $\mathbf{c} \in \mathbb{R}^k$. We have*

$$\rho_{\mathbf{\Sigma},\mathbf{c}}(\Lambda) \ \in \ \frac{\sqrt{\det\mathbf{\Sigma}}}{\det\Lambda}\cdot[1-\varepsilon, 1+\varepsilon] \quad \textit{and} \quad \frac{\rho_{\mathbf{\Sigma},\mathbf{c}}(\Lambda)}{\rho_{\mathbf{\Sigma}}(\Lambda)} \ \in \ \left[\frac{1-\varepsilon}{1+\varepsilon}, 1\right] \ .$$

The following lemma (adapted from [MKMS22, Lemma 1]) is at the core of Chapter 6. While [MKMS22] does not give explicit statistical bounds, we note that Lemma 2.13 above, which is applied at the end of the proof from [MKMS22], allows us to do so when combined with Lemma 2.3. A further adaptation is the use of the smoothing parameter bound from Lemma 2.12. In their setting of rank-deficient Gaussian distributions, we note that Lemma 2.13 is extended by using the lattice rank in the bound instead of the dimension, by considering an appropriate rotation of the lattice, contrary to [MKMS22, Lemma 4], which considers undefined quantities as $\rho_{\mathbf{\Sigma}}(\mathbb{Z}^n)$ is infinite if $\mathbf{\Sigma}$ is not full rank.

Lemma 2.14 (Gaussian decomposition, [MKMS22, Lemma 1])**.** *Let $k \geq \ell$, $\varepsilon \in (0,1)$ and $\mathbf{S} \in \mathbb{Z}^{k \times \ell}$. Let $s \geq \sqrt{8\ln(\ell - 1 + 2\ell/\varepsilon)/\pi}$ and $\sigma \geq \sqrt{2}\sigma_1(\mathbf{S}) \cdot s$. Define*

$$\mathbf{\Sigma}(\mathbf{S}) = \sigma^2 \mathbf{I}_k - s^2 \mathbf{S}\mathbf{S}^\top \ ,$$

and let $\mathbf{y} \hookleftarrow D_{\mathbb{Z}^k, \mathbf{\Sigma}(\mathbf{S})}$ and $\mathbf{k} \hookleftarrow D_{\mathbb{Z}^\ell, s, -\mathbf{c}/2}$ for any $\mathbf{c} \in \mathbb{Z}^\ell$. Then $\mathbf{\Sigma}(\mathbf{S})$ is positive definite and the distribution $P_{\mathbf{z}}$ of $\mathbf{z} = \mathbf{y} + \mathbf{S}(2\mathbf{k} + \mathbf{c})$ satisfies

$$R_\infty(P_{\mathbf{z}} \| D_{\mathbb{Z}^k, \sigma}) \leq \frac{1+\varepsilon}{1-\varepsilon} \quad \text{and} \quad \Delta(P_{\mathbf{z}}, D_{\mathbb{Z}^k, \sigma}) \leq \frac{2\varepsilon}{1-\varepsilon} \ .$$

Note that the matrix $\mathbf{\Sigma}(\mathbf{S})$ is positive definite since $\sigma \geq \sqrt{2}\sigma_1(\mathbf{S}) \cdot s$ ensures that all singular values of $\sigma^2 \mathbf{I}_k$ are larger than those of $s^2 \mathbf{S}\mathbf{S}^\top$.

2.4 Cryptographic Primitives

We recall the definitions of the cryptographic primitives of interest in this work. Namely, digital signatures and Σ-protocols, which are turned into signatures via the Fiat-Shamir transform, which we also recall.

2.4.1 Signatures

Here we briefly recall the formalism of digital signatures.

Definition 2.7 (Digital Signature)**.** *A signature scheme is a tuple of* PPT *algorithms* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *with the following specifications:*

- $\mathsf{KeyGen} : 1^\lambda \to (\mathsf{vk}, \mathsf{sk})$ *outputs a verification key* vk *and a signing key* sk*;*
- $\mathsf{Sign} : (\mathsf{sk}, \mu) \to \sigma$ *takes as inputs a signing key* sk *and a message μ and outputs a signature σ;*
- $\mathsf{Verify} : (\mathsf{vk}, \mu, \sigma) \to b \in \{0,1\}$ *is deterministic, takes as inputs a verification key* vk*, a message μ, and a signature σ and outputs a bit $b \in \{0,1\}$.*

Let $\gamma > 0$. We say that it is γ-correct if for any pair $(\mathsf{vk}, \mathsf{sk})$ *in the range of* KeyGen *and μ,*

$$\Pr[\mathsf{Verify}(\mathsf{vk}, \mu, \mathsf{Sign}(\mathsf{sk}, \mu)) = 1] \geq \gamma,$$

where the probability is taken over the random coins of the signing algorithm. We say that it is correct in the ROM if the above holds when the probability is also taken over the randomness of the random oracle modeling the hash function used in the scheme.

We also recall the definition of existential unforgeability against chosen message attacks (UF-CMA).

Definition 2.8 (Security)**.** *Let $T, \delta \geq 0$. A signature scheme* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is said to be* (T, δ)-UF-CMA *secure in the ROM if for any* PPT *adversary $\mathcal{A}$ with runtime $\leq T$ given access to a signing oracle and random oracle H, it holds that*

$$\Pr_{(\mathsf{vk},\mathsf{sk})\leftarrow\mathsf{KeyGen}(1^\lambda)}[\mathsf{Verify}(\mathsf{vk}, \mu^*, \sigma^*) = 1 | (\mu^*, \sigma^*) \leftarrow \mathcal{A}^{H,\mathsf{Sign}}(\mathsf{vk})] \leq \delta,$$

where the randomness is also taken over the random coins of $\mathcal{A}$. The adversary should also not have issued a sign query for m^*. The above probability of forging a signature is called the advantage of $\mathcal{A}$ and denoted by $\mathsf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathsf{SIG}}(\mathcal{A})$. If $\mathcal{A}$ does not output anything, then it automatically fails.

*If we allow the adversary to forge a new signature for a previously queried message, the security is called strong existential unforgeability against chosen message attack (*sUF-CMA*). Existential unforgeability against one-per-message (resp. no-message) chosen message attack, denoted by* $\mathsf{UF\text{-}CMA}_1$ *(resp.* UF-NMA*) is defined similarly except that the adversary is allowed to query at most one (resp. not allowed to query any) signature per message. Further, one can similarly define* $\mathsf{sUF\text{-}CMA}_1$ *by taking the conjunction of* sUF-CMA *and* $\mathsf{UF\text{-}CMA}_1$*.*

Note that for deterministic signatures, the $\mathsf{UF\text{-}CMA}_1$ and UF-CMA security notions coincide.

2.4.2 Σ-Protocols and Identification Protocol

We start by recalling various definitions pertaining to identification protocols.

Definition 2.9 (Σ-Protocol with Aborts)**.** *Let $\mathcal{X}, \mathcal{Y}$ be two finite sets. A Σ-protocol for a relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ with commitment set $\mathcal{W}$, challenge set $\mathcal{C}$ and response set $\mathcal{Z}$ is a 3-round interactive proof system between a prover written as $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$ and a verifier $\mathsf{V} = (\mathsf{V}_1, \mathsf{V}_2)$ with the following specifications:*

- $\mathsf{P}_1 : (x, y) \to (w, st)$ *is a* PPT *algorithm that takes as input a pair of strings in $\mathcal{X} \times \mathcal{Y}$ and outputs a commitment $w \in \mathcal{W}$ and a state $st \in \{0,1\}^*$;*
- $\mathsf{V}_1 : (x, w) \to c$ *is a* PPT *algorithm that takes as inputs a string $x \in \mathcal{X}$ and a commitment $w \in \mathcal{W}$ and outputs a challenge $c \hookleftarrow U(\mathcal{C})$ independent of its input;*
- $\mathsf{P}_2 : (x, y, w, c, st) \to z$ *is a* PPT *algorithm that takes as inputs a pair of strings in $\mathcal{X} \times \mathcal{Y}$, a commitment $w \in \mathcal{W}$, a challenge $c \in \mathcal{C}$, and a state st and outputs a response $z \in \mathcal{Z} \cup \{\perp\}$ (we say that P_2 aborts if it outputs $\perp$);*
- $\mathsf{V}_2 : (x, w, c, z) \to b \in \{0, 1\}$ *is a deterministic polynomial-time algorithm that takes as inputs a string $x \in \mathcal{X}$, a commitment $w \in \mathcal{W}$, a challenge $c \in \mathcal{C}$, and a response $z \in \mathcal{Z}$ and outputs a bit b which represents acceptance or rejection; in the case that $z = \perp$, it returns 0.*

A Σ-protocol is said to be public-coin if V_1 outputs a challenge string c that is uniformly sampled from the challenge space $\mathcal{C}$, independently from its input.

Note that the above definition (and the following ones) is implicitly parameterized by the security parameter λ, that we omit for the sake of simplicity.

For cryptographic purposes, we must be able to efficiently generate the statement and the witness. This is captured in the following definition.

Definition 2.10 (Identification Protocol)**.** *An identification protocol is a Σ-protocol for an NP relation R, where the prover and verifier are dealt their statement and witness by a* PPT *instance generator* IGen*.*

Given a language $\mathcal{L} = \{x \in \mathcal{X} \mid \exists y \in \mathcal{Y} : (x, y) \in R\}$ for a relation $R \subseteq \mathcal{X} \times \mathcal{Y}$, we are interested in the following properties of an identification protocol.

Definition 2.11 (Completeness and commitment-recoverability)**.** *Let $\gamma, \beta > 0$. An identification protocol $\mathsf{ID} = (\mathsf{IGen}, (\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is (γ, β)-correct if for every $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{IGen}(1^\lambda)$ the following holds.*

- *If the response of the prover is not $\perp$, the verifier accepts with probability at least γ:*

$$\Pr\left[\mathsf{V}_2(\mathsf{vk}, w, c, z) = 1 \;\middle|\; \begin{array}{l}(w, st) \leftarrow \mathsf{P}_1(\mathsf{vk}, \mathsf{sk}),\\ c \leftarrow \mathsf{V}_1(\mathsf{vk}, w), z \leftarrow \mathsf{P}_2(\mathsf{vk}, \mathsf{sk}, w, c, st),\\ z \neq \perp\end{array}\right] \geq \gamma.$$

- *The probability that the prover aborts is bounded by β:*

$$\Pr\left[z = \perp \;\middle|\; \begin{array}{l}(w, st) \leftarrow \mathsf{P}_1(\mathsf{vk}, \mathsf{sk}),\\ c \leftarrow \mathsf{V}_1(\mathsf{vk}, w), z \leftarrow \mathsf{P}_2(\mathsf{vk}, \mathsf{sk}, w, c, st)\end{array}\right] \leq \beta.$$

The scheme satisfies commitment-recoverability *if for any public key* vk*, challenge $c \in \mathcal{C}$, and answer z, there is at most one commitment w such that the transcript (w, c, z) is valid, and there is a PPT algorithm* Rec *such that $w = \mathsf{Rec}(\mathsf{vk}, c, z)$.*

We let β denote *the probability of aborting* and we omit it if it is 0. We are interested in the regime of parameters in which $\gamma \geq 1 - \lambda^{-\omega(1)}$ and $\beta \leq 1 - 1/\mathsf{poly}(\lambda)$. Note that by repeating the protocol $\mathsf{poly}(\lambda)$ times, the parameter β is pushed toward 0, whereas γ stays close to 1.

We refer to the following definition as the one that is usually used in the literature of Fiat-Shamir with aborts. We however do not use it and we discuss our modifications in Section 3.4.1.

Definition 2.12 (No-Abort Statistical Honest-Verifier Zero-Knowledge)**.** *Let $\varepsilon_{zk} > 0$ and $T \geq 0$. An identification protocol is (ε_{zk}, T)-*naHVZK *if there exists a simulator* Sim *with runtime at most T, that given x, outputs a transcript (w, c, z) such that the distribution of (w, c, z) has statistical distance at most ε_{zk} from a honestly generated transcript (w', c', z') produced by the interaction conditioned on $z \neq \perp$.*

We then recall the definitions of *honest-verifier zero-knowledge* in the case of non-aborting identification protocols, which allow to reduce $\mathsf{EU\text{-}CMA}$ security of $\mathsf{FS}[\mathsf{ID}, H]$ to its $\mathsf{EU\text{-}NMA}$ security.

Definition 2.13 (HVZK)**.** *An identification scheme* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{P}, \mathsf{V})$ *is* Honest-Verifier Zero-Knowledge *if there exists a PPT simulator* Sim *such that one of the following holds for* $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{IGen}(1^\lambda)$*:*

- $\Delta((w, c, z) \leftarrow (\mathsf{P}(\mathsf{sk}, \mathsf{vk}) \leftrightarrow \mathsf{V}(\mathsf{vk}))\ ,\ \mathsf{Sim}(c, \mathsf{vk})) \leq \varepsilon$*. In this case, we say that* ID *is* ε*-HVZK.*
- $R_\infty((w, c, z) \leftarrow (\mathsf{P}(\mathsf{sk}, \mathsf{vk}) \leftrightarrow \mathsf{V}(\mathsf{vk})) \parallel \mathsf{Sim}(c, \mathsf{vk})) \leq 1 + \varepsilon$*. In this case, we say that* ID *is* $(1 + \varepsilon)$*-divergence HVZK.*

The challenge c can be sampled uniformly from the challenge space $\mathcal{C}$ and passed over as input to the simulator Sim.

A necessary statistical property of an identification protocol is the min-entropy of the commitments.

Definition 2.14 (Commitment Min-Entropy)**.** *Let* $\alpha > 0$*. An identification scheme* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{P}, \mathsf{V})$ *satisfies* α-Min-Entropy *or has* α *bits of commitment min-entropy if for any* $(\mathsf{vk}, \mathsf{sk})$ *in the range of* IGen*:*

$$H_\infty\Big(w | (w, c, z) \leftarrow (\mathsf{P}(\mathsf{sk}, \mathsf{vk}) \leftrightarrow \mathsf{V}(\mathsf{vk}))\Big) \geq \alpha\ .$$

Note that we could accommodate our results to schemes for which the above holds only with overwhelming probability over the randomness of IGen.

2.4.2.1 Additional properties for non-aborting Identification Schemes

Finally, we recall the notions of *lossiness* and *lossy-soundness*, which allow to prove $\mathsf{EU\text{-}NMA}$ security of $\mathsf{FS}[\mathsf{ID}, H]$ in the QROM.

Definition 2.15 (Lossiness and lossy-soundness)**.** *An identification scheme* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{P}, \mathsf{V})$ *is* lossy *and* $\varepsilon_{\mathsf{ls}}$-lossy sound *for some* $\varepsilon_{\mathsf{ls}} > 0$ *if there exists a PPT lossy key generation algorithm* $\mathsf{LossyIGen}$ *that, on input a security parameter, outputs a verification key* $\mathsf{vk}_{\mathsf{ls}}$ *such that* $\mathsf{vk}_{\mathsf{ls}}$ *is indistinguishable from a verification key* vk *generated by* IGen*.*

Moreover, for any (unbounded) P^* *interacting with* V*, we have:*

$$\Pr\Big[\mathsf{V}(\mathsf{vk}_{\mathsf{ls}}, (w, c, z)) = 1 \mid (w, c, z) \leftarrow (\mathsf{P}^*(\mathsf{vk}_{\mathsf{ls}}) \leftrightarrow \mathsf{V}(\mathsf{vk}_{\mathsf{ls}}))\Big] \ \leq\ \varepsilon_{\mathsf{ls}}\ .$$

If we only consider classical adversaries, $\mathsf{EU\text{-}NMA}$ security of $\mathsf{FS}[\mathsf{ID}, H]$ can be argued by relying on the simpler notion of *special soundness*.

Definition 2.16 (Special soundness)**.** *Let* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{P}, \mathsf{V})$ *be an identification scheme. It is* special sound *if for any PPT adversary* $\mathcal{A}$*, the quantity*

$$\Pr\Big[\mathsf{V}(\mathsf{vk}, (w, c_0, z_0)) = 1 \wedge \mathsf{V}(\mathsf{vk}, (w, c_1, z_1)) = 1 \mid (w, c_0, z_0, c_1, z_1) \leftarrow \mathcal{A}(\mathsf{vk})\Big]$$

is $\mathsf{negl}(\lambda)$*, where the probability is over the randomness of* $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{IGen}(1^\lambda)$ *and* $\mathcal{A}$*.*

$\mathsf{KeyGen}(1^\lambda)$:	$\mathsf{Sign}(\mathsf{sk}, \mu)$:	$\mathsf{Ver}(\mathsf{vk}, \mu, \sigma)$:
1: $(x, y) \leftarrow \mathsf{IGen}(1^\lambda)$	1: $\kappa := 1$	1: Parse $\sigma = (w, z)$
2: $(\mathsf{vk}, \mathsf{sk}) = (x, (x, y))$	2: **while** $z = \perp$ **and** $\kappa \leq B$	2: $c = H(w \| \mu)$
3: **return** $(\mathsf{vk}, \mathsf{sk})$	3: $(w, st) \leftarrow \mathsf{P}_1(\mathsf{sk})$	3: **return** $\mathsf{V}_2(\mathsf{vk}, w, c, z)$
	4: $c = H(w \| \mu)$	
	5: $z \leftarrow \mathsf{P}_2(\mathsf{sk}, w, c, st)$	
	6: $\kappa := \kappa + 1$	
	7: **if** $z = \perp$ **return** $\perp$	
	8: **return** $\sigma = (w, z)$	

Figure 2.3: Signatures $\mathsf{SIG}_B = \mathsf{FS}_B[\mathsf{ID}, H]$ and $\mathsf{SIG}_\infty = \mathsf{FS}_\infty[\mathsf{ID}, H]$. Signature SIG_B uses blocks highlighted with the blue color, whereas SIG_∞ does not.

2.4.3 Fiat-Shamir Transforms

In this section, we recall the Fiat-Shamir transform, which allows to transform an identification scheme into a digital signature. It removes interaction by sampling the challenge as a hash function evaluation $H(w, \mu)$ with w being the prover's commitment and μ the signed message. The hash function is then modeled as a random oracle in the analysis. The signature is the pair (w, z), which is verified by checking validity of the transcript $(w, H(w, \mu), z)$.

As the challenge c being typically much shorter than w, it is desirable to replace w by c in the signature. This is possible if the underlying identification scheme is commitment-recoverable (see Definition 2.11). Verification simply starts by recovering $w \leftarrow \mathsf{Rec}(\mathsf{vk}, c, z)$. Our protocols satisfy this property, thus we describe the signature obtained applying this version of the Fiat-Shamir transform.

2.4.3.1 The Fiat-Shamir with Aborts Transform

Let $\mathsf{ID} = (\mathsf{IGen}, (\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ be an identification protocol for a binary relation R. Further, let $H : \{0,1\}^* \to \mathcal{C}$ be a hash function where $\mathcal{C}$ is the challenge space of ID. Then, for every positive integer B, one can construct a signature scheme $\mathsf{SIG}_B = \mathsf{FS}_B[\mathsf{ID}, H]$ by applying the Fiat-Shamir transform with bounded aborts (FSwBA) as in Figure 2.3. We are particularly interested in applying the Fiat-Shamir transform without imposing a bound on the number of iterations in the rejection sampling as it is the case for Dilithium [DKL+18], among other schemes. One can define the unbounded version $\mathsf{SIG}_\infty = \mathsf{FS}_\infty[\mathsf{ID}, H]$ of the Fiat-Shamir transform as in Figure 2.3. Note that the signing algorithm of SIG_∞ may not be PPT as required in Definition 2.7. Ideally, it would still be expected polynomial-time.

Chapter 3 is dedicated to proving the properties that a signature resulting from a Fiat-Shamir with aborts transform inherits from its underlying identification protocol. For more details, we refer the reader to prior works (e.g., [Lyu09,Lyu12,AFLT16, DFMS19]).

- In [AFLT16, KLS18], the authors consider *lossy identification schemes* They reduce $\mathsf{UF\text{-}NMA}$ security of a signature based on the Fiat-Shamir transform to the $\varepsilon_{\mathsf{ls}}$-soundness of the underlying identification scheme and the indistinguishability of the outputs of IGen and $\mathsf{IGen}_{\mathsf{ls}}$.

$\mathsf{KeyGen}(1^\lambda)$:	$\mathsf{Sign}(\mathsf{sk}, \mu)$:	$\mathsf{Verify}(\mathsf{vk}, (c, z), \mu)$:
1: $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{IGen}(1^\lambda)$	1: $(w, \mathsf{st}) \leftarrow \mathsf{P}_1(\mathsf{sk})$	1: $w \leftarrow \mathsf{Rec}(\mathsf{vk}, c, z)$
2: **return** vk and sk	2: $c \leftarrow H(w, \mu)$	2: **if** $c \neq H(w, \mu)$ **then**
	3: $z \leftarrow \mathsf{P}_2(\mathsf{sk}, \mathsf{st}, w, c)$	3: **return** 0
	4: **return** (c, z)	4: **end if**
		5: **return** $\mathsf{V}(\mathsf{vk}, (w, c, z))$

Figure 2.4: Fiat-Shamir Signature $\mathsf{FS}[\mathsf{ID}, H]$.

- In [DFMS19] and implicitly in [Lyu09, Lyu12], the authors reduce UF-NMA security of a signature based on the Fiat-Shamir transform to the *proof of knowledge* property of the underlying identification protocol. Their reduction is less tight than the one of [KLS18].

2.4.3.2 The Fiat-Shamir Transform

We now describe the Fiat-Shamir with aborts when the identification protocol has probability 0 of aborting as well as its known properties.

For the sake of completeness, we state the following lemma arguing correctness of the signature scheme $\mathsf{FS}[\mathsf{ID}, H]$, which immediately follows from the completeness and commitment-recoverability of the underlying identification scheme.

Lemma 2.15. *Let* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{P}, \mathsf{V})$ *denote an identification scheme. Further assume that* ID *is* ε*-complete and commitment-recoverable. Then the signature scheme* $\mathsf{FS}[\mathsf{ID}, H]$ *described in Figure 2.4 is* ε*-correct in the ROM.*

Security of $\mathsf{FS}[\mathsf{ID}, H]$ can be proven by successive claims. First, one can reduce EU-CMA security of $\mathsf{FS}[\mathsf{ID}, H]$ to its EU-NMA security assuming ID has large commitment min-entropy and is honest-verifier zero-knowledge (see Definition 2.13). This can be shown by relying on the following theorem.

Theorem 2.16 (Adapted from [GHHM21, Theorem 3])**.** *Let* ID *be an identification scheme which has* α*-min-entropy and satisfies* ε*-statistical HVZK. Let* H *a hash function modeled as a random oracle. Then, for any (possibly quantum) adversary* $\mathcal{A}$ *against the EU-CMA security of* $\mathsf{FS}[\mathsf{ID}, H]$ *making at most* Q_S *(classical) sign queries and at most* Q_H *(possibly quantum) hash queries, there exists an adversary* $\mathcal{B}$ *against the EU-NMA security of* $\mathsf{FS}[\mathsf{ID}, H]$ *such that:*

$$\mathsf{Adv}^{\mathsf{EU-CMA}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{EU-NMA}}(\mathcal{B}) + Q_s\varepsilon + 3\frac{Q_S}{2} \cdot \sqrt{(Q_H + Q_S + 1) \cdot 2^{-\alpha}} \ .$$

Furthermore, if ID *is* $(1+\varepsilon)$*-divergence HVZK, the following bound applies:*

$$\mathsf{Adv}^{\mathsf{EU-CMA}}(\mathcal{A}) \leq (1+\varepsilon)^{Q_s}\mathsf{Adv}^{\mathsf{EU-NMA}}(\mathcal{B}) + 3Q_S/2 \cdot \sqrt{(Q_H + Q_S + 1) \cdot 2^{-\alpha}} \ .$$

The result can be adapted to sEU-CMA security by adding $Q_S 2^{-\alpha}$ *to the bounds.*

It remains to prove EU-NMA-security. It can be argued via the following statement for lossy identification schemes (see Definition 2.15).

Theorem 2.17 ([KLS18, Theorem 3.4]). *Let* ID *be a lossy identification scheme satisfying* $\varepsilon_{\mathsf{ls}}$*-lossy soundness for some* $\varepsilon_{\mathsf{ls}} > 0$*. Let* H *a hash function modeled as a random oracle. For any (possibly quantum) adversary* $\mathcal{A}$ *against the* EU-NMA *security of* $\mathsf{FS}[\mathsf{ID}, H]$ *making at most* Q_H *(possibly quantum) hash queries, there exists a quantum adversary* $\mathcal{B}$ *against the lossiness of* ID *such that*

$$\mathsf{Adv}(\mathcal{A}) \leq \mathsf{Adv}(\mathcal{B}) + 8(Q_H + 1)^2 \cdot \varepsilon_{\mathsf{ls}} \ .$$

Finally, we describe a reduction in the (classical) ROM which relies on weaker properties compared to the above QROM reduction. Various folklore reductions are known in this setting, and we consider a variant based on special soundness (see Definition 2.16), which is first reduced to the soundness as recalled below.

Definition 2.17 (Soundness). *Let* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{P}, \mathsf{V})$ *be an identification scheme. It is* sound *if for any PPT adversary* $\mathcal{A}$*, the quantity*

$$\Pr\left[\mathsf{V}(\mathsf{vk}, (w, c, z)) = 1 \mid (w, c, z) \leftarrow \mathcal{A}(\mathsf{vk})\right]$$

is $\mathsf{negl}(\lambda)$*, where the probability is over the choice of* vk *and the coins of* $\mathcal{A}$*.*

We recall the Reset Lemma, which is a reduction from special soundness to soundness.

Lemma 2.18 (Reset Lemma [BP02]). *Let* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{P}, \mathsf{V})$ *be an identification scheme. Given any adversary* $\mathcal{A}$ *against the soundness of* ID*, there exists an adversary* $\mathcal{B}$ *against the special soundness of* ID *such that*

$$\mathsf{Adv}^{\mathsf{special-sound}}(\mathcal{B}) \geq \left(\mathsf{Adv}^{\mathsf{sound}}(\mathcal{A}) - \frac{1}{|\mathcal{C}|}\right)^2 .$$

And we show that special soundness implies EU-NMA security in the ROM.

Lemma 2.19. *Let* ID *be an identification scheme and* H *a hash function modeled as a random oracle. For any adversary* $\mathcal{A}$ *against the EU-NMA security of* $\mathsf{FS}[\mathsf{ID}, H]$ *making* Q_H *classical hash queries, there exists an adversary* $\mathcal{B}$ *against the special soundness of* ID *such that:*

$$\mathsf{Adv}^{\mathsf{EU-NMA}}(\mathcal{A}) \leq Q_H \cdot \left(\sqrt{\mathsf{Adv}^{\mathsf{special-sound}}(\mathcal{B})} + \frac{2}{|\mathcal{C}|}\right) \ .$$

Proof. We first reduce the soundness of ID to the EU-NMA security of $\mathsf{FS}[\mathsf{ID}, H]$. First, if $\mathcal{A}$ outputs a forgery $(\mu^*, (c^*, z^*))$ such that $H(\mathsf{Rec}(\mathsf{vk}, c^*, z^*), \mu^*)$ was never queried, it has probability at most $1/|\mathcal{C}|$ of outputting a valid forgery.

The reduction $\mathcal{B}'$ guesses the hash query $H(w^*, \mu^*)$ made by $\mathcal{A}$ which is used in $\mathcal{A}$'s forgery. When this query is made, $\mathcal{B}'$ answers it by running sending w^* as commitments to its challenger. The latter replies with a challenge c^* and $\mathcal{B}'$ programs $H(w^*, \mu^*)$ as c^*. With probability $1/Q_H$, $\mathcal{B}'$'s guess is correct and the adversary $\mathcal{A}$ halts with a forgery $(\mu^*, (c^*, z^*))$ with $\mathsf{Rec}(\mathsf{vk}, c^*, z^*) = w^*$. We then have

$$\mathsf{Adv}^{\mathsf{sound}}(\mathcal{B}') \geq \frac{1}{Q_H} \cdot \mathsf{Adv}^{\mathsf{EU-NMA}}(\mathcal{A}) - 1/|\mathcal{C}| \ .$$

Finally, Lemma 2.18 gives an adversary $\mathcal{B}$ against the special soundness such that

$$\mathsf{Adv}^{\mathsf{special-sound}}(\mathcal{B}) \geq \left(\mathsf{Adv}^{\mathsf{sound}}(\mathcal{B}') - \frac{1}{|\mathcal{C}|}\right)^2,$$

which completes the proof. □

2.5 Lattice-based Security Assumptions

The Learning With Errors (LWE) and Short Integer Solution (SIS) problems serve as security foundation of Lyubashevsky's signature schemes. In the parameter instantiation section, we will use their module counterparts (see [LS15]).

Definition 2.18 (SIS)**.** *Let $m \geq n \geq 1$, $q \geq 2$ and $\beta > 0$. The* SIS *problem with parameters m, n, q, β is as follows: given as input $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n\times m})$, the goal is to find $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{Ax} = \mathbf{0} \bmod q$ and $0 < \|\mathbf{x}\| \leq \beta$.*

Definition 2.19 (LWE)**.** *Let $m \geq n \geq 1$, $q \geq 2$ and χ a distribution over $\mathbb{Z}_q$. The* LWE *problem with parameters m, n, q, χ consists in distinguishing between the distributions $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ and $(\mathbf{A}, \mathbf{u})$, where $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m\times n})$, $\mathbf{u} \leftarrow U(\mathbb{Z}_q^n)$, $\mathbf{s} \leftarrow \chi^n$ and $\mathbf{e} \leftarrow \chi^m$.*

2.6 The Lyubashevsky and BLISS Σ-protocols

All the following parameters are functions of a security parameter λ. Let $k, m, n \geq 1$ and $q \geq 2$ specify matrix spaces over $\mathbb{Z}_q$, with $m > n$. The distribution $P_{\mathbf{S}}$ over $\mathbb{Z}^{m\times k}$ is for signing keys and has support $\mathcal{S} = \mathrm{Supp}(P_{\mathbf{S}})$. Let $\mathcal{M}$ be the message space. Let $\mathcal{C} \subset \mathbb{Z}^k$ finite and $H : \mathbb{Z}_q^n \times \mathcal{M} \to \mathcal{C}$ a hash function, which is modeled as a random oracle in the signature scheme analysis. The parameter $\gamma > 0$ is used in the verification algorithm to quantify the smallness of the answer. To obtain a 2^λ security against known attacks, one typically sets $m, n, k = \Omega(\lambda)$ and $\gamma, q = \mathsf{poly}(\lambda)$.

Let $\varepsilon \geq 0$ and $M \geq 1$ be parameters related to rejection sampling, for a source distribution Q and a target distribution P over $\mathbb{Z}^m$. Most works directly instantiate these distributions. For example, uniform distributions in well-chosen hypercubes are used in [Lyu09] and $P = Q$ Gaussian are used in [Lyu12, DDLL13]. We assume that the support of Q is contained in $(-q/2, q/2]^m$.

We consider the Σ-protocols presented in Figure 2.5 borrowed from both [Lyu09] and [DDLL13] with the aforementioned rejection sampling generalization. For simplicity, we assume that the verification key $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ is in Hermite normal form, i.e., we have $\mathbf{A} = (\mathbf{B}|\mathbf{I}_n)$ for some matrix $\mathbf{B}$ and with $\mathbf{I}_n \in \mathbb{Z}_q^{n\times n}$ denoting the identity matrix. Up to mild conditions on k, n, m, q, this is without loss of generality.

For Lyubashevsky's and BLISS [DDLL13] Σ-protocols, the public key is $\mathbf{T} = \mathbf{AS}$ and $\mathbf{A}$. In the case of BLISS, it holds that $\mathbf{T} = -\mathbf{T} \bmod q$, which is usually achieved by setting $q = 2q'$ with $q' > 2$ a prime and generating $\mathbf{A}, \mathbf{S}$ such that $\mathbf{T} = q\mathbf{Id}$. Correctness of the Fiat-Shamir with aborts transform follows from the lemma below.

Lemma 2.20 (Correctness)**.** *Let $\varepsilon \geq 0$ and $M \geq 1$. Let P and Q such that it holds that $\max_{(\mathbf{S},\mathbf{c})\in\mathcal{S}\times\mathcal{C}} R_\infty^\varepsilon(P\|Q_{+\mathbf{Sc}}) \leq M$. Let $(\mathbf{y}_0^\top|\mathbf{y}_1^\top)^\top \hookleftarrow Q$, where $\mathbf{y}_0$ takes value*

$\mathsf{P}(\mathbf{A},\mathbf{S})$		$\mathsf{V}(\mathbf{A},\mathbf{T})$
$\mathbf{y} \hookleftarrow Q$		
$\mathbf{w} \leftarrow \mathbf{A}\mathbf{y}$	$\xrightarrow{\mathbf{w}}$	
	$\xleftarrow{\mathbf{c}}$	$\mathbf{c} \hookleftarrow U(\mathcal{C})$
$\mathbf{z} \leftarrow \mathbf{y}+\mathbf{S}\mathbf{c}$		
w.p. $\frac{P(\mathbf{z})}{M \cdot Q(\mathbf{y})}$	$\xrightarrow{\mathbf{z}}$	Accept if
else		$\mathbf{A}\mathbf{z}=\mathbf{w}+\mathbf{T}\mathbf{c}$
abort		and $\|\mathbf{z}\| \leq \gamma$

$\mathsf{P}(\mathbf{A},\mathbf{S})$		$\mathsf{V}(\mathbf{A},\mathbf{T})$
$\mathbf{y} \hookleftarrow Q$		
$\mathbf{w} \leftarrow \mathbf{A}\mathbf{y}$	$\xrightarrow{\mathbf{w}}$	
	$\xleftarrow{\mathbf{c}}$	$\mathbf{c} \hookleftarrow U(\mathcal{C})$
$b \hookleftarrow U(\{-1,1\})$		
$\mathbf{z} \leftarrow \mathbf{y} + b\mathbf{S}\mathbf{c}$		
w.p. $\frac{P(\mathbf{z})}{MQ_{\pm\mathbf{S}\mathbf{c}}(\mathbf{z})}$	$\xrightarrow{\mathbf{z}}$	Accept if
else		$\mathbf{A}\mathbf{z}=\mathbf{w}+\mathbf{T}\mathbf{c}$
abort		and $\|\mathbf{z}\| \leq \gamma$

Figure 2.5: Left-hand side: Lyubashevsky's Σ-protocol. Right-hand side: BLISS underlying Σ-protocol. All computations are done mod$q$.

in $\mathbb{Z}^n$. *Further assume that* $2^{-H_\infty(\mathbf{y}_0|\mathbf{y}_1)_Q} \leq \mathsf{negl}(\lambda)$, $\varepsilon \leq \mathsf{negl}(\lambda)$ *and the probability that* Sign *terminates is* $\geq 1 - \mathsf{negl}(\lambda)$. *Then, in the ROM, Lyubashevsky's signature is correct if* $\gamma \geq \gamma_P$ *with* γ_P *such that* $\Pr_{\mathbf{z}\hookleftarrow P}(\|\mathbf{z}\| \geq \gamma_P) \leq \mathsf{negl}(\lambda)$.

The statement holds for BLISS by replacing $Q_{+\mathbf{S}\mathbf{c}}$ with $Q_{\pm\mathbf{S}\mathbf{c}}$. Runtime and security are studied in Chapter 3.

Chapter 3

Security Analysis of Fiat-Shamir with Aborts

While many different proofs of the Fiat-Shamir with Aborts (FSwA) paradigm have been proposed, we found that many of them were flawed. This chapter proposes a way to fix them. In this chapter, we use the notations FSwBA (Fiat-Shamir with Bounded Aborts) when there is a bound B on the number of iterations of the signature algorithm. If this threshold is reached, then the signature algorithm fails. When there is no such bound, we use the notation FSwUA (Fiat-Shamir with Unbounded Aborts).

As we discussed in the introduction, Definition 2.12 is not sufficient for our purposes. Instead we consider the following statistical HVZK definition, which benefits from a simulator even for aborting transcripts of the Σ-protocol. One can see this modification as a return to the classic definition in the literature of the zero-knowledge interactive proof systems.

Definition 3.1 (Statistical Honest-Verifier Zero-Knowledge)**.** *Let $\varepsilon_{zk}, T \geq 0$. A Σ-protocol is (ε_{zk}, T)-*HVZK *if there exists a simulator* Sim *with runtime at most T, that given x, outputs a transcript (w, c, z) such that the distribution of (w, c, z) has statistical distance at most ε_{zk} from a honestly generated transcript (w', c', z') produced by the interaction. This includes aborting transcripts, i.e., those for which $z = \perp$.*

The challenge c can be sampled uniformly from the challenge space $\mathcal{C}$ and passed over as input to the simulator Sim.

We give a corrected analysis of FSwBA in Section 3.1. We extend this result to FSwUA with a tweak in the runtime definition in Section 3.3, as we also show that the usual definition may not be satisfied in Section 3.2. Finally, we show that the Lyubahsevsky and BLISS identification protocols satisfy Definition 3.1 in Section 3.4. In parallel, we study the benefits of a Rényi divergence-based approach in Sections 3.1.2 and 3.4.1.3. This chapter focuses on the ROM analysis of the transform. However, further results concerning the QROM analyses can be found in [DFPS23].

3.1 ROM Analysis of FSwBA

In this section we discuss the security of the Fiat-Shamir transform with bounded aborts. We prove the UF-CMA security of the signature in the ROM based on the adaptive reprogramming technique from [GHHM21].

We also propose a tweak of the analysis which relies on the Rényi divergence instead of the statistical distance, as we show that in the case of Lyubashevsky's signature, this leads to increased performances.

3.1.1 The Adaptive Reprogramming Approach

We show how to reduce UF-CMA security and sUF-CMA security of the signature to UF-NMA security in the ROM. We use the framework for adaptive reprogramming (Lemma 2.8). Also, we note that our proof is crucially based on our new zero-knowledge simulator.

Theorem 3.1. *Let $\varepsilon_{zk}, \alpha, T_{\mathsf{Sim}} \geq 0$, $B \geq 0$ and H a hash function modeled as a random oracle. Assume that $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is a $(\varepsilon_{zk}, T_{\mathsf{Sim}})$-HVZK public-coin identification protocol and that the commitment message of the prover has min-entropy α. Let $\mathcal{A}$ be a ppt adversary against UF-CMA security of $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ that issues at most Q_H queries to the random oracle H and Q_S queries to the signing oracle. Let $\mathsf{X} \in \{\mathsf{UF}, \mathsf{sUF}\}$; we define Δ_X as follows: $\Delta_{\mathsf{UF}} = 0$ and $\Delta_{\mathsf{sUF}} = BQ_S \cdot 2^{-\alpha}$.*

In the ROM, there exists an adversary $\mathcal{B}$ against UF-NMA security of SIG_B with runtime $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H) \log(B \cdot Q_S + Q_H))$ such that

$$\mathsf{Adv}^{\mathsf{X\text{-}CMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) + 2^{-\alpha} \cdot B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1) + \varepsilon_{zk} \cdot B \cdot Q_S + \Delta_{\mathsf{X}} \ .$$

The result also holds if we replace HVZK by sc-HVZK and assume ε_{zk} to be negligible in the security parameter.

Proof. The proof is based on a sequence of hybrid games.

Game G_0. The first game is the UF-CMA security game (Figure 3.1).

Game G_1. In this game, the challenges of the transcripts are not computed by the random oracle anymore, but sampled independently and uniformly each time. Then, the random oracle is reprogrammed according to the new challenges as in Figure 3.2.

To bound the distance between Game_0 and Game_1, we construct a wrapper $\mathcal{D}$ around $\mathcal{A}$ that uses $\mathcal{A}$ to solve a reprogramming game. It works as in Figure 3.3.

Note that if $b = 0$ in Figure 3.3, then $\mathcal{D}$ perfectly simulates G_0, and otherwise it perfectly simulates G_1. Therefore,

$$\left|\Pr[1 \Leftarrow G_0^{\mathcal{A}}] - \Pr[1 \Leftarrow G_1^{\mathcal{A}}]\right| \leq \left|\Pr[1 \Leftarrow \mathsf{Reprogram}_0^{\mathcal{D}}] - \Pr[1 \Leftarrow \mathsf{Reprogram}_1^{\mathcal{D}}]\right|.$$

During the game, distinguisher $\mathcal{D}$ makes $B \cdot Q_S$ reprogramming queries and $B \cdot Q_S + Q_H + 1$ random oracle queries. In the ROM, Lemma 2.8 bounds the advantage of $\mathcal{D}$ by $B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1)2^{-\alpha}$.

Game G_2. Let Sim be the zero-knowledge simulator for Σ. In this game we modify GetTrans such that the transcripts are now produced by Sim and without the secret key. See Figure 3.4.

Game :

1: $\mathcal{M} := \varnothing$
2: $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
3: $(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{H,\ \mathsf{Sign}(sk,\cdot)}(vk)$
4: Parse $\sigma^* = (w^*, z^*)$
5: $c^* := H(w^* \| \mu^*)$
6: **return** $[[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

Sign(sk, μ) **:**

1: $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
2: $(w, c, z) \leftarrow \mathsf{GetTrans}(\mu)$
3: **if** $z = \perp$ **return** $\perp$
4: **return** $\sigma = (w, z)$

GetTrans(μ) **:**

1: $\kappa := 0$
2: **while** $z = \perp$ and $\kappa \leq B$
3: $(w, st) \leftarrow \mathsf{P}_1(sk)$
4: $c := H(w \| \mu)$
5: $z \leftarrow \mathsf{P}_2(sk, w, c, st)$
6: $\kappa := \kappa + 1$
7: **return** (w, c, z)

Figure 3.1: Game G_0

Game :

1: $\mathcal{M} := \varnothing$
2: $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
3: $(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{H,\ \mathsf{Sign}(sk,\cdot)}(vk)$
4: Parse $\sigma^* = (w^*, z^*)$
5: $c^* := H(w^* \| \mu^*)$
6: **return** $[[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

Sign(sk, μ) **:**

1: $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
2: $(w, c, z) \leftarrow \mathsf{GetTrans}(\mu)$
3: **if** $z = \perp$ **return** $\perp$
4: **return** $\sigma = (w, z)$

GetTrans(μ) **:**

1: $\kappa := 0$
2: **while** $z = \perp$ and $\kappa \leq B$
3: $(w, st) \leftarrow \mathsf{P}_1(sk)$
4: $c \leftarrow U(\mathcal{C})$
5: $z \leftarrow \mathsf{P}_2(sk, w, c, st)$
6: $H = H^{w\|\mu \mapsto c}$
7: $\kappa := \kappa + 1$
8: **return** (w, c, z)

Figure 3.2: Game G_1. The difference from G_0 is highlighted in blue.

We would like to bound the distance between games G_1 and G_2 using the zero-knowledge property. Suppose that we are given a $B \cdot Q_S$ transcripts that are either sampled honestly or sampled by the simulator. We use them to simulate G_1 or G_2, respectively, by simulating H with the lazy sampling technique. Note that in both games, after each transcript, the random oracle is reprogrammed according to the transcript. In order to simulate the reprogrammed random oracle perfectly, we keep track of a list $\mathfrak{D}$ of the values in which the random oracle must be reprogrammed. We describe the details in Figure 3.5.

Note that $\mathcal{C}$ can perfectly simulate G_1 or G_2 with its respective transcripts. Furthermore, it is given $B \cdot Q_S$ transcripts. By the statistical HVZK property of the Σ-protocol, it follows that

$$\left| \Pr[1 \Leftarrow G_1^{\mathcal{A}}] - \Pr[1 \Leftarrow G_2^{\mathcal{A}}] \right| \leq B \cdot Q_S \cdot \varepsilon_{zk}.$$

Game G_3. In this game, we add one more statement to the winning conditions. Let $(\mu^*, (w^*, z^*))$ be the forgery. If the value $w^* \| \mu^*$ has been programmed in the

$\underline{\mathcal{D}^{H_b,\mathsf{Reprogram}}:}$
1: $\mathcal{M} := \varnothing$
2: $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
3: $(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{H_b,\ \mathsf{Sign}(sk,\cdot)}(vk)$
4: Parse $\sigma^* = (w^*, z^*)$
5: $c^* := H_b(w^* \| \mu^*)$
6: **return** $[[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

$\underline{\mathsf{Reprogram}(\mu, sk):}$
1: $(w, st) \leftarrow \mathsf{P}_1(sk)$
2: $c \leftarrow U(\mathcal{C})$
3: $H_1 := H_1^{(w\|\mu)\mapsto c}$
4: **return** (w, st)

$\underline{\mathsf{Sign}(sk, \mu):}$
1: $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
2: $\kappa := 0$
3: **while** $z = \perp$ and $\kappa \leq B$
4: $(w, st) \leftarrow \mathsf{Reprogram}(\mu, sk)$
5: $c := H_b(w \| \mu)$
6: $z \leftarrow \mathsf{P}_2(sk, w, c, st)$
7: $\kappa := \kappa + 1$
8: **if** $z = \perp$ **return** $\perp$
9: **return** $\sigma = (w, z)$

Figure 3.3: The distinguisher $\mathcal{D}$.

$\underline{\mathsf{Game}:}$
1: $\mathcal{M} := \varnothing$
2: $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
3: $(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{H,\ \mathsf{Sign}(sk,\cdot)}(vk)$
4: Parse $\sigma^* = (w^*, z^*)$
5: $c^* := H(w^* \| \mu^*)$
6: **return** $[[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

$\underline{\mathsf{Sign}(sk, \mu):}$
1: $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
2: $(w, c, z) \leftarrow \mathsf{GetTrans}(\mu)$
3: **if** $z = \perp$ **return** $\perp$
4: **return** $\sigma = (w, z)$

$\underline{\mathsf{GetTrans}(\mu):}$
1: $\kappa := 0$
2: **while** $z = \perp$ and $\kappa \leq B$
3: $c \leftarrow U(\mathcal{C})$
4: $(w, z) \leftarrow \mathsf{Sim}(vk, c)$
5: $H := H^{w\|\mu\mapsto c}$
6: $\kappa := \kappa + 1$
7: **return** (w, c, z)

Figure 3.4: Game G_2. The difference from G_1 is highlighted in blue.

random oracle H during the game, then we abort. The value $w^*\|\mu^*$ would be programmed during the game if the adversary has made a sign query with μ^*. As the winning condition in the UF-CMA game already requires a forgery for a message that has not been queried before, the adversary's view is identical to that of the previous one.

It remains to reduce G_3 to UF-NMA security. The signing algorithm does not use the signing key anymore and uses the zero-knowldege simulator to answer the sign queries. The last remaining technicality lies in how to simulate the random oracle. In the ROM, we use the lazy sampling method. At each query to the random oracle, we return a match if there exists any in the database, otherwise we return a fresh sampled element from the range of H and add it in the database.

Strong Unforgeability. For the sUF-CMA security, we modify the above games. Now, the challenger maintains the list $\mathcal{M}$ of message-signature pairs that were queried by the adversary via the signature oracle. Each game, at its final step, also checks

$\underline{\mathcal{C}^{|H'\rangle}(\{w_{i,\kappa}, c_{i,\kappa}, z_{i,\kappa}\}_{i\in[Q_S],\kappa\in[B]}) :}$
1: $\mathcal{M} := \varnothing$
2: $i := 0$
3: $\mathfrak{D} := \varnothing$
4: $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
5: $(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{|H\rangle,\ \mathsf{Sign}(sk,\cdot)}(vk)$
6: Parse $\sigma^* = (w^*, z^*)$
7: $c^* := H_b(w^*\|\mu^*)$
8: **return** $[[\mu^* \notin \mathcal{M}]] \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

$\underline{H(w\|\mu) :}$
1: **if** $\exists c$ such that $(w, \mu, c) \in \mathfrak{D}$
2: **return** c
3: **return** $H'(w\|\mu)$

$\underline{\mathsf{Sign}(sk, \mu) :}$
1: $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
2: $i := i + 1$
3: $\kappa := 0$
4: **while** $z = \perp$ and $\kappa \leq B$
5: $(w, c, z) = (w_{i,\kappa}, c_{i,\kappa}, z_{i,\kappa})$
6: **if** $\exists c'$ such that $(w, \mu, c') \in \mathfrak{D}$
7: $\mathfrak{D} := \mathfrak{D} \setminus (w, \mu, c')$
8: $\mathfrak{D} := \mathfrak{D} \cup (w, \mu, c)$
9: $\kappa := \kappa + 1$
10: **if** $z = \perp$ **return** $\perp$
11: **return** $\sigma = (w, z)$

Figure 3.5: The distinguisher $\mathcal{C}$ for real and simulated transcripts of Σ based on $\mathcal{A}$.

whether the forgery $(\mu^*, (w^*, z^*))$ belongs to this list or not, and if it is it returns 0. With these modifications, everything remains the same up to $\mathsf{Game}\ G_2$. The last two games G_2 and G_3 behave differently only if we have the following conditions: $(\mu^*, (w^*, z^*)) \notin \mathcal{M}$, the random oracle has been reprogrammed on input $w^*\|\mu^*$, and $\mathsf{V}_2(vk, w^*, c^*, z^*) = 1$. The input $w^*\|\mu^*$ has been reprogrammed only if the adversary has made a sign query on μ^*. The probability of w^* appearing in any given loop iteration of the rejection sampling of $\mathsf{Sign}(\mathsf{sk}, \mu^*)$ is bounded by $2^{-\alpha}$. In total, there are at most B iterations per sign query, and the adversary makes at most Q_S queries. By the union bound, the probability that $w^*\|\mu^*$ has been reprogrammed is bounded by $BQ_S \cdot 2^{-\alpha}$. The reduction from G_3 to the $\mathsf{UF\text{-}NMA}$ game works as before.

Runtime. In the ROM, each sign query requires to run the zero-knowledge simulator up to B times. For each hash (resp. sign) query, the reduction performs 1 (resp. up to B) programming operation. It maintains a sorted data structure $\mathfrak{D}$ in order to search and insert in $\mathcal{O}(\log(B \cdot Q_S + Q_H))$ steps. The runtime of the reduction is of order $\mathsf{Time}(\mathcal{A}) + \mathcal{O}(T_{\mathsf{Sim}} \cdot (B \cdot Q_S + Q_H) \cdot \log(B \cdot Q_S + Q_H))$. □

3.1.2 Rényi Divergence Approach: FSwBA Security Analysis

In this section, we start the Rényi divergence-based analysis of Fiat-Shamir with Aborts. This allows us to achieve smaller signature sizes in Section 4.4.5. We note however that the Rényi divergence is useful for *non-aborting* transcripts, meanwhile for *aborting* transcripts the statistical distance analysis is sufficient. To take this into account, we introduce the following definition.

Definition 3.2 (Decomposable Simulator)**.** *Let $p \in [0, 1]$. Let Sim be a zero-knowledge simulator for a Σ-protocol. We say that Sim admits a p-decomposition if there exist two algorithms $\mathsf{Sim}_\perp$ and $\mathsf{Sim}_{\not\perp}$ such that the former only outputs transcripts with $z = \perp$, the latter only outputs transcripts with $z \neq \perp$, and Sim can be defined as in Figure 3.6*

$\underline{\mathsf{Sim}(x):}$
1: **with** probability p
2: $\quad(w,c,z) \leftarrow \mathsf{Sim}_{\perp}(x)$
3: **with** probability $1-p$
4: $\quad(w,c,z) \leftarrow \mathsf{Sim}_{\not\perp}(x)$
5: **return** (w,c,z)

Figure 3.6: Simulator decomposition.

With this formalism, we are able to extend the HVZK definition to the Rényi divergence.

Definition 3.3 (Decomposable Divergence HVZK)**.** *Let $R_{zk} \geq 1, \varepsilon_{zk} > 0, p \in [0,1]$ and $T_{\perp}, T_{\not\perp} \geq 0$. A Σ-protocol is said to be $(\varepsilon_{zk}, T_{\perp}, R_{zk}, T_{\not\perp})$-DDHVZK if there exists a p-decomposable simulator $\mathsf{Sim} = (\mathsf{Sim}_{\perp}, \mathsf{Sim}_{\not\perp})$ such that*

- *algorithm $\mathsf{Sim}_{\perp}$ is $(\varepsilon_{zk}, T_{\perp})$-HVZK (or sc-HVZK) simulator for the Σ-protocol transcript (w', c', z') conditioned on $z' = \perp$,*
- *algorithm $\mathsf{Sim}_{\not\perp}$ has runtime $T_{\not\perp}$, and given x outputs a transcript (w,c,z) such that its distribution and the one of a transcript (w',c',z') of the Σ-protocol conditioned on $z' \neq \perp$ are such that*

$$R_{\infty}\Big((w,c,z) \| (w',c',z')\Big) \leq R_{zk} \ .$$

Note that p can possibly differ from β, but we are interested in the case where their difference is negligible (as in the following theorem). We adapt Theorem 3.1 and its proof to this new setting.

Theorem 3.2. *Let $R_{zk} \geq 1, \varepsilon_{zk}, T_{\perp}, T_{\not\perp} \geq 0, p \in [0,1]$ and H a hash function modeled as a random oracle. If $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is an $(\varepsilon_{zk}, T_{\perp}, R_{zk}, T_{\not\perp})$-DDHVZK public-coin identification protocol with a p-decomposable simulator and probability of aborting β, then we have the following updates on Theorem 3.1. In the ROM, there exists an adversary $\mathcal{B}$ against UF-NMA security of SIG_B with runtime $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\perp}(B-1)Q_S + T_{\not\perp}Q_S)\log(B \cdot Q_S + Q_H))$ such that*

$$\mathsf{Adv}^{\mathsf{X\text{-}CMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq R_{zk}^{Q_S} \cdot (\mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) + (\varepsilon_{zk} + |p-\beta|) \cdot B \cdot Q_S + \Delta_{\mathsf{X}}) + 2^{-\alpha} \cdot B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1) \ .$$

Proof. The proof is almost identical to the one of Theorem 3.1. We replace Game G_2 with three different games $G_{2.1}, G_{2.2}$ and $G_{2.3}$. The other changes between games remain similar. Let $\mathsf{Sim} = (\mathsf{Sim}_{\perp}, \mathsf{Sim}_{\not\perp})$ be the decomposition of the zero-knowledge simulator. We proceed as follows.

Game G_1. It is the same as in the proof of Theorem 3.1.

Game $G_{2.1}$ In this game, we change the signing algorithm. As soon as a transcript (w,c,z) with $z \neq \perp$ is being sampled during the rejection sampling loop, we discard it and replace it with a transcript generated by $\mathsf{Sim}_{\not\perp}$. The multiplicativity of the Rényi divergence implies that

$$\Pr[1 \Leftarrow G_1^{\mathcal{A}}] \leq (1+\varepsilon_{zk})^{Q_S} \cdot \Pr[1 \Leftarrow G_{2.1}^{\mathcal{A}}].$$

Game $G_{2.2}$. We modify the signing algorithm one step further. Let $\mathsf{Bernoulli}(\beta)$ denote the Bernoulli distribution with parameter β (i.e., the probability of sampling 1 is β). We replace the honestly generated transcripts with the following distribution. Sample $b \leftarrow \mathsf{Bernoulli}(\beta)$ and $c \leftarrow U(\mathcal{C})$. If $b = 1$ run $(w, z) \leftarrow \mathsf{Sim}_{\perp}(pk, c)$, and if $b = 0$ run $(w, z) \leftarrow \mathsf{Sim}_{\not\perp}(pk, c)$. Since the transcripts are being sampled independently from each other in both games $G_{2.1}$ and $G_{2.2}$, one can bound the advantage of the distinguisher by $\varepsilon_{zk} \cdot (B-1) \cdot Q_S$.

Game $G_{2.3}$. We replace $\mathsf{Bernoulli}(\beta)$ with $\mathsf{Bernoulli}(p)$. The distinguishing advantage of the adversary between $G_{2.2}$ and $G_{2.3}$ would be less than $|p - \beta| \cdot (B-1) \cdot Q_S$.

The rest of the proof is similar to that of Theorem 3.1. □

3.2 Concrete Analysis of FSwUA: Negative Result

In the two last sections of this chapter, we focus on analyzing formally signatures constructed from combining an identification protocol with the Fiat-Shamir with unbounded aborts paradigm. To the best of our knowledge, this is the first complete analysis of FSwUA.

In this first section, we exhibit a signature constructed using FS_∞ for which the signing runtime is infinite for an instantiation of the hash function H. Therefore, the expected runtime is also infinite and the standard definition of runtime must be changed. We propose minor updates to the signature definitions so that they support such pathological behaviors. Note that FSwUA is the main paradigm used in practice: there is no reason to add a bound for the number of loop iterations in the code if the algorithm never reaches it except with negligible probability, but the latter statement thus needs to be proven.

In Section 3.3, we prove Fiat-Shamir with unbounded aborts does yield signatures which satisfy all correctness, runtime, and security requirements. Correctness of FSwBA is also addressed in Section 3.3 as a corollary of our analysis.

3.2.1 Infinite Signing Runtime in the Worst Case of FSwUA

In this section, we aim to prove the following theorem.

Theorem 3.3. *There exists a parametrization of* dk-LWE$_{m,n,q,Q}$ *such that the following holds assuming the hardness of* dk-LWE$_{m,n,q,Q}$*. There exists a public-coin identification protocol* Σ *with instance generator* IGen *such that, with overwhelming probability over the randomness of* IGen*, there exists a hash function* H_{bad} *such that the signing algorithm of* $\mathsf{SIG}_\infty := \mathsf{FS}_\infty[\Sigma, H_{bad}]$ *on inputs the signing key and any message does not halt.*

The proof relies on constructing the appropriate identification protocol, and then identifying a specific bad instantiation for the hash function. The main idea is to instantiate Lyubashevsky's signature scheme with source distribution Q being the uniform distribution over a ball B and target distribution being the uniform distribution over a corona C, as illustrated in Figure 3.7. For a keypair $\mathbf{A}, \mathbf{S}$, a loop iteration samples $\mathbf{y} \leftarrow U(B)$, defines a commitment $\mathbf{w} \leftarrow \mathbf{A}\mathbf{y} \bmod q$, and returns $\mathbf{y} + \mathbf{S}\mathbf{c}$ with $\mathbf{c} \leftarrow H(\mathbf{A}\mathbf{y} \bmod q \| \mu)$, if and only if $\mathbf{y} + \mathbf{S}\mathbf{c} \in C$.

The cornerstone of our proof is to show that there exists a hash function H_{bad} such that, for every message μ and every $\mathbf{y}$, the challenge $\mathbf{c} = H(\mathbf{A}\mathbf{y} \bmod q \| \mu)$ is such that $\mathbf{y} + \mathbf{S}\mathbf{c} \notin C$. This implies that the signing algorithm of $\mathsf{FS}_\infty[\Sigma, H_{bad}]$ never halts on any input message.

Theorem 3.3. We instantiate Lyubashevsky's signature in the low-density regime. We first construct the identification protocol, and then explain how to instantiate H_{bad} to obtain the result.

We use the following parameters:

- dimensions $n > 0$ and $m = 2n \geq 14$;
- a challenge bound $\tau > 24\sqrt{m}$;
- a good conditioning parameter $d = 300$;
- a crust width $t = d\tau$ and a corona width $t' = d\tau/3 - (d+1)\sqrt{m}$;
- a radius $r = m(t + t')$;
- a prime modulus $q \leq \mathsf{poly}(n)$ that satisfies $q \geq 16(r + \sqrt{m})^4$.

We define the following relation R:

$$R := \left\{ ((\mathbf{A}, \mathbf{AS}), \mathbf{S}) \mid \mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{S} = \frac{2}{3} d\mathbf{I}_m + \mathbf{E} \in \mathbb{Z}^{m \times m}, \sigma_1(\mathbf{E}) \leq \frac{d}{3} \right\},$$

where $\sigma_1(\mathbf{E})$ denotes the largest singular value of $\mathbf{E}$ (when viewed as a real-valued matrix). Note that d is a multiple of 3 so that $\mathbf{S}$ is indeed integral.
Our choice of matrix $\mathbf{S}$ makes it so that $\sigma_1(\mathbf{S}) \leq d$ and $\mathbf{S}$ is full-rank (note that this is a real-valued matrix). We have $\mathbf{S}^{-1} = (2d/3)^{-1} \sum_{k \geq 0} (-(2d/3)^{-1}\mathbf{E})^k$, which satisfies $\sigma_1(\mathbf{S}^{-1}) \leq 3/d$. The matrix $\mathbf{S}$ is the relation witness. We now consider the challenge space $\mathcal{C}$. We set:

$$\mathcal{C} := \{\mathbf{c} \in \mathbb{Z}^m | \|\mathbf{c}\| \leq \tau\}.$$

As $t = d\tau$, we have $t \geq \|\mathbf{S}\mathbf{c}\|$ for all $\mathbf{c} \in \mathcal{C}$ and all $\mathbf{S} \in \mathbb{Z}^{m \times m}$ with $\sigma_1(\mathbf{S}) \leq d$.

We further define the ball B and corona C as follows.

$$B := \mathcal{B}_m(r) \text{ and } C := \mathcal{B}_m(r - t) \setminus \mathcal{B}_m(r - t - t') \ .$$

A graphical representation is given in Figure 3.7.

We instantiate Lyubashevsky's signature scheme as recalled in Section 3.4.1, with the source distribution Q set as the uniform distribution over $\mathbb{Z}^m \cap B$ and the target distributions P set as the uniform distribution over $\mathbb{Z}^m \cap C$. The norm bound check of the verification algorithm is instantiated to $\|\mathbf{z}\| \leq r$, where $\mathbf{z}$ is the vector output by the prover. Finally, the rejection parameter M is set to $M = 100$.

Lemma 3.4. *The identification protocol Σ obtained by instantiating Figure 2.5 as described above is $(1, 1/M)$-correct. Under the* dk-LWE$_{m,n,q,Q}$ *hardness assumption, it is* sc-HVZK.

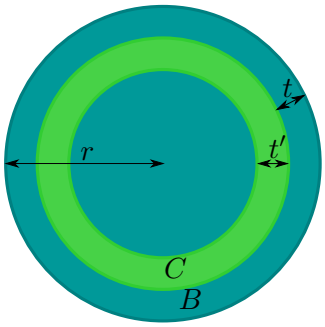

Figure 3.7: The sets B and C in dimension 2.

Proof. We prove each property as follows:
Correctness. The perfect correctness ($\gamma = 1$) follows from the fact that if the prover outputs something, it is by definition a rounding of an element belonging to C and satisfies the relation that the verifier checks. By design, the probability that the verifier outputs some $\mathbf{z} \neq \perp$ is $1/M$.

Zero-Knowledge. We now aim at using Theorem 3.15 to argue the zero-knowledge of the protocol. It suffices to show that for all $\mathbf{c} \in \mathcal{C}$ and all $\mathbf{z} \in \mathbb{Z}^m \cap C$, we have that $P(\mathbf{z}) \leq M \cdot Q(\mathbf{z} - \mathbf{S}c)$.

Note first that for the considered $\mathbf{S}$'s and $\mathbf{c}$'s, if $\mathbf{z}$ belongs to the support of P, then $\mathbf{z} - \mathbf{S}c$ belongs to the support of Q. For such a $\mathbf{z}$, we have:

$$\begin{aligned}
\frac{Q(\mathbf{z} - \mathbf{S}c)}{P(\mathbf{z})} &= \frac{|\mathbb{Z}^m \cap C|}{|\mathbb{Z}^m \cap B|} \\
&\geq \frac{\mathsf{Vol}(\mathcal{B}(r - t - \sqrt{m})) - \mathsf{Vol}(\mathcal{B}(r - t - t' + \sqrt{m}))}{\mathsf{Vol}(\mathcal{B}(r + \sqrt{m}))} \\
&= \left(1 - \frac{t + 2\sqrt{m}}{r + \sqrt{m}}\right)^m - \left(1 - \frac{t + t'}{r + \sqrt{m}}\right)^m .
\end{aligned}$$

By expanding the difference of powers, we then obtain:

$$\begin{aligned}
\frac{Q(\mathbf{z} - \mathbf{S}c)}{P(\mathbf{z})} &= \frac{t' - 2\sqrt{m}}{r + \sqrt{m}} \cdot \sum_{k=0}^{m-1} \left(1 - \frac{t + 2\sqrt{m}}{r + \sqrt{m}}\right)^{m-1-k} \left(1 - \frac{t + t'}{r + \sqrt{m}}\right)^k \\
&\geq \frac{t' - 2\sqrt{m}}{r + \sqrt{m}} \cdot m \cdot \left(1 - \frac{t + t'}{r + \sqrt{m}}\right)^{m-1} \\
&\geq \frac{t' - 2\sqrt{m}}{t + t' + 1} \cdot \left(1 - \frac{1}{2m}\right)^m .
\end{aligned}$$

In the last inequality, we use the fact that $r = m(t + t')$. Now, using the definitions of t and t', we obtain that the latter is $\geq 1/100$.

Let us now consider the probability β that some answer is output by P_2. Note that our choice of t is such that for any $\mathbf{S}$ and challenge $\mathbf{c}$, it holds that

$$C \subseteq B + \mathbf{S}\mathbf{c}.$$

Therefore, the probability that a uniform element from $B + \mathbf{Sc}$ belongs to C is:

$$\begin{aligned}\beta = \frac{\mathsf{Vol}(C)}{\mathsf{Vol}(B)} &= \left(1 - \frac{t}{r}\right)^m - \left(1 - \frac{t+t'}{r}\right)^m \\ &= \left(1 - \frac{t}{r} - 1 + \frac{t+t'}{r}\right) \cdot \sum_{k=0}^{m-1} \left(1 - \frac{t}{r}\right)^{m-1-k} \left(1 - \frac{t+t'}{r}\right)^k \\ &\geq \frac{t'}{r} \cdot m \cdot \left(1 - \frac{t+t'}{r}\right)^{m-1} \\ &\geq \frac{t'}{t+t'} \cdot \left(1 - \frac{1}{m}\right)^m .\end{aligned}$$

For the last inequality, we used the fact that $r = m(t + t')$ and $1 - 1/m < 1$. By using the definitions of t and t', we obtain:

$$\beta \geq \frac{t'}{4(t+t')} \geq \frac{1}{4} \cdot \frac{\tau - 3(1+1/d)\sqrt{m}}{4\tau - 3(1+1/d)\sqrt{m}} \ .$$

We claim that the latter is $\geq 1/20$. Indeed, having this inequality is equivalent to $\tau \geq 12(1 + 1/d)\sqrt{m}$, which is satisfied when $\tau \geq 24\sqrt{m}$. □

We then show that, for any choice of $\mathbf{A}, \mathbf{S}$ such that $((\mathbf{A}, \mathbf{AS}), \mathbf{S}) \in R$, there exists a hash function H such that, using H to instantiate FWsUA, the signing algorithm of $\mathsf{FS}_\infty[\Sigma, H]$ never halts on any input message. That is, for every message μ and every $\mathbf{y}$, the challenge $\mathbf{c} = H(\mathbf{Ay} \bmod q \| \mu)$ is such that $\mathbf{y} + \mathbf{Sc} \notin C$.

Fix the matrices $\mathbf{A}$ and $\mathbf{S}$. We now show how to instantiate the hash function H so that the above holds. A first important observation is that multiplication by $\mathbf{A}$ of a short integer vector is injective. Note that $\mathbf{Ay} = \mathbf{Ay'} \bmod q$ for some $\mathbf{y} \neq \mathbf{y'} \in B$ implies that there exists an integer vector $\mathbf{x} \in \mathbb{Z}^m$ (namely $\mathbf{y} - \mathbf{y'}$) such that $\mathbf{Ax} = \mathbf{0} \bmod q$ and $0 < \|\mathbf{x}\| \leq 2r$. Applying the following lemma with $B = 2r$, it holds that with probability at least $1 - 2^{-\Omega(n)}$ over the random choice of $\mathbf{A}$, such a vector $\mathbf{x}$ does not exist, by our choices of m and q.

Lemma 3.5. *Let $m, n > 0$ and q a prime. Let $B < q$. Then:*

$$\Pr_{\mathbf{A} \in \mathbb{Z}_q^{n \times m}} \left(\lambda_1(\Lambda_q^\perp(\mathbf{A})) < B\right) \leq \mathsf{Vol}(\mathcal{B}_m(1)) \frac{(B + \sqrt{m}/2)^m}{q^n} \ .$$

Proof. The following relations follow from a union bound, the statistical independence of the rows of $\mathbf{A}$ and the fact that every short enough integer vector is non-zero modulo q.

$$\begin{aligned}\Pr_{\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m})}(\lambda_1(\Lambda_q^\perp(\mathbf{A})) < B) &\leq \sum_{\substack{\mathbf{y} \in \mathbb{Z}^m \\ 0 < \|\mathbf{y}\| \leq B}} \Pr_{\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m})}(\mathbf{Ay} = \mathbf{0} \bmod q) \\ &= \sum_{\substack{\mathbf{y} \in \mathbb{Z}^m \\ 0 < \|\mathbf{y}\| \leq B}} \left(\Pr_{\mathbf{a} \hookleftarrow U(\mathbb{Z}_q^m)}(\mathbf{a}^\top \mathbf{y} = \mathbf{0} \bmod q)\right)^n \\ &= \sum_{\substack{\mathbf{y} \in \mathbb{Z}^m \\ 0 < \|\mathbf{y}\| \leq B}} \frac{1}{q^n} \ .\end{aligned}$$

Finally, we note that the volume of the m-dimensional hyperball of center $\mathbf{0}$ and radius $B + \sqrt{m}/2$ is an upper bound on the number of summands. □

As a consequence, we can define H as a function of $\mathbf{y}$ as $\mathbf{A}\mathbf{y}$ uniquely determines $\mathbf{y}$. Based on the protocol, it suffices to find a challenge $\mathbf{c} \in \mathcal{C}$ for each $\mathbf{y} \in \mathbb{Z}^m \cap B$, it holds that $\mathbf{y} + \mathbf{S}\mathbf{c} \notin C$. We then set $H(\mathbf{A}\mathbf{y} \bmod q, \mu)$ to be this $\mathbf{c}$ for all messages μ.

First, note that if $\mathbf{y} \notin C$, then setting $\mathbf{c} := \mathbf{0}$ leads to $\mathbf{y} + \mathbf{S}\mathbf{c}$ being rejected. Thus, we focus on the other case. Let $\Lambda(\mathbf{S})$ be the full-rank lattice generated by the matrix $\mathbf{S}$ (recall that $\mathbf{S}$ is full-rank). Define the scaling $\lambda = t'/\|\mathbf{y}\|$ and note that $\|\lambda\mathbf{y}\| = t'$. Let $\mathbf{x} \in \Lambda(\mathbf{S})$ be such that $\lambda\mathbf{x} \in \mathbf{y} + \mathcal{P}(\mathbf{S})$, where $\mathcal{P}(\mathbf{S}) = \mathbf{S} \cdot [0,1]^n$ denotes the (closed) fundamental parallelepiped spanned by $\mathbf{S}$. In particular there exists a lattice point $\mathbf{e} \in \mathbf{x} + \mathcal{P}(\mathbf{S})$ such that

$$\langle \mathbf{e} - \lambda\mathbf{y}, \lambda\mathbf{y} \rangle \geq 0 \ , \tag{3.1}$$

since otherwise there would exist an affine hyperplane separating $\lambda\mathbf{y} \in \mathbf{x} + \mathcal{P}(\mathbf{S})$ from $\lambda\mathbf{y}$, which would contradict the definition of $\mathbf{x}$. Note that $\|\mathbf{e} - \lambda\mathbf{y}\| \leq d\sqrt{m}$: indeed, when written in the basis $\mathbf{S}$, all of its coordinates belong to $[-1, 1]$, and we have $\sigma_1(\mathbf{S}) \leq d$. Since $\mathbf{e} \in \Lambda(\mathbf{S})$, there exists $\mathbf{k} \in \mathbb{Z}^n$ such that $\mathbf{e} = \mathbf{S}\mathbf{k}$. We set the challenge $\mathbf{c}$ as $\mathbf{k}$. To conclude, we prove the following statements.

$$\|\mathbf{c}\| \leq \tau \quad \text{and} \quad \mathbf{y} + \mathbf{e} \notin C \ .$$

The first one follows from the following (recall that $t' = d\tau/3 - (d+1)\sqrt{m}$):

$$\begin{aligned} \|\mathbf{c}\| = \|\mathbf{S}^{-1}\mathbf{e}\| &\leq \sigma_1(\mathbf{S}^{-1})[\|\mathbf{e} - \lambda\mathbf{y}\| + \|\lambda\mathbf{y}\|] \\ &\leq \frac{3}{d}(d\sqrt{m} + t' + \sqrt{m}) = \tau \ . \end{aligned}$$

By using Equation (3.1), we obtain the following.

$$\begin{aligned} \|\mathbf{y} + \mathbf{e}\|^2 &= \|\lambda\mathbf{y} + \mathbf{y} + (\mathbf{e} - \lambda\mathbf{y})\|^2 \\ &= \|\lambda\mathbf{y}\|^2 + \|\mathbf{y}\|^2 + \|\mathbf{e} - \lambda\mathbf{y}\|^2 \\ &\quad + 2\langle\lambda\mathbf{y}, \mathbf{y}\rangle + 2\langle\lambda\mathbf{y}, \mathbf{e} - \lambda\mathbf{y}\rangle + 2\langle\mathbf{e} - \lambda\mathbf{y}, \mathbf{y}\rangle \\ &\geq \|\lambda\mathbf{y}\|^2 + \|\mathbf{y}\|^2 + 2\langle\lambda\mathbf{y}, \mathbf{y}\rangle \\ &= ((\lambda+1)\|\mathbf{y}\|)^2 \ . \end{aligned}$$

Using the definition of λ and the lower bound on $\|\mathbf{y}\|$, we obtain that

$$\|\mathbf{y} + \mathbf{e}\| \geq (t' + r - t - t') = r - t \ .$$

This completes the proof: instantiated with this hash function, the signing algorithm of the Fiat-Shamir transform of the above Σ-protocol never halts. □

So far, this only exhibits a single bad choice for the hash function, while signatures based on FSwUA support messages of unbounded length. Hence, there are infinitely many possible hash functions (functions with domain $\mathcal{W} \times \{0,1\}^*$ and range $\mathcal{C}$, with $\mathcal{W}$ being the commitment space). As a consequence, it is not immediate that a single bad hash function implies an infinite expected runtime for the signature scheme in the ROM, and one could think that simply considering the runtime when H is a random oracle could be sufficient to fix it.

Corollary 3.6. *We have* $\Pr_H[\forall w \in \mathcal{W}, H(w\|\mu) = H_{bad}(w\|\mu)] \geq |\mathcal{C}|^{-|\mathcal{W}|}$ *for any message* μ*. Therefore, the expected runtime of* $\mathsf{Sign}(\mathsf{sk}, \mu)$ *over the choice of the random oracle* H *is infinite.*

Our result relies on the hardness of the dk-LWE problem when the weight vector is sampled from the uniform distribution over a hyperball. This is an unusual distribution for dk-LWE. However, it can be checked that for appropriate parameters, the proof of [BLR+18, Section 5] that decision LWE is hard for a noise distribution that is uniform in a hypercube carries over to the hyperball setting.

3.2.2 Updated Signature Definition

As shown in Section 3.2.1, there are instances of identification protocols that yield signature schemes with infinite expected runtime of the signing algorithm. This requires relaxing the runtime requirement in the definition to be expected polynomial time with overwhelming probability over the choice of the hash function. Yet, there is another subtlety doing so: in the security game, an adversary might make a sign query that never halts. In the case of the above construction, the challenger, which is unbounded, can still notice it as the commitment space is bounded and the rejection step is deterministic. Once all the potential commitments have failed to produce a valid signature, the challenger knows that it cannot answer the query. This is however not the case of every signature scheme. To take such event into account, we consider that an attacker automatically wins if the challenger takes more than T' time to answer a signature query, for some parameter T'. An alternative choice could be to consider that an adversary which makes a non-terminating sign query loses, since the challenger does not answer anymore. We prefer to add this parameter T' as this makes the definition stronger by further guaranteeing that an adversary cannot find a query which forces the signer to run for a long time, which could be desirable in practice as well.

We now state our updated definition for signatures. It is highly similar to the standard Definition 2.7 and we only highlight the differences.

Definition 3.4 (Modified Digital Signature in the ROM)**.** *Let* H *be a random oracle to which all algorithms have oracle access. A signature scheme is a tuple* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *of algorithms with the following specifications. Everything is as in Definition 2.7, except for the runtime of* Sign*, which we define below, and a minor tweak in the security game.*

- $\mathsf{Sign}^H : (\mathsf{sk}, \mu) \to \sigma$ *is a probabilistic algorithm that takes as inputs a signing key* sk *and a message* $\mu \in \mathcal{M}$ *and outputs a signature* σ*. We denote with* $T_{\mathsf{Sign}^H(\mathsf{sk},\mu)}$ *the runtime of* $\mathsf{Sign}(\mathsf{sk}, \mu)$*.*

Let $\gamma > 0, T = \mathsf{poly}(\lambda)$ *and* $\varepsilon = \mathsf{negl}(\lambda)$*. We say that the signature scheme is* γ*-correct if for any pair* $(\mathsf{vk}, \mathsf{sk})$ *in the range of* KeyGen *and* μ*,*

$$\Pr[\mathsf{Verify}(\mathsf{vk}, \mu, \mathsf{Sign}(\mathsf{sk}, \mu)) = 1 \mid \mathsf{Sign}(\mathsf{sk}, \mu) \textit{ halts}] \geq \gamma,$$

and we say that it is (T, ε)*-efficient if for any pair* $(\mathsf{vk}, \mathsf{sk})$ *in the range of* KeyGen *and* μ*,*

$$\Pr_H[T_{\mathsf{Sign}^H(\mathsf{sk},\mu)} > T] < \varepsilon.$$

where both probabilities are taken over the random coins of the two algorithms and the random oracle.

In addition, we update the security game as follows. Let T' be another function of λ. We define T'-UF-CMA security exactly as UF-CMA *security in Definition 2.8, except that we further make the adversary win as soon as it makes a sign query for which the signing algorithm takes more than T' steps to halt.*

Definition 3.4 does not forbid the situation described in Subsection 3.2.1 from occurring but guarantees that it should be hard to find non-halting queries.

3.3 Concrete Analysis of FSwUA: Positive Results

Equipped with this updated definition, we prove that signatures constructed from applying FSwUA to an identification protocol yields a signature scheme that satisfies all three *correctness*, *runtime*, and *security* requirements. This result extends to prove that FSwBA signatures satisfy *correctness*.

Theorem 3.7 (Runtime)**.** *Let $\gamma > 0, \beta \in (0,1)$ and H a hash function modeled as a random oracle. Let $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ be an identification protocol that is (γ, β)-correct and has commitment min-entropy α. Let $\mathsf{SIG}_\infty = \mathsf{FS}_\infty[\Sigma, H]$. Let $\mathcal{M}$ be the message space and $I_{\mathsf{Sign}^H}(sk, \mu)$ denote the random variable counting the number of iterations of the signing algorithm on input (sk, μ) using a random oracle H where $\mu \in \mathcal{M}$. It holds that for any $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$, any message $\mu \in \mathcal{M}$, and any integer i:*

$$\Pr_H(I_{\mathsf{Sign}^H}(\mathsf{sk}, \mu) > i) \leq \beta^i + \frac{2^{-\alpha}}{(1-\beta)^3}.$$

Proof. Let us start by introducing the random variables $(w_i, c_i, z_i, \mathtt{acc}_i)_{i \geq 1}$. It denotes an infinite sequence of transcripts, where $\mathtt{acc}_i$ is the random variable denoting whether the transcript is accepted or not. It takes value in $\{0,1\}$, where 0 denotes rejection and 1 acceptance. For the sake of the proof, let the sequence continue regardless of whether a prior transcript was accepted or not. Let $N = I_{\mathsf{Sign}^H(\mathsf{sk},\mu)}$. It denotes the index of the first accepting transcript, i.e., $N = \mathrm{argmin}_i(\{\mathtt{acc}_i = 1\})$. Let us denote by M the index of the first collision, i.e., $M = \min\{i | \exists j < i, w_j = w_i\}$. Note that once H is fixed, a transcript is a deterministic function of w_i.

Let $i \geq 1$. Let us decompose:

$$\begin{aligned}\Pr_H(N > i) &= \Pr_H(N < M) \cdot \Pr_H(N > i | N < M) \\ &\quad + \Pr_H(N \geq M) \cdot \Pr_H(N > i | N \geq M) \\ &\leq 1 \cdot \Pr_H(N > i | N < M) + \Pr_H(N \geq M) \cdot 1.\end{aligned}$$

We now focus on studying each of these probabilities. The second one can be rewritten as

$$\Pr_H(N \geq M) = \sum_{k=2}^{\infty} \Pr_H(M = k) \cdot \Pr_H(N \geq M | M = k).$$

Let us first focus on $\Pr_H(M = k)$. The random variable M only depends on the w_i's, which are i.i.d.: we can bound the collision probability with their min-entropy: $\Pr_H(M = k) \leq k^2 \cdot 2^{-\alpha - 1}$. Next, as long as no collision occurred, all c_i's

can be seen as "fresh" randomness, i.e., all c_i's are uniform over the challenge space and most importantly, they are independent. Hence conditioned on $M = k$, we know that the probability of rejecting the first $k-1$ samples is β^{k-1}. Then

$$\begin{aligned}\Pr_H(N \geq M) &\leq \sum_{k=2}^{\infty} k^2 \cdot 2^{-\alpha-1} \cdot \beta^{k-1} = 2^{-\alpha-1} \cdot \frac{\beta + 1 - (1-\beta)^3}{(1-\beta)^3} \\ &\leq 2^{-\alpha} \cdot \frac{1}{(1-\beta)^3} \ ,\end{aligned}$$

where the equality comes from the fact that $\sum_{k\geq 1} k^2 \cdot \beta^{k-1} = (\beta+1)/(1-\beta)^3$. Now, as we previously stated, conditioned on $N < M$, the distribution of N is geometric with parameter $1-\beta$. Hence, we have $\Pr_H(N > i | N < M) = \beta^i$. Plugging everything together, we obtain

$$\Pr_H(N > i) \leq \beta^i + \frac{2^{-\alpha}}{(1-\beta)^3} \ .$$

□

Assume that $\alpha = \omega(\log(\lambda))$. Setting $i = \omega(\log(\lambda)/\log(1/\beta))$ ensures that with overwhelming probability over the choice of H, signing runs in polynomial time. We note that this bound does not contradict the previous (negative) result. Indeed, it does not imply any statement on the finiteness of the expected value of T_{Sign^H}, which is infinite in the previous section.

We move on to checking that FSwUA satisfies the new γ-correctness property, assuming that the underlying identification protocol is (γ, β)-correct.

Theorem 3.8. *Let $\gamma > 0, \beta \in (0,1)$ and let H denote a hash function modeled as a random oracle. Let $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ be an identification protocol that is (γ, β)-correct. Let T denote the runtime of one interaction in the worst-case. Let $\alpha > 0$ be its commitment min-entropy. Let $\mathsf{SIG}_\infty = \mathsf{FS}_\infty[\Sigma, H]$. Then for any $i = \omega(\log(\lambda)/\log(1/\beta))$, it is γ-correct as well as $(iT, \beta^i + 2^{-\alpha}/(1-\beta)^3)$-efficient.*

Proof. Let $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}$ and $\mu \in \mathcal{M}$. Conditioned on $\mathsf{Sign}(\mathsf{sk}, \mu)$ halting, the output transcript follows the same distribution as a transcript from the identification protocol conditioned on not being $\perp$. In particular, the challenge is uniform over $\mathcal{C}$, as it is a hash that comes from the random oracle. Only its marginal distribution is important here, as well as the fact that it is independent from the first and last message of the prover. Hence, this transcript is accepted with probability γ over the random coins of Sign and the random oracle. □

With FSwBA, the problem is reversed: bounding the runtime becomes easy, whereas proving the correctness becomes mildly more tedious, as one needs to check that $\perp$ is not output too often.

Theorem 3.9. *Let $\gamma > 0, \beta \in (0,1)$ and $B > 0$. Let H be a hash function modeled a random oracle. Let $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ be an identification protocol that is (γ, β)-correct and has commitment min-entropy α. Let $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$. Then, for any $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and any message $\mu \in \mathcal{M}$, we have*

$$\Pr[\mathsf{Verify}(\mathsf{vk}, \mu, \mathsf{Sign}(\mathsf{sk}, \mu)) = 1] \geq \gamma \cdot \left(1 - \beta^B - \frac{2^{-\alpha}}{(1-\beta)^3}\right),$$

where the randomness is taken over H as well as the coins of Sign.

Proof. The result follows from Theorem 3.7. Indeed, assuming that Sign did not output $\perp$, then the final challenge that it outputs is uniform over the challenge space $\mathcal{C}$. It may not be independent from previous executions of the identification protocol, but nonetheless its marginal distribution is uniform over $\mathcal{C}$. Hence, assuming that Sign did not output $\perp$, it outputs a signature that is accepted by Verify with probability at least γ, by correctness of the identification protocol. In the case where Sign outputs $\perp$, this signature is of course rejected by Verify. Hence, by the law of total probabilities we have

$$\Pr[\mathsf{Verify}(\mathsf{vk}, \mu, \mathsf{Sign}(\mathsf{sk}, \mu)) = 1] \geq \gamma \cdot \left(1 - \beta^B - \frac{2^{-\alpha}}{(1-\beta)^3}\right).$$

□

We finally prove the security of the unbounded version of the Fiat-Shamir transform in the ROM. We reduce the T'-$\mathsf{UF\text{-}CMA}$ security of the unbounded signature scheme to the $\mathsf{UF\text{-}CMA}$ security of the bounded one in the ROM.

Theorem 3.10. *Let $\alpha \geq 0, \beta \in (0,1)$, and let H be a hash function modeled as a random oracle. Assume that $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is a (γ, β)-correct identification protocol, and that the commitment message of P_1 has min-entropy α. Let T denote the runtime of one iteration of the protocol with the hash function. Let $T' > BT$. For any arbitrary adversary $\mathcal{A}$ against T'-$\mathsf{UF\text{-}CMA}$ security of $\mathsf{SIG}_\infty = \mathsf{FS}_\infty[\Sigma, H]$ that issues at most Q_H queries to the random oracle H and Q_S classical queries to the signing oracle and for any fixed integer B, the same adversary $\mathcal{A}$ against $\mathsf{UF\text{-}CMA}$ security of $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ is such that $|\mathsf{Adv}^{T'\text{-}\mathsf{UF\text{-}CMA}}_{\mathsf{SIG}_\infty}(\mathcal{A}) - \mathsf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathsf{SIG}_B}(\mathcal{A})|$ is bounded as*

$$Q_S \cdot \beta^B + \frac{\beta^B \cdot 2^{-\alpha}}{(1-\beta)^3} + 2^{-\alpha} \cdot B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1).$$

This also holds replacing $\mathsf{UF\text{-}CMA}$ *with* $\mathsf{UF\text{-}CMA}_1$ *or* $\mathsf{sUF\text{-}CMA}$ *security.*

Proof. We proceed with three hybrid games.

$\mathsf{Game}\ G_0$. We define $\mathsf{Game}\ G_0$ as the $\mathsf{UF\text{-}CMA}$ security of SIG_B.

$\mathsf{Game}\ G_1$. Let $\mathsf{Game}\ G_1$ be game T'-$\mathsf{UF\text{-}CMA}$ in which the adversary is promised to not make any sign query that takes more than T' steps to halt. In the ROM, if the advantage of the adversary $\mathcal{A}$ to distinguish these games is non-zero, then $\mathcal{A}$ must have queried a message μ such that $\mathsf{Sign}(sk, \mu) = \perp$ in Game G_0. Note that we cannot assume $\mathcal{A}$ is a purified quantum circuit since the queries to the signing oracle must be classical and cannot be purified. Nevertheless, we can purify $\mathcal{A}$ between the sign queries (the random oracle queries are quantum and would cause no problem for purification). This is equivalent to saying that after the i-th sign query μ_i, and receiving σ_i as the outcome, the adversary applies U_i, where U_i comes from a distribution derived from $\{\sigma_j\}_{j \leq i}$, and then measures one of its registers to obtain μ_{i+1}. It repeats this process Q_S times. By doing so, we can prove the above statement. As long as $\mathsf{Sign}(sk, \mu_i) \neq \perp$, the distributions of σ_i and thus U_i are

identical. It follows that the mixed state of the adversary remains identical in both games.

Let $\mathcal{R}^{G_0,\mathcal{A}}$ be an algorithm that runs G_0 with $\mathcal{A}$ as a subroutine, records the sign queries of $\mathcal{A}$, and wins if one of them is answered by $\perp$. We have

$$\left|\Pr[1 \Leftarrow G_1^{\mathcal{A}}] - \Pr[1 \Leftarrow G_0^{\mathcal{A}}]\right| \leq \Pr[\mathsf{win}(\mathcal{R}^{G_0,\mathcal{A}})].$$

We aim at bounding the winning probability of $\mathcal{R}$. Remember G_1 from Figure 3.2, which we rename G_0' in this proof. In Theorem 3.1, we proved that

$$\left|\Pr[1 \Leftarrow G_0^{\mathcal{A}}] - \Pr[1 \Leftarrow G_0'^{\mathcal{A}}]\right| \leq 2^{-\alpha} \cdot B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1),$$

in the ROM. It follows that we can replace Game G_0 in $\Pr[\mathsf{win}(\mathcal{R}^{G_0,\mathcal{A}})]$ with G_0' and only lose the above terms in their corresponding random oracle models.

Finally, using the union bound and the β-correctness of the identification protocol, the winning probability of the algorithm $\mathcal{R}$ relative to G_0' is bounded by $Q_S \cdot \beta^B$.

Game G_2. This is the genuine T'-UF-CMA game. The distinguishing advantage of $\mathcal{A}$ is bounded by the probability that $\mathcal{A}$ makes a sign query that takes more than T' steps to halt. Theorem 3.7 implies that this probability is bounded by $\beta^{T'/T} + 2^{-\alpha}/(1-\beta)^3 \geq \beta^B + 2^{-\alpha}/(1-\beta)^3$. This completes the proof. □

3.4 Application to Lyubashevsky's Σ-Protocol

While we consider in this chapter generic Σ-protocols, our central application of the Fiat-Shamir with aborts paradigm is Lyubashevsky's signature scheme [Lyu09, Lyu12]. We show here that the underlying Σ-protocol satisfies the zero-knowledge property of Definition 3.1, i.e., admits an efficient simulator for all transcripts including the aborting ones.

3.4.1 A Simulator for Lyubashevsky's Σ-protocol

We consider the simulator Sim described in Figure 3.8 for the underlying Σ-protocol of Section 2.6. For the rest of this section, we drop the runtime considerations, and note that the simulator runs in roughly the time necessary to sample from P and compute two matrix multiplications in the non-aborting case or the time necessary to sample a uniform $\mathbf{w}$ in the aborting case.

$\underline{\mathsf{Sim}(\mathbf{T}, \mathbf{c}) :}$
with probability $1/M$
1: $\mathbf{z} \leftarrow P$
2: $\mathbf{w} := \mathbf{Az} - \mathbf{Tc}$
else
3: $\mathbf{w} \leftarrow U(\mathbb{Z}_q^n)$
4: $\mathbf{z} := \perp$
5: **return** $(\mathbf{w}, \mathbf{z})$

Figure 3.8: Simulator Sim of Lyubashevsky's Σ-protocol.

The proof that the simulation is correct in the non-aborting case is quite standard and derives from the rejection sampling. For the aborting case, our proof relies on the

leftover hash lemma and requires the source distribution Q to have high min-entropy. The case of low min-entropy source distributions Q is handled later on.

3.4.1.1 High Min-Entropy Source Distributions

We first consider the case where Q has high min-entropy. In that case, we obtain statistical zero-knowledge as per Definition 3.1.

Theorem 3.11. *Let $m \geq n$, $k > 0$, q prime, $\varepsilon, \beta_{\mathsf{SIS}} > 0$ and $\eta \in [0, \frac{1}{2}]$. Assume that*

$$H_\infty(Q) \geq n \log q + \log\left(1 - \frac{1-\eta}{M}\right) + 2\log\frac{1}{\varepsilon}.$$

Let $(\mathbf{S}, \mathbf{T}) \in R_{m,n,k,q,\beta_{\mathsf{SIS}}}(\mathbf{A})$ for some $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n\times m})$. Assume that

$$\forall \mathbf{c} \in \mathcal{C} : R_\infty^\varepsilon(P \| Q_{+\mathbf{Sc}}) \leq M.$$

Then Lyubashevsky's Σ-protocol satisfies the $\varepsilon + \eta(1 + 1/M)$-$\mathsf{HVZK}$ property.

Observe that $\mathbf{c} \hookleftarrow U(\mathcal{C})$ in both genuine and simulated transcripts. It hence suffices to study the distribution of the rest of the transcript conditioned on the value of $\mathbf{c}$.

The first part of the following result derives from Lemma 2.7, and the second part derives from the description of Sim. The claim ensures that the probabilities of the event $\mathbf{z} = \perp$ in the genuine and simulated transcripts are close-by.

Lemma 3.12. *For all $\mathbf{c}$ output by V_1, the probability (over the random coins of P_1 and P_2) that P_2 outputs $\perp$ belongs to $[1-1/M, 1-(1-\eta)/M]$. For all $\mathbf{c}$, the probability (over its random coins) that the last component of Sim is equal to $\perp$ is $1 - 1/M$.*

We now consider the transcript distribution conditioned on the event $\mathbf{z} \neq \perp$.

Lemma 3.13. *Conditioned on $\mathbf{z} \neq \perp$, the distribution of the transcript $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ generated by (P, V) is within statistical distance η from the simulated distribution.*

Proof. For all $\mathbf{c}$ and conditioned on $\mathbf{z} \neq \perp$, the distribution of $\mathbf{z}$ output by P_2 is within statistical distance η from P (see Lemma 2.7). The latter is exactly the distribution of $\mathbf{z}$ conditioned on $\mathbf{z} \neq \perp$.

To complete the proof of Lemma 3.13, we argue that when $\mathbf{z} \neq \perp$, the first coefficient of the triple is fully determined by the two others, and equal to $\mathbf{A}z - \mathbf{T}c$ in both transcript and simulation. □

Finally, we consider the statistical distance of the distributions conditioned on $\mathbf{z} = \perp$.

Lemma 3.14. *Conditioned on $\mathbf{z} = \perp$, the distribution of the transcript $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ generated by (P, V) is within statistical distance ε from the simulated distribution.*

Proof. It suffices to prove that for all $\mathbf{c}$ and conditioned on $\mathbf{z} = \perp$, the distribution of $\mathbf{w}$ in the transcript generated by (P, V) is statistically close to uniform over $\mathbb{Z}_q^n$. Thanks to the first claim above, we have:

$$\begin{aligned} H_\infty[\mathbf{y} | \mathbf{c} \wedge \mathbf{z} = \perp] &\geq H_\infty[\mathbf{y}] - \log \Pr[\mathbf{z} = \perp | \mathbf{c}] \\ &\geq H_\infty[\mathbf{y}] - \log\left(1 - \frac{1-\eta}{M}\right). \end{aligned}$$

We conclude by using the leftover hash lemma (Lemma 2.1). □

Theorem 3.11 follows from the above lemmas by term collection.

3.4.1.2 Low Min-Entropy Source Distributions

The above handles many settings of Lyubashevsky's signature, as the source distribution Q is often chosen to have high min-entropy so that the map $\mathbf{y} \mapsto \mathbf{Ay} \bmod q$ is (very) surjective. In some cases, however, it is chosen of lower entropy and the map $\mathbf{y} \mapsto \mathbf{Ay} \bmod q$ is very far from surjective. For example, this allows to avoid the forking lemma in the security proof [AFLT16], which both leads to a tight security proof and facilitates unforgeability proofs in the QROM. Our pathological construction from Section 3.2.1 also relies on this regime.

We explain how this can be handled, for some distributions. First, we consider computational zero-knowledge rather than statistical zero-knowledge. As one needs to be able to replace real transcripts of (many) sign queries by simulated ones in the security proof, we consider a strong notion of computational zero-knowledge: computational indistinguishability is required to hold even when the distinguisher is given the witness (of course, the simulator does not use the witness). This definition is compatible with our Fiat-Shamir with aborts analyses. In the analysis based on adaptive reprogramming (Section 3.1.1), transcripts can be replaced one at a time by simulated ones using a hybrid argument, since the witness allows to generate real signatures. In particular, our definition implies the notion of computational HVZK for multiple transcripts used in [GHHM21, Definition 2], which they use to argue that all transcripts can be replaced by simulated ones in a single step. In our reduction, when using the zero-knowledge property, the witness x is available to the challenger.

Definition 3.5 (Strong Computational HVZK)**.** *Let $\varepsilon_{zk}, T \geq 0$ with ε_{zk} a negligible function of the security parameter. A Σ-protocol $((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ for a relation R is (ε_{zk}, T)-sc-HVZK if there exists a simulator Sim with runtime at most T such that for all polynomial-time algorithm $\mathcal{A}$ and all $(x, y) \in R$, the following is $\leq \varepsilon_{zk}$:*

$$\mathrm{Adv}(\mathcal{A}) = \Bigg| \Pr\left[\mathcal{A}((w,c,z),y) = 1 \;\middle|\; \begin{array}{c} (w, st) \leftarrow \mathsf{P}_1(x,y), \\ c \leftarrow \mathsf{V}_1(x,w), \\ z \leftarrow \mathsf{P}_2(x,y,c,w,st) \end{array} \right] - \Pr\left[\mathcal{A}((w,c,z),y) = 1 \,\middle|\, (w,c,z) \leftarrow \mathsf{Sim}(x) \right] \Bigg|.$$

One may consider classical or quantum adversaries $\mathcal{A}$.

As in the statistical case, if the Σ-protocol is public-coin, then without loss of generality, the challenge c can be sampled uniformly from the challenge space $\mathcal{C}$ and passed over as input to the simulator Sim. In the following, we use this formalism.

The computational assumption that we rely on is the Learning With Errors problem [Reg09]. We use its knapsack form, introduced in [MM11].

Definition 3.6 (k-LWE)**.** *Let $m \geq n \geq 1$, $q \geq 2$ and D a distribution over $\mathbb{Z}_q^m$. The search knapsack-LWE problem* sk-LWE$_{m,n,q,D}$ *with parameters m, n, q, D consists in recovering $\mathbf{e}$ from $(\mathbf{A}, \mathbf{Ae})$, where $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n \times m})$ and $\mathbf{e} \leftarrow D$. The decision*

knapsack-LWE problem dk-LWE$_{m,n,q,D}$ *with parameters* m, n, q, D *consists in distinguishing between the distributions* $(\mathbf{A}, \mathbf{Ae})$ *and* $(\mathbf{A}, \mathbf{u})$*, where* $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n\times m})$*,* $\mathbf{e} \leftarrow D$ *and* $\mathbf{u} \leftarrow U(\mathbb{Z}_q^n)$*.*

We argue that for some distributions Q, it is possible to prove computational zero-knowledge in the sense of Definition 3.5, with the above simulator (Figure 3.8).

Theorem 3.15. *Let* $m \geq n$ *and* $k > 0$*,* $q \leq \mathsf{poly}(m, n)$ *prime and* $\beta_{\mathsf{SIS}} > 0$*. Let* Q *be such that the* dk-LWE$_{m,n,q,Q}$ *problem is hard. Let* $(\mathbf{S}, \mathbf{T}) \in R_{m,n,k,q,\beta_{\mathsf{SIS}}}(\mathbf{A})$ *for some* $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n\times m})$*. Assume that*

$$\forall \mathbf{c} \in \mathcal{C} : \Pr_{\mathbf{z}\leftarrow P}\left[P(\mathbf{z}) \leq M \cdot Q(\mathbf{z} - \mathbf{S}c)\right] \geq 1 - \eta,$$

where $1 + 1/\mathsf{poly}(m, n) \leq M \leq \mathsf{poly}(m, n)$ *and* $\eta \geq 0$ *is negligible.*

Then the distribution of the transcript $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ *generated by* $\langle \mathsf{P}(\mathbf{S}), \mathsf{V}(\mathbf{T})\rangle$ *is computationally indistinguishable from the distribution of the triple* $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ *obtained by sampling* $\mathbf{c} \hookleftarrow U(\mathcal{C})$ *and* $(\mathbf{w}, \mathbf{z}) \leftarrow \mathsf{Sim}(\mathbf{T}, \mathbf{c})$*, even if the distinguisher is given* $\mathbf{S}$*.*

The first two claims (Lemmas 3.12 and 3.13) of the proof of Theorem 3.11 still hold. It hence suffices to prove the statistical indistinguishability of the genuine and simulated transcripts $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ conditioned on $\mathbf{z} = \perp$.

We first show that the genuine distribution of $\mathbf{y}$ conditioned on $\mathbf{z}$ being rejected resembles the distribution Q of $\mathbf{y}$.

Lemma 3.16. *Assume that* $M > 1$*. Consider the execution* $\langle \mathsf{P}(\mathbf{S}), \mathsf{V}(\mathbf{T})\rangle$*. Let* $Q^\perp$ *denote the distribution of* $\mathbf{y}$ *conditioned on* $\mathbf{z} = \perp$*. Then we have:*

$$R_\infty(Q^\perp \| Q) \leq \frac{M}{M-1}.$$

Proof. For all $\mathbf{y}$, we have

$$Q^\perp(\mathbf{y}) = \frac{\Pr[\mathbf{y} \wedge \mathbf{z} = \perp]}{\Pr[\mathbf{z} = \perp]} \leq \frac{Q(\mathbf{y})}{\Pr[\mathbf{z} = \perp]}.$$

Lemma 3.12 ensures that the denominator is at least $1 - 1/M$. □

The following result states that if $(\mathbf{A}, \mathbf{A}y)$ is pseudo-random for $\mathbf{y} \leftarrow D$, then so is it for $\mathbf{y} \leftarrow D'$ for any distribution D' such that $R_\infty(Q'\|Q)$ is polynomially bounded.

Lemma 3.17. *Let* $m \geq n \geq 1$*. Let* $q \leq \mathsf{poly}(m, n)$ *prime. Let* D *and* D' *be two distributions over* $\mathbb{Z}^m$ *such that* $R_\infty(D'\|D) \leq \mathsf{poly}(m, n)$*. Then* dk-LWE$_{m,n,q,D}$ *reduces to* dk-LWE$_{m,n,q,D'}$*.*

Proof. Note first that dk-LWE$_{m,n,q,D}$ reduces to sk-LWE$_{m,n,q,D}$. Moreover as we have $R_\infty(D'\|D) \leq \mathsf{poly}(m, n)$, the probability preservation property (see Lemma 2.2) implies that sk-LWE$_{m,n,q,D}$ reduces to sk-LWE$_{m,n,q,D'}$. Finally, as shown by [MM11, Theorem 3.1], sk-LWE$_{m,n,q,D'}$ reduces to dk-LWE$_{m,n,q,D'}$. The composition of these reductions leads to the above claim. □

Theorem 3.15 now follows from combining Lemmas 3.16, 3.17, 3.12 and 3.13.

3.4.1.3 Rényi Divergence Approach: DDHVZK Property

As a conclusion to Section 3.1.2, we prove the following statement.

Theorem 3.18. *Let $m \geq n$, $k > 0$, q prime, $\eta, \beta_{\mathsf{SIS}} > 0$ and $\varepsilon \in [0, \frac{1}{2}]$. Assume that*

$$H_\infty(Q) \geq n \log q + \log\left(1 - \frac{1-\varepsilon}{M}\right) + 2\log\frac{1}{\eta}.$$

Let $(\mathbf{S}, \mathbf{T}) \in R_{m,n,k,q,\beta_{\mathsf{SIS}}}(\mathbf{A})$ for some $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n\times m})$. Assume that

$$\forall \mathbf{c} \in \mathcal{C} : R_\infty^\varepsilon(P \| Q_{+\mathbf{Sc}}) \leq M.$$

Then Lyubashevsky's Σ-protocol satisfies the $(\eta, 1/(1-\varepsilon))$-DDHVZK property.

Proof. We use the same simulator from Figure 3.8 and we note that by construction, it is decomposable. The analysis in the aborting case is already done in Lemma 3.14, while we change the proof of Lemma 3.13, where we use the Rényi divergence bounds from Lemma 2.7 in the non-aborting case. □

In conclusion, while the two approaches lead to similar bounds, we note that the reductions are stateful when $\varepsilon = \mathsf{negl}(\lambda)$ in the statistical distance analysis, while the Rényi divergence approach allows for $\varepsilon = O(1/Q_s)$.

Chapter 4

Optimized Use of Rejection Sampling in Fiat-Shamir with Aborts

This chapter takes a theoretical approach on the use of rejection sampling in Lyubashevsky's and BLISS signatures. Namely, we study the expected number of iterations and show that it is optimal in the "perfect" setting (Section 4.2). We then look for a choice of source and target distributions minimizing the expected Euclidean norm of the signature. To do so, we first give lower bounds on the compactness of the signature for a fixed target number of rejections (Section 4.3). Finally, we compare three choices of distributions: discrete gaussians, hypercube-uniform and hyperball-uniform, both in theory and for practical parameters (Section 4.4). Further results, in particular a bounded rejection technique, can be found in [DFPS22].

4.1 Preliminaries: Beta Function and Hyperspherical Cap

The beta function is a special function related to the gamma function. Its link with hyperballs is necessary when considering imperfect rejection sampling, as seen in Lemma 4.10.

Definition 4.1 (Regularized Incomplete Beta Function)**.** *The incomplete beta function is defined over* $[0,1] \times \mathbb{R}^+ \times \mathbb{R}^+$ *as*

$$B : (x; a, b) \mapsto \int_0^x t^{a-1}(1-t)^{b-1}\,\mathrm{d}t.$$

When $x = 1$*, this is the Beta function, and we use the notation* $B(a, b)$ *in that case. For fixed* a, b*, the function* $I.(a, b) : x \mapsto B(x; a, b)/B(a, b)$ *is invertible.*

The function $x \mapsto I_x(a, b)$ and its inverse are useful when we consider the area of a hyperspherical cap, as defined below.

Lemma 4.1 (Hyperspherical Cap)**.** *Let $n > 1$, $\eta > 1$ and $\mathbf{x} \in \mathcal{S}_n(1)$. Then the set $\{\mathbf{y} \in \mathcal{B}_n(1) | \langle \mathbf{y}, \mathbf{x} \rangle \geq 1/\eta\}$ is the intersection of a half-space and the unit hyperball. It is called a hyperspherical cap and has volume V_η and area A_η satisfying*

$$V_\eta = \frac{V_n(1)}{2} \cdot I_{1-\frac{1}{\eta^2}}\left(\frac{n+1}{2}, \frac{1}{2}\right) \quad \text{and} \quad A_\eta = \frac{S_n}{2} \cdot I_{1-\frac{1}{\eta^2}}\left(\frac{n-1}{2}, \frac{1}{2}\right).$$

From [MV10, Lemma 4.1], by placing an appropriate cone in the hyperspherical cap:

$$I_{1-\frac{1}{\eta^2}}\left(\frac{n+1}{2}, \frac{1}{2}\right) > \left(1 - \frac{1}{\eta^2}\right)^{n-\frac{1}{2}} \cdot \frac{1-\frac{1}{\eta}}{n}.$$

By placing the hyperspherical cap into a cylinder of 1-dimensional height $1 - 1/\eta$:

$$I_{1-\frac{1}{\eta^2}}\left(\frac{n+1}{2}, \frac{1}{2}\right) < \left(1 - \frac{1}{\eta^2}\right)^{n-1} \cdot n \cdot \left(1 - \frac{1}{\eta}\right).$$

Letting ε denote $I_{1-1/\eta^2}(\frac{n+1}{2}, \frac{1}{2})$, we obtain the following consequence of the above two inequalities, which we use to estimate the smooth Rényi divergence between uniform distributions in hyperballs:

$$1 - (2n\varepsilon)^{\frac{1}{n+1/2}} < \frac{1}{\eta^2} < 1 - \left(\frac{\varepsilon}{n}\right)^{\frac{1}{n-1}}.$$

For $\varepsilon = 2^{-c \cdot n}$ for a constant $c > 0$, we obtain that $1/\eta^2$ tends to $1 - 2^{-c}$ when n goes to infinity. For ε satisfying $\varepsilon \geq 2^{-o(n)}$ and $\varepsilon = o(1/n)$ with n going to infinity, we obtain that $1/\eta^2 \sim -\ln(\varepsilon)/n$.

4.2 Optimality of Generic Rejection Sampling

This section focuses on rejection sampling itself, before instantiating it in the case of Lyubashevsky's signature. We show the optimality of its expected runtime. We first recall another designs for rejection sampling, namely the greedy technique described in [HJMR07], which optimizes the expected value of the logarithm of number of iterations. However, when we want to minimize the expected number of iterations, we show that rejection sampling as described in Section 2.2.3 is optimal.

4.2.1 Another Rejection Sampling Algorithm

In this section, we study the rejection sampling procedure described in [HJMR07].

Let $m > 0$, P_t and P_s two probability distributions over $\mathbb{Z}^m$ with $R_1(P_t \| P_s) < \infty$. Let $p_0(\mathbf{z}) = 0$ for any $\mathbf{z} \in \mathbb{Z}^m$ and recursively define the following:

$$\begin{cases} \alpha_i(\mathbf{z}) & = \min(P_t(\mathbf{z}) - p_{i-1}(\mathbf{z}), (1 - p^*_{i-1})P_s(\mathbf{z})), \\ p_i(\mathbf{z}) & = p_{i-1}(\mathbf{z}) + \alpha_i(\mathbf{z}), \\ p^*_i & = \sum_{(\mathbf{z}) \in \mathbb{Z}^m} p_i(\mathbf{z}). \end{cases}$$

Finally, define $\beta_i(\mathbf{z}) = \min\left(\frac{P_t(\mathbf{z}) - p_{i-1}(\mathbf{z})}{(1-p^*_{i-1})P_s(\mathbf{z})}, 1\right)$.

$\mathcal{A}$:	$\mathcal{A}'$:
1: $i \leftarrow 1$	1: Sample $\mathbf{z} \hookleftarrow P_t$.
2: $\mathbf{z} \hookleftarrow P_s$	2: Return $\mathbf{z}$.
3: $u \hookleftarrow [0,1]$	
4: **if** $u \leq \beta_i(\mathbf{z})$ **then**	
5: return $\mathbf{z}$	
6: **else**	
7: $i \leftarrow i+1$	
8: go to 2	
9: **end if**	

Figure 4.1: Greedy rejection sampling

Lemma 4.2 (Correctness)**.** *For any $i > 0$ and $\mathbf{z} \in \mathbb{Z}^m$, let $r(i, \mathbf{z})$ be the probability that $\mathcal{A}$ returns $\mathbf{z}$ after exactly i iterations. Then it holds that $r(i, \mathbf{z}) = \alpha_i(\mathbf{z})$ and $\sum_{j=1}^{\infty} r(j, \mathbf{z}) = P_t(\mathbf{z})$. Put differently, the statistical distance between the distribution of the output of $\mathcal{A}$ and $\mathcal{A}'$ is 0.*

Proof. The probability that $(i, \mathbf{z})$ is output is $P_s(\mathbf{z})\bar{p}_{i-1} \cdot \beta_i(\mathbf{z})$, where $\bar{p}_{i-1}$ denotes the probability that the $i-1$ first values are rejected. We then show by induction that $\bar{p}_i = 1 - p_i^*$ for any $i \in \mathbb{N}$. In the case $i = 0$, we have $\bar{p}_0 = 1 = 1 - p_0^*$.

Let us now assume that this holds for some $i \in \mathbb{N}$. By induction, let us compute $\bar{p}_{i+1} = \bar{p}_i \cdot \sum_{\mathbf{z} \in \mathbb{Z}^m} (1 - \beta_{i+1}(\mathbf{z}))P_s(\mathbf{z})$. We have:

$$\begin{aligned}
\bar{p}_{i+1} &= (1 - p_i^*) \sum_{\mathbf{z} \in \mathbb{Z}^m} (1 - \beta_{i+1}(\mathbf{z}))P_s(\mathbf{z}) \\
&= 1 - p_i^* - \sum_{\mathbf{z} \in \mathbb{Z}^m} \min(P_t(\mathbf{z}) - p_i(\mathbf{z}), P_s(\mathbf{z})(1 - p_i^*)) \\
&= 1 - p_i^* - \sum_{\mathbf{z} \in \mathbb{Z}^m} \alpha_{i+1}(\mathbf{z}) \\
&= 1 - p_{i+1}^*.
\end{aligned}$$

Then $\forall \mathbf{z} \in \mathbb{Z}^m, \forall i \in \mathbb{N}, r(i, \mathbf{z}) = \alpha_i(\mathbf{z})$. As $p_i(\mathbf{z}) = \sum_{j=1}^{i} \alpha_i(\mathbf{z})$, we study these partial sums and show that they indeed converge to $P_t(\mathbf{z})$. To do so, we recall the proof from [HJMR07, Claim IV.1]. We reproduce it here for completeness.

Let us first show that $\alpha_i(\mathbf{z}) \geq (P_t(\mathbf{z}) - p_{i-1}(\mathbf{z}))P_s(\mathbf{z})$. We have

$$\begin{aligned}
1 - p_{i-1}^* &= \sum_{\mathbf{z} \in \mathbb{Z}^m} (P_t(\mathbf{z}) - p_{i-1}(\mathbf{z})) \\
&\geq P_t(\mathbf{z}) - p_{i-1}(\mathbf{z}).
\end{aligned}$$

The above holds by definition of $\alpha_i(\mathbf{z})$: both $P_t(\mathbf{z}) - p_{-1}(\mathbf{z})$ and $(1 - p_{i-1}^*)P_s(\mathbf{z})$ are $\geq (P_t(\mathbf{z}) - p_{i-1}(\mathbf{z}))P_s(\mathbf{z})$. From that, we find that

$$P_t(\mathbf{z}) - p_i(\mathbf{z}) \leq (P_t(\mathbf{z}) - p_{i-1}(\mathbf{z}))(1 - P_s(\mathbf{z})),$$

and a straightforward induction show that this is $\leq P_t(\mathbf{z})(1 - P_s(\mathbf{z}))^i$. Finally, by definition of $p_{i-1}(\mathbf{z})$, it holds that $P_t(\mathbf{z}) - p_i(\mathbf{z}) \geq 0$. □

4.2.2 Optimality of the Expected Number of Iterations

We now analyze to which extent the expected number of iterations of the rejection step could be reduced in the case of exact rejection sampling from P to Q, and prove the classical strategy to be optimal. This question arises from the variety of rejection sampling techniques that have been studied in other fields.

There exist multiple variants of rejection sampling. For instance, the aforementioned procedure described in [HJMR07] takes a greedy approach to rejection sampling and differs from the one we presented up until now. We are in the setting where we have access to a sampler from distribution Q. These samples are denoted by $(X_i)_{i\geq 1}$ with $X_i \in \mathcal{X}$ for some set $\mathcal{X}$ and we are required to output a sample from the distribution P over $\mathcal{X}$. Any design of procedure is allowed, as long as the output is one of the observed samples X_i. Let i^* be the random variable denoting the number of samples observed by an algorithm and we wish to determine how small $\mathbb{E}(i^*)$ can be. We note that the work of [HJMR07], establishes that there exists a rejection sampling algorithm achieving $\mathbb{E}(\log i^*) = \log R_1(P\|Q)$ up to lower order terms in $R_1(P\|Q)$, and that this is optimal. Here, we show that the minimum value for $\mathbb{E}(i^*)$ is $R_\infty(P\|Q)$.

In this section, we model a rejection sampling algorithm by a family of randomized functions $A_i : \mathcal{X}^i \to \{1, \ldots, i\} \cup \{\mathrm{r}\}$. At step i, it sees the new sample X_i and based on $X_1, \ldots, X_i$ it computes $A_i(X_1, \ldots, X_i)$. If it is equal to r, the algorithm asks for one more sample and otherwise if $A_i(X_1, \ldots, X_i) \in \{1, \ldots, i\}$, the algorithm terminates and outputs the sample $X_{A_i(X_1,\ldots,X_i)}$. Note that the running time of the algorithm is defined by $i^* = \inf\{i \geq 1 : A_i(X_1, \ldots, X_i) \neq \mathrm{r}\}$. We only consider algorithms for which $i^* < \infty$ almost surely. Define the random variable $J = A_{i^*}(X_1, \ldots, X_{i^*}) \in \mathbb{N}_+$, note that $J \leq i^*$ and the output of the algorithm is X_J (i.e., the output sample may not be the last one that was generated).

Theorem 4.3. *Let P, Q be two discrete probability distributions. Any rejection sampling algorithm $(A_i)_{i\geq 1}$ sampling from P satisfies $\mathbb{E}(i^*) \geq R_\infty(P\|Q)$.*

Proof. We have by assumption for any $x \in \mathcal{X}$,

$$P(x) = \Pr[X_J = x] = \sum_{j=1}^{\infty} \Pr[J = j, X_j = x] \leq \sum_{j=1}^{\infty} \Pr[i^* \geq j, X_j = x],$$

where we used the fact that the event $[J = j]$ is contained in $[i^* \geq j]$. Now, observe that the event $[i^* < j]$ only depends on $X_1, \ldots, X_{j-1}$ and as such it is independent of the event $[X_j = x]$. This implies that $[i^* \geq j]$ is independent of $[X_j = x]$. Then

$$P(x) \leq \sum_{j=1}^{\infty} \Pr[i^* \geq j] \Pr[X_j = x] = \mathbb{E}(i^*)Q(x) \ .$$

□

In the context of Lyubashevsky's signature schemes with source distribution Q', target distribution P', challenge set $\mathcal{C}$ and signing key $\mathbf{S}$, we have $P = P' \otimes U(\mathcal{C})$ and Q would be the distribution of the pair $(\mathbf{z}, \mathbf{c})$ obtained by sampling $\mathbf{y}$ from Q', as well as $\mathbf{c}$ from $U(\mathcal{C})$ and defining $\mathbf{z} = \mathbf{y} + \mathbf{Sc}$.

The above proof can be adapted in the setting where P and Q are continuous distributions by considering a sequence of balls converging to $\{x\}$ instead of x.

4.3 Lower Bounds for Perfect Rejection Sampling

We start by studying *the case of perfect rejection sampling*, which corresponds to the setting of [Lyu09, DDLL13]. That is, we set $\varepsilon = 0$ in the formalism of Section 2.6. We prove two lower bounds: (1) regarding signature size in both unimodal and bimodal settings (Sections 4.3.1 and 4.3.2), and (2) regarding the expected number of iterations of the rejection step (Section 4.2.2).

First, we analyze to which extent the expected norm of a distribution P can be decreased, under the constraint that we can reject to it using shifted samples from Q, where the Euclidean norm of the shift is bounded from above. This gives lower bounds on the norm of the signature vector $\mathbf{z}$ in Lyubashevsky's signature scheme, as recalled in Section 2.6. We start by studying the easier case of continuous distributions, and then provide a way to discretize the results.

Second, we prove than the classical rejection sampling strategy described above is optimal if one aims to minimize the expected number of iterations of the rejection step in the case of perfect rejection sampling from P to Q. Specifically, the expected number of iterations of any strategy is at least $R_\infty(P\|Q)$, which is reached by classical rejection sampling.

4.3.1 Optimal Compactness in the Unimodal Setting

The main result of this subsection is the following.

Theorem 4.4. *Let $m > 1, t > 0$, $V = \mathcal{B}_m(t)$ and $M > 1$. Let $f, g : \mathbb{R}^m \to [0,1]$ be two probability densities over $\mathbb{R}^m$ such that $\sup_{\mathbf{v}\in V} R_\infty(f\|g_{+\mathbf{v}}) \leq M$. Then we have:*

$$\mathbb{E}_{\mathbf{x}\hookleftarrow f}(\|\mathbf{x}\|) \geq \frac{t}{M^{1/(m-1)} - 1}.$$

Note that we place ourselves in a setup where shifts belong to a hyperball. In the context of Lyubashevsky's signature scheme, the shift is $\mathbf{Sc}$, where $\mathbf{S}$ is the signing key and $\mathbf{c}$ is the challenge (which is part of the signature). As $\mathbf{S}$ is unknown, replacing the set of $\mathbf{Sc}$'s by a hyperball seems to be a reasonable approach. Refining this approximation would lead to significant difficulties in the proof, with unlikely gains.

We now discuss the parameters M and m. As we exhibit later in Lemma 3.7, the variable M is related to the rejection probability. The smaller M, the faster we expect signing to be. To obtain a signing algorithm that terminates in polynomial time with overwhelming probability, we are interested in $M \leq \mathsf{poly}(\lambda)$. Recall that $m = \Omega(\lambda)$. In this parameter regime, we have $t/(M^{1/(m-1)} - 1) \approx t(m-1)/\log M$.

The role of distribution g in Theorem 4.4 may seem puzzling, as it does not appear in the result. It acts as a control of the discrepancy of f: distribution f must be sufficiently wide to hide (in the Rényi divergence sense) a version of V that is blurred by g. This forces $\mathbb{E}_{\mathbf{x}\hookleftarrow f}(\|\mathbf{x}\|)$ to be rather large. The proof proceeds in two steps. The first one consists in showing that there is no point favoring any direction and that we can restrict the study to isotropic distributions, i.e., distributions whose density is a function of the norm of the vector. The proof proceeds by averaging on shells. Theorem 4.4 is then obtained by integrating the local constraint $\sup_{\mathbf{v}\in V} R_\infty(f\|g_{+\mathbf{v}}) \leq M$ over the whole support, with appropriate scaling.

Lemma 4.5. *Let $m \geq 1, t > 0$ and $V = \mathcal{B}_m(t)$. Let $f, g : \mathbb{R}^m \to [0,1]$ be two probability densities over $\mathbb{R}^m$ and define $M = \sup_{\mathbf{v}\in V} R_\infty(f\|g_{+\mathbf{v}})$. Then there exist two probability densities f^*, g^* that satisfy*

- $\sup_{\mathbf{v}\in V} R_\infty(f^*\|g^*_{+\mathbf{v}}) \leq M$,
- $\|\mathbf{x}\| = \|\mathbf{y}\| \implies g^*(\mathbf{x}) = g^*(\mathbf{y})$ *and* $f^*(\mathbf{x}) = f^*(\mathbf{y})$,
- $\mathbb{E}_{\mathbf{z}\hookleftarrow f}(\|\mathbf{z}\|) = \mathbb{E}_{\mathbf{z}\hookleftarrow f^*}(\|\mathbf{z}\|)$.

Proof. Let us first take care of the $m = 1$ case. Define $g^* : x \mapsto (g(x) + g(-x))/2$ as well as $f^* : x \mapsto (f(x) + f(-x))/2$. First, for any $x \in \mathbb{R}$ and $v \in [-t, t]$, we have $f(x) \leq M \cdot g(x - v)$ as well as $f(-x) \leq M \cdot g(-x + v)$. This implies that $R_\infty(f^*\|g^*) \leq M$. Now, by construction, these two functions are even. Moreover, they are normalized and are thus probability densities. Finally, we have the equality $\mathbb{E}_{x\hookleftarrow f^*}(|x|) = (\mathbb{E}_{x\hookleftarrow f}(|x|) + \mathbb{E}_{x\hookleftarrow f}(|-x|))/2 = \mathbb{E}_{x\hookleftarrow f}(|x|)$.

In the following, we assume that $m \geq 2$. To define f^* and g^*, we switch from Cartesian to hyperspherical coordinates. Let $(x_1, \ldots, x_m)$ and $(\rho, \theta_1, \ldots, \theta_{m-1})$ both representing $\mathbf{x}$ in respectively Cartesian and hyperspherical coordinates. They satisfy the relations

$$\begin{aligned}
\|\mathbf{x}\| &= \rho \\
x_1 &= \rho\cos(\theta_1) \\
x_2 &= \rho\sin(\theta_1)\cos(\theta_2) \\
\vdots &= \vdots \\
x_{m-1} &= \rho\left(\prod_{i\leq m-2}\sin(\theta_i)\right)\cos(\theta_{m-1}) \\
x_m &= \rho\left(\prod_{i\leq m-1}\sin(\theta_i)\right).
\end{aligned}$$

Let $\vec{\theta} = (\theta_1, \ldots, \theta_{m-1})$ and $\mathbf{x}(\rho, \vec{\theta})$ be the vector whose coordinates are defined as above. Notice that the absolute value of the determinant of the variable change Jacobian is of the form $\rho^{m-1}D(\vec{\theta})$ for some $D : [0,\pi)^{m-2} \times [0, 2\pi) \to \mathbb{R}_{\geq 0}$, as all columns except the first one are of the form $\rho \cdot \mathbf{y}_i(\vec{\theta})$, and the first column does not depend on ρ. It then holds that

$$1 = \int_{\mathbb{R}^m} f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_0^\infty \rho^{m-1} \int_{[0,\pi]^{m-2}\times[0,2\pi]} f(\mathbf{x}(\rho,\vec{\theta}))D(\vec{\theta})\,\mathrm{d}\vec{\theta}\,\mathrm{d}\rho.$$

We then define:

$$f^* : \mathbf{z} \mapsto \frac{\int_{[0,\pi]^{m-2}\times[0,2\pi]} f(\mathbf{x}(\|\mathbf{z}\|,\vec{\theta}))D(\vec{\theta})\,\mathrm{d}\vec{\theta}}{\int_{[0,\pi]^{m-2}\times[0,2\pi]} D(\vec{\theta})\,\mathrm{d}\vec{\theta}}.$$

This is a probability density, as it is integrable and $\int_{\mathbb{R}^m} f^*(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1$. The latter can be seen by switching once more to hyperspherical coordinates. By construction, it is isotropic. We also have $\mathbb{E}_{\mathbf{z}\hookleftarrow f}(\|\mathbf{z}\|) = \mathbb{E}_{\mathbf{z}\hookleftarrow f^*}(\|\mathbf{z}\|)$, which is also seen by applying

the same change of variables. We define g^* in the same way. It remains to prove that $\sup_{\mathbf{v}\in V} R_\infty(f^*\|g^*_{+\mathbf{v}}) \leq M$.

Let $\mathbf{z} \in \mathbb{R}^m$ and $\mathbf{v} \in V$. Let $\lambda = \|\mathbf{z}-\mathbf{v}\|/\|\mathbf{z}\|$. We consider the scaling $s : \mathbf{y} \mapsto \lambda\cdot\mathbf{y}$. For any $\mathbf{y} \in \mathcal{S}_m(\|\mathbf{z}\|)$ (i.e., any $\mathbf{y} \in \mathbb{R}^m$ with $\|\mathbf{y}\| = \|\mathbf{z}\|$), we have $f(\mathbf{y}) \leq M \cdot g(s(\mathbf{y}))$ as $s(\mathbf{y})$ can be written as $\mathbf{y}-\mathbf{v}'$, where $\mathbf{v}' \in V$. Indeed, using the triangle inequality:

$$\|s(\mathbf{y}) - \mathbf{y}\| = |\lambda - 1|\|\mathbf{z}\| = |\|\mathbf{z} - \mathbf{v}\| - \|\mathbf{z}\|| \leq \|\mathbf{v}\| \leq t.$$

Decomposing every element $\mathbf{y} \in \mathcal{S}_m(\|\mathbf{z}\|)$ in hyperspherical coordinates as above with $\rho = \|\mathbf{z}\|$ for unique $(\theta_1, \ldots, \theta_{m-1}) \in [0,\pi]^{m-2} \times [0, 2\pi]$, we multiply both sides by $D(\vec{\theta})$, which is nonnegative, and we integrate over $[0,\pi]^{m-2} \times [0, 2\pi]$ to get:

$$\int_{[0,\pi]^{m-2}\times[0,2\pi]} f(\mathbf{x}(\|\mathbf{z}\|, \vec{\theta}))D(\vec{\theta})\,\mathrm{d}\vec{\theta} \leq M \int_{[0,\pi]^{m-2}\times[0,2\pi]} g(\lambda\mathbf{x}(\|\mathbf{z}\|, \vec{\theta}))D(\vec{\theta})\,\mathrm{d}\vec{\theta}.$$

We recall the definition of λ and divide both sides by $\int_{[0,\pi]^{m-2}\times[0,2\pi]} D(\vec{\theta})\,\mathrm{d}\vec{\theta}$. □

We finally move on to proving Theorem 4.4.

Theorem 4.4. Thanks to Lemma 4.5, we can, without loss of generality, assume that both f and g are isotropic. For $k \geq 0$, we define $\mu_k = \int_0^\infty r^k f(r)\,\mathrm{d}r$, which is the k-th order moment of f. In particular, we have $\mu_{m-1} = 1/S_m$ and $\mu_m = \mathbb{E}_{\mathbf{x}\hookleftarrow f}(\|\mathbf{x}\|)/S_m$. Indeed, using a hyperspherical variable change, we see that, for any $\beta \in \{0, 1\}$:

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}\hookleftarrow f}(\|\mathbf{x}\|^\beta) &= \int_{\mathbb{R}^m} \|\mathbf{x}\|^\beta f(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&= \int_0^\infty \rho^{m-1+\beta} f(\rho) \int_{[0,\pi]^{m-2}\times[0,2\pi]} D(\vec{\theta})\,\mathrm{d}\vec{\theta}\,\mathrm{d}\rho \\
&= S_m \cdot \mu_{m-1+\beta}.
\end{aligned}$$

The above implies that $\mathbb{E}_{\mathbf{x}\hookleftarrow f}(\|\mathbf{x}\|) = \mu_m/\mu_{m-1}$.

For any $x \geq 0$ and $u \in [-t, t]$, it holds that $f(x) \leq M \cdot g(|x - u|)$. In particular, for $x \geq t$, we have $f(x - t) \leq M \cdot g(x)$. Let us multiply both sides by x^{m-1} and integrate over $[t, +\infty)$. With a change of variable on the left-hand side, this gives

$$\begin{aligned}
\int_0^\infty (x + t)^{m-1} f(x)\,\mathrm{d}x &\leq M \cdot \int_t^\infty x^{m-1} g(x)\,\mathrm{d}x \\
&\leq M \cdot \int_0^\infty x^{m-1} g(x)\,\mathrm{d}x \\
&= M \cdot \int_0^\infty x^{m-1} f(x)\,\mathrm{d}x,
\end{aligned}$$

by recognizing that the right-hand side is $M \cdot \mu_{m-1}$ (which is the same for f and g). Grouping everything on the same side, we have

$$0 \leq \int_0^\infty \left(Mx^{m-1} - (x + t)^{m-1}\right) f(x)\,\mathrm{d}x. \tag{4.1}$$

Let $C = t/(M^{1/(m-1)} - 1)$. For $m > 2$, we rewrite the integrand as

$$\begin{aligned} Mx^{m-1} - (x+t)^{m-1} &= \left(M^{\frac{1}{m-1}}x - (x+t)\right) \cdot \sum_{k=0}^{m-2} \left(xM^{\frac{1}{m-1}}\right)^k (x+t)^{m-2-k} \\ &= \left(M^{\frac{1}{m-1}} - 1\right)(x-C) \cdot \sum_{k=0}^{m-2} \left(xM^{\frac{1}{m-1}}\right)^k (x+t)^{m-2-k}. \end{aligned}$$

For $m = 2$, the above holds by replacing the sum by 1. Now, note that the inequality $xM^{1/(m-1)} \geq x + t$ holds if and only if $x \geq C$. Hence the following upper bound holds for any $x \geq 0$, if $m > 2$:

$$(x-C) \cdot \sum_{k=0}^{m-2} (xM^{\frac{1}{m-1}})^k (x+t)^{m-2-k} \leq (x-C)(m-1)M^{\frac{m-2}{m-1}}x^{m-2}.$$

When $m > 2$, we divide by $(M^{1/(m-1)}-1)M^{(m-2)/(m-1)}(m-1) > 0$ in Equation (4.1):

$$C \cdot \int_0^\infty x^{m-2} f(x)\,\mathrm{d}x \leq \int_0^\infty x^{m-1} f(x)\,\mathrm{d}x.$$

Note that it also holds for $m = 2$. This can be rewritten as $\mu_{m-1}/\mu_{m-2} \geq C$.

Observe that $\mu_m \cdot \mu_{m-2} \geq (\mu_{m-1})^2$. Indeed, the Cauchy-Schwarz inequality states that for any real random variables X, Y, it holds that $|\mathbb{E}(XY)|^2 \leq \mathbb{E}(X^2)\mathbb{E}(Y^2)$. We instantiate it with the two (non-independent) random variables $X = \|\mathbf{x}\|^{m/2}$ and $Y = \|\mathbf{x}\|^{(m-2)/2}$, where $\mathbf{x} \hookleftarrow f$. Then $XY = \|\mathbf{x}\|^{\frac{m}{2}+\frac{m-2}{2}} = \|\mathbf{x}\|^{m-1}$. To conclude, note $\mu_m \cdot \mu_{m-2} \geq (\mu_{m-1})^2$ implies that $\mu_m/\mu_{m-1} \geq \mu_{m-1}/\mu_{m-2} \geq C$. This completes the proof. □

For the discrete case, given a discrete distribution P, we let $f : \mathbf{x} \mapsto P(\lceil \mathbf{x} \rfloor)$ be a probability density over $\mathbb{R}^m$, and we have, by the triangle inequality

$$\mathbb{E}_{\mathbf{x}\hookleftarrow f}(\|\mathbf{x}\|) \leq \mathbb{E}_{\mathbf{x}\hookleftarrow P}(\|\mathbf{x}\|) + \frac{\sqrt{m}}{2}.$$

Theorem 4.4 can then be adapted to the discrete case, up to subtracting $\sqrt{m}/2$ from the lower bound. In all setups considered in this work, this term is significantly smaller than $t/(M^{1/(m-1)} - 1)$.

Corollary 4.6. *Let $m, M > 1$, $t > 0$, and $V = \mathcal{B}_m(t) \cap \mathbb{Z}^m$. Let P and Q be two discrete probability distributions over $\mathbb{Z}^m$ such that $\sup_{\mathbf{v}\in V} R_\infty(P\|Q_{+\mathbf{v}}) \leq M$. Then:*

$$\mathbb{E}_{\mathbf{x}\hookleftarrow P}(\|\mathbf{x}\|) \geq \frac{t}{M^{1/(m-1)} - 1} - \frac{\sqrt{m}}{2}.$$

4.3.2 Optimal Compactness in the Bimodal Setting

The following result holds in the bimodal setting, with a similar proof to Theorem 4.4.

Theorem 4.7. *Let $m \geq 3, t > 0$, $V = \mathcal{B}_m(t)$ and $M > 1$. Let $f, g : \mathbb{R}^m \to [0,1]$ be two probability densities over $\mathbb{R}^m$ such that $\sup_{\mathbf{v}\in V} R_\infty(f\|g_{\pm\mathbf{v}}) \leq M$, where $g_{\pm\mathbf{v}}$ is the density $\mathbf{x} \mapsto \frac{1}{2}(g(\mathbf{x}-\mathbf{v}) + g(\mathbf{x}+\mathbf{v}))$. Then the following holds:*

$$\mathbb{E}_{\mathbf{x}\hookleftarrow f}(\|\mathbf{x}\|) \geq \frac{t}{\sqrt{M^{\frac{2}{m-2}} - 1}}.$$

As in the unimodal case, we first radialize the densities.

Lemma 4.8. *Let $m \geq 1, t > 0$ and $V = \mathcal{B}_m(t)$. Let $f, g : \mathbb{R}^m \to [0,1]$ be two probability densities over $\mathbb{R}^m$ and define $M = \sup_{\mathbf{v}\in V} R_\infty(f\|g_{\pm\mathbf{v}})$, where $g_{\pm\mathbf{v}}$ is as in Theorem 4.7. Then there exist two probability densities f^*, g^* that satisfy*

- $\sup_{\mathbf{v}\in V} R_\infty(f^*\|g^*_{\pm\mathbf{v}}) \leq M$,
- $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^m, \|\mathbf{x}\| = \|\mathbf{y}\| \implies f^*(\mathbf{x}) = f^*(\mathbf{y})$ *and* $g^*(\mathbf{x}) = g^*(\mathbf{y})$,
- $\mathbb{E}_{\mathbf{z}\hookleftarrow f}(\|\mathbf{z}\|) = \mathbb{E}_{\mathbf{z}\hookleftarrow f^*}(\|\mathbf{z}\|)$.

Proof. We proceed as in the proof of Lemma 4.5 to define f^* and g^*, and use the same notations. Let

$$f^* : \mathbf{z} \mapsto \frac{\int_{[0,\pi]^{m-2}\times[0,2\pi]} f(\mathbf{x}(\|\mathbf{z}\|,\vec{\theta}))D(\vec{\theta})\,\mathrm{d}\vec{\theta}}{\int_{[0,\pi]^{m-2}\times[0,2\pi]} D(\vec{\theta})\,\mathrm{d}\vec{\theta}}.$$

We define g^* in the same way, swapping f with g. As for Lemma 4.5, the first two claims hold.

Let $\mathbf{z} \in \mathbb{R}^m$ and $\mathbf{v} \in V$. For any $\mathbf{y} \in \mathcal{S}_m(\|\mathbf{z}\|)$, let $s_\mathbf{y}$ be the rotation that maps $\mathbf{z}$ to $\mathbf{y}$. It is an isometry, so for any $b \in \{0,1\}$, it holds $\|\mathbf{y} + (-1)^b s_\mathbf{y}(\mathbf{v})\| = \|\mathbf{z} + (-1)^b s_\mathbf{y}(\mathbf{v})\|$. We also have $\|s_\mathbf{y}(\mathbf{v})\| = \|\mathbf{v}\|$, implying that $s_\mathbf{y}(\mathbf{v}) \in V$. Then for any $\mathbf{y} \in S_m(\|\mathbf{z}\|)$:

$$f(s_\mathbf{y}(\mathbf{z})) \leq \frac{M}{2}\left(g(s_\mathbf{y}(\mathbf{z}-\mathbf{v})) + g(s_\mathbf{y}(\mathbf{z}+\mathbf{v}))\right).$$

By construction, there exist $\vec{\theta}_0$ such that $s_{\mathbf{x}(\|\mathbf{z}\|,\vec{\theta})}(\mathbf{z}) = \mathbf{x}(\|\mathbf{z}\|, \vec{\theta} + \vec{\theta}_0)$. We multiply both sides with $D(\vec{\theta})$, which is nonnegative, and integrate over $[0,\pi]^{m-2}\times[0,2\pi]$. It yields the first claim, up to dividing by the normalisation constant. □

We finally move on to proving the main result.

Theorem 4.7. Thanks to Lemma 4.8, we assume without loss of generality that both f and g are isotropic. We define the moments of f, g by $\mu_k^{(\phi)} := \int_0^\infty x^k \cdot \phi(x)\,\mathrm{d}x$ for $k \geq 0$ and $\phi \in \{f, g\}$. We have $\mu_m^{(\phi)} = \mathbb{E}_{\mathbf{x}\hookleftarrow\phi}(\|\mathbf{x}\|)/S_m$ and $\mu_{m-1}^{(\phi)} = 1/S_m$ as in the proof of Theorem 4.4. For any $r \geq 0$, $u \in [0,t]$ and $\theta \in [0, 2\pi)$, it holds that:

$$f(r) \leq \frac{M}{2}\left(g\left(\sqrt{r^2+u^2-2ru\cos(\theta)}\right) + g\left(\sqrt{r^2+u^2+2ru\cos(\theta)}\right)\right).$$

We will only consider $\theta = \pi/2$ as it will suffice to obtain the bound. This gives for any $r \geq 0$ and $u \in [0,t]$:

$$f(r) \leq Mg\left(\sqrt{r^2+u^2}\right).$$

Let us then multiply both sides by $r\left(\sqrt{r^2+t^2}\right)^{m-2}$ and integrate over $\mathbb{R}_{\geq 0}$. On the right-hand side, we use the change of variable $y = \sqrt{r^2+t^2}$, which yields in turn $dy = r/\sqrt{r^2+t^2}dr$, to obtain:

$$\int_0^\infty r\left(\sqrt{r^2+t^2}\right)^{m-2} f(r)\,\mathrm{d}r \leq M \cdot \int_t^\infty y^{m-1} g(y)\,\mathrm{d}r.$$

Since $y \mapsto y^{m-1}g(y)$ takes values in $\mathbb{R}_{\geq 0}$, we have that $M/S_m = M\mu_{m-1}^{(g)}$ is an upper bound for the right-hand side. By reordering terms, we obtain:

$$0 \leq \int_0^\infty r\left[\left(M^{\frac{2}{m-2}}r^2\right)^{\frac{m-2}{2}} - \left(r^2+t^2\right)^{\frac{m-2}{2}}\right] f(r)\,\mathrm{d}r.$$

Now, note that for $m \geq 4$, we have:

$$\left(M^{\frac{2}{m-2}}r^2\right)^{\frac{m-2}{2}} - \left(r^2+t^2\right)^{\frac{m-2}{2}} = \frac{M^{\frac{2}{m-2}}r^2 - r^2 - t^2}{M^{\frac{1}{m-1}}r + \sqrt{r^2+t^2}} \sum_{k=0}^{m-3} \left(M^{\frac{2}{m-2}}r^2\right)^{\frac{k}{2}} \left(r^2+t^2\right)^{\frac{m-3-k}{2}}.$$

This also holds for $m = 3$ if replacing the sum by 1. Note that for $r \geq 0$, we have $M^{\frac{1}{m-1}}r + \sqrt{r^2+t^2} \geq t$. Let $C = t/(M^{\frac{2}{m-2}}-1)^{1/2}$. Note that $r^2+t^2 \leq M^{\frac{2}{m-2}}r^2$ holds if and only if $r \geq C$. Then for $r \geq 0$ and $m \geq 4$, we have

$$\left(M^{\frac{2}{m-2}}r^2\right)^{\frac{m-2}{2}} - \left(r^2+t^2\right)^{\frac{m-2}{2}} \leq (M^{\frac{2}{m-2}}-1)(r^2-C^2)\frac{m-2}{t} \cdot M^{\frac{m-3}{m-2}}r^{m-3}.$$

Since all constants are positive, we obtain (including for $m \geq 3$):

$$0 \leq \int_0^\infty (r^m - C^2 r^{m-2}) f(r)\,\mathrm{d}r.$$

Equivalently, $\mu_m^{(f)} \geq C^2\mu_{m-2}^{(f)}$, which we rewrite as $(\mu_m^{(f)}/\mu_{m-1}^{(f)}) \cdot (\mu_{m-1}^{(f)}/\mu_{m-2}^{(f)}) \geq C^2$. As we have seen in the proof of Theorem 4.4, the Cauchy-Schwarz inequality implies that $\mu_{m-1}^{(f)}/\mu_{m-2}^{(f)} \leq \mu_m^{(f)}/\mu_{m-1}^{(f)}$. This leads to the desired lower bound. □

For $M \leq \mathsf{poly}(\lambda)$ and $m = \Omega(\lambda)$ as in the discussion following Theorem 4.4, we have $t/(M^{2/(m-2)}-1)^{1/2} \approx t\sqrt{(m-2)/(2\log M)}$. Similarly to the unimodal case, the lower bound can be adapted to integer distributions with limited loss (for all setups considered in this work).

Corollary 4.9. *Let $m \geq 3, t > 0, V = \mathcal{B}_m(t) \cap \mathbb{Z}^m$ and $M > 1$. Let P and Q be two discrete probability distributions over $\mathbb{Z}^m$ such that $\sup_{\mathbf{v}\in V} R_\infty(P\|Q_{\pm\mathbf{v}}) \leq M$, where $Q_{\pm\mathbf{v}}$ is as in Theorem 4.7. Then the following holds:*

$$\mathbb{E}_{\mathbf{x}\hookleftarrow P}(\|\mathbf{x}\|) \geq \frac{t}{\sqrt{M^{\frac{2}{m-2}}-1}} - \frac{\sqrt{m}}{2}.$$

4.4 Approaching the Lower Bounds with Hyperballs

We show that continuous uniform distributions in hyperballs almost reach the lower bounds in both the unimodal and bimodal perfect rejection sampling settings. We also consider the imperfect unimodal setting and find parameters that are asymptotically at least as good as the ones obtained for the Gaussian distribution (using our analysis described in Section 3.4.1.3). As continuous hyperball uniform distributions are easier to study than their discrete counterpart, Further, we show that a slight modification of Lyubashevsky's signature allows for the target and source distributions to be continuous.

We also compare this choice of distributions with the uniform distributions in hypercubes and with Gaussians, both asymptotically and with concrete parameters.

4.4.1 Uniform Distributions in Hyperballs

The first step is to compute the divergence in the three settings: unimodal, either perfect or imperfect rejection sampling and bimodal perfect rejection sampling. The first case can actually be seen as a particular case of the second one, and we summarize both in the following lemma. The function I appearing in the statement is defined in Section 4.1, and comes into play when dealing with hyperspherical caps.

Lemma 4.10 (Smooth Divergence)**.** *Let $m \geq 1$ and $\mathbf{v} \in \mathbb{R}^m$. Let $\eta > 1$ and let $\varepsilon = I_{1-1/\eta^2}(\frac{m+1}{2}, \frac{1}{2})/2 \in [0, 1/2)$. Let $r, r' > 0$ with $r'^2 \geq r^2 + \|\mathbf{v}\|^2 + 2r\|\mathbf{v}\|/\eta$. Then:*

$$R_\infty^\varepsilon\Big(U(\mathcal{B}_m(r))\|U(\mathcal{B}_m(r', \mathbf{v}))\Big) = \left(\frac{r'}{r}\right)^m.$$

Let $M > 1$. The above is $\leq M$ if $r \geq \|\mathbf{v}\| \cdot \frac{\frac{1}{\eta} + \sqrt{\frac{1}{\eta^2} + M^{2/m} - 1}}{M^{2/m} - 1}$ and $r' = M^{1/m} r$.

For $\varepsilon = 0$, we have $\eta = 1$. In that case, we can set $r = \|\mathbf{v}\|/(M^{1/m} - 1)$, which almost matches the lower bound from Theorem 4.4. As seen in Section 4.1, for $\varepsilon = 2^{-c \cdot m}$ with a constant $c > 0$, we have that $1/\eta^2 \xrightarrow[m \to +\infty]{} 1 - 2^{-c}$. For ε satisfying $\varepsilon \geq 2^{-o(m)}$ and $\varepsilon = o(1/m)$ with m going to infinity, we have that $1/\eta^2 \sim -\log(\varepsilon)/m$.

Proof. Assume that there exists some cut $\mathcal{C}$ with $\mathrm{vol}(\mathcal{C})/V_m(r) \leq \varepsilon$ such that the divergence is defined, i.e., with $\mathcal{B}_m(r) \backslash \mathcal{C} \subseteq \mathcal{B}_m(r', \mathbf{v})$. Then the divergence is $(r'/r)^m$, as the ratio of densities is constant and equal to $(r'/r)^m$ over $\mathcal{B}_m(r) \setminus \mathcal{C}$. To prove the first claim, it hence suffices to show that such a cut $\mathcal{C}$ exists.

Let $\mathcal{C}_\eta := \{\mathbf{x} \in \mathcal{B}_m(r) | \langle \mathbf{x}, \mathbf{v} \rangle \geq -\|\mathbf{v}\| r/\eta\}$. This is the intersection of a ball with an affine half-space, i.e., an m-dimensional hyperspherical cap. By Lemma 4.1, its volume is $\frac{V_m(r)}{2} \cdot I_{1-1/\eta^2}(\frac{m+1}{2}, \frac{1}{2})$. The definition of η ensures that $\mathrm{vol}(\mathcal{C}_\eta)/V_m(r) = \varepsilon$. We now check that $\mathcal{B}_m(r) \setminus \mathcal{C}_\eta \subseteq \mathcal{B}_m(r', \mathbf{v})$. Let $\mathbf{x} \in \mathcal{B}_m(r) \setminus \mathcal{C}_\eta$. We have

$$\|\mathbf{x} - \mathbf{v}\| \leq \sqrt{r^2 + \|\mathbf{v}\|^2 + 2r\|\mathbf{v}\|/\eta}.$$

By assumption, the latter is no larger than r', implying that $\mathbf{x} \in \mathcal{B}_m(r', \mathbf{v})$.

Finally, by combine the condition on r and r' and the equality $r' = M^{1/m} r$, we get

$$r^2 + \|\mathbf{v}\|^2 + 2\frac{r\|\mathbf{v}\|}{\eta} \leq M^{2/m} r^2,$$

which is a degree-2 inequality on r. Solving it completes the proof. □

Lemma 4.11 (Divergence in the Bimodal Setting)**.** *Let $m \geq 1$ and $\mathbf{v} \in \mathbb{R}^m$. Let $r, r' > 0$ such that $r'^2 \geq r^2 + \|\mathbf{v}\|^2$. Let $U(\mathcal{B}_m(r'), \pm\mathbf{v})$ denote the distribution of $\mathbf{z}$ where $b \hookleftarrow U(\{0, 1\})$ and $\mathbf{z} \hookleftarrow U(\mathcal{B}_m(r', (-1)^b \mathbf{v}))$. Then:*

$$R_\infty\Big(U(\mathcal{B}_m(r))\|U(\mathcal{B}_m(r'), \pm\mathbf{v})\Big) = \big(1 + \chi_{<r+\|\mathbf{v}\|}(r')\big) \cdot \left(\frac{r'}{r}\right)^m,$$

where $\chi_{<r+\|\mathbf{v}\|}$ denotes the indicator function of the set $\{x \in \mathbb{R} | x \leq r + \|\mathbf{v}\|\}$. Let $M > 1$. The above is $\leq M$ if $r \geq \|\mathbf{v}\|/\sqrt{(M/2)^{2/m} - 1}$ and $r' = (M/2)^{1/m} r$.

Note that the choice of r almost matches the lower bound from Theorem 4.7.

Proof. The support of $U(\mathcal{B}_m(r'), \pm\mathbf{v})$ is exactly $\mathcal{B}_m(r', \mathbf{v}) \cup \mathcal{B}_m(r', -\mathbf{v})$ and its density is $\mathbf{z} \mapsto (\chi_{\mathcal{B}_m(r',\mathbf{v})}(\mathbf{z}) + \chi_{\mathcal{B}_m(r',-\mathbf{v})}(\mathbf{z}))/(2V_m(r'))$. Thus, the divergence is finite when $\mathcal{B}_m(r) \subseteq \mathcal{B}_m(r', \mathbf{v}) \cup \mathcal{B}_m(r', -\mathbf{v})$. It is the case if any $\mathbf{x}$ with $\|\mathbf{x}\| \leq r$ satisfies $\|\mathbf{x} - \mathbf{v}\| \leq r'$ or $\|\mathbf{x} + \mathbf{v}\| \leq r'$. We assume, w.l.o.g., that $\|\mathbf{x} - \mathbf{v}\| \leq \|\mathbf{x} + \mathbf{v}\|$. Then

$$\|\mathbf{x} - \mathbf{v}\| = \sqrt{\|\mathbf{x}\|^2 + \|\mathbf{v}\|^2 - 2\langle \mathbf{x}, \mathbf{v}\rangle} \leq \sqrt{\|\mathbf{x}\|^2 + \|\mathbf{v}\|^2}.$$

Thanks to the assumption on r and r', we conclude that the divergence is finite.

Now, the ratio of the densities only takes three values. If $\mathbf{x} \notin \mathcal{B}_m(r)$ then the ratio is 0. If $\mathbf{x} \in \mathcal{B}_m(r) \cap \mathcal{B}_m(r', \mathbf{v}) \cap \mathcal{B}_m(r', -\mathbf{v})$ then the ratio is $(r'/r)^m$. Finally, if $\mathbf{x}$ belongs to $\mathcal{B}_m(r) \cap \mathcal{B}_m(r', \mathbf{v})$ but not to $\mathcal{B}_m(r', -\mathbf{v})$, then the ratio is $2(r'/r)^m$. This last case only occurs if $\mathcal{B}_m(r) \not\subseteq \mathcal{B}_m(r', -\mathbf{v})$. This is the case only if $r' < r + \|\mathbf{v}\|$. This completes the proof of the first claim.

For the second claim, note that the assumption on r and r' is satisfied, and that the divergence bound is indeed $\leq M$. □

In this bimodal case, the rejection test is as follows. One computes the norms of both $\mathbf{z}$ and $\mathbf{z} - 2(-1)^b\mathbf{v}$, where $\mathbf{z} = \mathbf{y} + (-1)^b\mathbf{v}$. If only the first one is $\leq r$, then the sample is accepted. If both are $\leq r$, then it is accepted and rejected with probability 1/2.

Finally, in order to use the uniform distribution in a hyperball, we verify that there is sufficient min-entropy in the first n coordinates given the remaining $m - n$ coordinates.

Lemma 4.12. *Let $m \geq 6, n \geq 1$ and $r \geq 2\sqrt{m}$. Let $\mathbf{x} = (\mathbf{x}_0^\top | \mathbf{x}_1^\top)^\top$ be a random variable over $\mathbb{R}^m$ whose distribution is $U(\mathcal{B}_m(r))$, where $\mathbf{x}_0$ has dimension n. It holds that*

$$H_\infty\big(\lceil \mathbf{x}_0 \rfloor | \lceil \mathbf{x}_1 \rfloor\big)_{U(\mathcal{B}_m(r))} \geq \left(\log_2 \frac{1}{0.85}\right) \cdot n \ .$$

Proof. We omit the $U(\mathcal{B}_m(r))$ subscripts for the min-entropies. First, note that we have $H_\infty(\lceil \mathbf{x}_0 \rfloor | \lceil \mathbf{x}_1 \rfloor) \geq H_\infty(\lceil \mathbf{x}_0 \rfloor | \mathbf{x}_1)$. By definition of the conditional min-entropy, we have

$$2^{-H_\infty(\lceil \mathbf{x}_0 \rfloor | \mathbf{x}_1)} = \int_{\mathbf{x}_1 \in \mathcal{B}_{m-n}(r)} \max_{\mathbf{x}_0^{\mathrm{int}} \in \mathbb{Z}^n} \left(\int_{\mathbf{x}_0 \in \mathcal{B}_n^\infty(1/2, \mathbf{x}_0^{\mathrm{int}})} p_{\mathbf{x}_0, \mathbf{x}_1}(\mathbf{x}_0, \mathbf{x}_1)\, \mathrm{d}\mu(\mathbf{x}_0) \right) \mathrm{d}\mu(\mathbf{x}_1) \ ,$$

where the density satisfies

$$p_{(\mathbf{x}_0, \mathbf{x}_1)}(\mathbf{x}_0, \mathbf{x}_1) = \frac{1}{V_m(r)} \chi_{<r^2}(\|\mathbf{x}_0\|^2 + \|\mathbf{x}_1\|^2).$$

Recall that $\chi_{<r^2}(y) = 1$ if $y \leq r^2$ and 0 otherwise and that $V_m(r)$ is the volume of the Euclidean ball of radius r in dimension m.

The maximum is achieved when $\mathbf{x}_0^{\text{int}} = \mathbf{0}$. Indeed, for any $\mathbf{x}_0^{\text{int}} \in \mathbb{Z}^m$, we have

$$\begin{aligned}
&\int_{\mathbf{x}_0 \in \mathcal{B}_n^\infty(1/2, \mathbf{x}_0^{\text{int}})} \chi_{<r^2}(\|\mathbf{x}_0\|^2 + \|\mathbf{x}_1\|^2)\, \mathrm{d}\mu(\mathbf{x}_0) \\
&= \int_{\mathbf{x}_0 \in \mathcal{B}_n^\infty(1/2, 0)} \chi_{<r^2}(\|\mathbf{x}_0 + \mathbf{x}_0^{\text{int}}\|^2 + \|\mathbf{x}_1\|^2)\, \mathrm{d}\mu(\mathbf{x}_0) \\
&\leq \int_{\mathbf{x}_0 \in \mathcal{B}_n^\infty(1/2, 0)} \chi_{<r^2}(\|\mathbf{x}_0\|^2 + \|\mathbf{x}_1\|^2)\, \mathrm{d}\mu(\mathbf{x}_0) \ ,
\end{aligned}$$

where we used the fact that if $\|\mathbf{x}_0\|_\infty \leq \frac{1}{2}$ and $\mathbf{x}_0^{\text{int}} \in \mathbb{Z}^n$, then $\|\mathbf{x}_0 + \mathbf{x}_0^{\text{int}}\| \geq \|\mathbf{x}_0\|$. As a result, we can write

$$2^{-H_\infty(\lceil \mathbf{x}_0 \rfloor | \mathbf{x}_1)} = \Pr\left[\|\mathbf{x_0}\|_\infty \leq \frac{1}{2}\right] \ .$$

Now we use the sub-independence of the coordinates slabs in the Euclidean ball (see [BP98]), i.e., denoting $\mathbf{x_0} = (x_{01}, x_{02}, \ldots, x_{0n})^T$, we have

$$\Pr\left[\|\mathbf{x}_0\|_\infty \leq \frac{1}{2}\right] \leq \prod_{i=1}^{n} \Pr\left[|x_{0i}| \leq \frac{1}{2}\right] \ .$$

We now use Lemma 4.13 below to get

$$\begin{aligned}
\Pr\left[|x_{01}| \leq \frac{1}{2}\right] &= \Pr\left[\frac{|x_{01}|}{r} \leq \frac{1}{2r}\right] \\
&\leq \Pr\left[\frac{|x_{01}|}{r} \leq \frac{1}{4\sqrt{m}}\right] \\
&\leq 0.843 + 2\exp(-m) \ \leq \ 0.85 \ ,
\end{aligned}$$

for $r \geq 2\sqrt{m}$ and $m \geq 6$. □

As we used it in the proof above, we prove the following result.

Lemma 4.13. *Let $(x_1, \ldots, x_m)^T$ be uniformly chosen in the m-dimensional Euclidean ball of radius* 1*. Then*

$$\Pr\left[|x_1| \leq \frac{1}{4\sqrt{m}}\right] \leq 0.843 + 2\exp(-m) \ .$$

Proof. To show this, we use the fact (see [BGMN05]) that we can obtain samples $(x_1, \ldots, x_m)^T$ by first sampling m independent Gaussian variables $g_1, \ldots, g_m$ with densities $t \mapsto \rho_{1/\sqrt{2}}(t)/\sqrt{\pi}$ and then setting $(x_1, \ldots, x_m)^T = \frac{(g_1, \ldots, g_m)^T}{\sqrt{\sum_{i \leq m} g_i^2 + z}}$, where z is independent and has an exponential distribution (density $t \mapsto \exp(-t)$).

With this notation, we have, for all $\delta > 0$:

$$\begin{aligned}
\Pr\left[x_1^2 > \frac{\delta^2}{m}\right] &= \Pr\left[\frac{g_1^2}{\sum_{i\leq m} g_i^2 + z} > \frac{\delta^2}{m}\right] \\
&\geq \Pr\left[g_1^2 > 4\delta^2 \text{ and } \sum_{i\leq m} g_i^2 \leq 3m \text{ and } z \leq m\right] \\
&\geq \Pr\left[|g_1| > 2\delta\right] - \Pr\left[\sum_{i\leq m} g_i^2 > 3m\right] - \Pr\left[z > m\right] \\
&\geq \Pr\left[|g_1| > 2\delta\right] - \exp(-m) - \exp(-m) \ .
\end{aligned}$$

For the last inequality, note that the distribution of $2\sum_{i\leq m} g_i^2$ is the chi-squared distribution of parameter m. If F denotes its cumulative density function, then we have the tail bound $1 - F(x) \leq ((x/m)\exp(1 - x/m))^{m/2}$ for $x > m$, which we use with $x = 6m$. Taking $\delta = 1/2$ and numerically evaluating the first term allows to complete the proof of the lemma. □

4.4.2 Lyubashevsky's Signature with Continuous Distributions

We consider continuous distributions over hyperballs, which are not directly compatible with Lyubashevsky's signature scheme, as recalled in Section 2.6. We argue that it is possible to extend Lyubashevsky's signature scheme to the case of continuous distributions, and that this comes with very limited complications (in the case of Gaussians, it could be simpler to use continuous Gaussians with this modified scheme, than using discrete Gaussians with the original scheme).

We use the same notations as in Section 2.6, with the source density g and the target density f being with supports over $\mathbb{R}^m$ rather than $\mathbb{Z}^m$. The modified signature scheme handling continuous source and target distributions is presented in Figure 4.2. Key generation is unchanged from Figure 2.5. Concretely, the changes compared to the construction described in Figure 2.5 are as follows: (i) $\mathbf{y}$ is now sampled from a continuous distribution with density g, (ii) $\mathbf{c}$ is now computed as $H(\mathbf{A}\lceil\mathbf{y}\rfloor, \mu)$, (iii) with $\mathbf{z}$ still being defined as $\mathbf{y} + \mathbf{Sc}$, if the test passes, and the returned signature is now $(\lceil\mathbf{z}\rfloor, \mathbf{c})$. Note that if f and g were actually discrete densities, then we would exactly recover the scheme from Figure 2.5. For correctness, note that

$$\mathbf{A}\lceil\mathbf{z}\rfloor - \mathbf{Tc} = \mathbf{A}\lceil\mathbf{y} + \mathbf{Sc}\rfloor - \mathbf{Tc} = \mathbf{A}\left(\lceil\mathbf{y}\rfloor + \mathbf{Sc}\right) - \mathbf{Tc} = \mathbf{A}\lceil\mathbf{y}\rfloor,$$

where the second equality holds because $\mathbf{Sc}$ is an integer vector.

Looking back to Chapter 3, we explain how to tweak the analysis from Section 3.4.1 to the continuous case. Having large commitment min-entropy is implied by having that $H_\infty(\lceil\mathbf{x}_0\rfloor|\lceil\mathbf{x}_1\rfloor)$ is large, where $\mathbf{x} = (\mathbf{x}_0^\top|\mathbf{x}_1^\top)^\top$ is a random variable over $\mathbb{R}^m$ whose distribution is g and $\mathbf{x}_0$ has dimension n. In the case of the uniform distribution in a hyperball, this is provided by Lemma 4.12. We obtain the same flavor of zero-knowledge, as the data processing inequality of the statistical distance and the Rényi divergence are used similarly.

Finally, we note that the modified scheme involves computations over real numbers. These can be securely replaced by finite precision computations, using standard techniques such as described in [Pre17] and as we apply in Chapter 5.

$\mathsf{Sign}(\mu, \mathbf{A}, \mathbf{S})$:	$\mathsf{Verify}(\mu, \mathbf{z}, \mathbf{c}, \mathbf{A}, \mathbf{T} = \mathbf{AS})$:
1: $\mathbf{y} \hookleftarrow g$	1: **if** $\|\mathbf{z}\| \leq \gamma$ and $\mathbf{c} = H(\mathbf{Az} - \mathbf{Tc}, \mu)$ **then**
2: $\mathbf{c} \leftarrow H(\mathbf{A}\lceil \mathbf{y} \rfloor, \mu)$	2: return 1
3: $\mathbf{z} \leftarrow \mathbf{y} + \mathbf{Sc}$	3: **else**
4: $u \hookleftarrow U([0,1])$	4: return 0
5: **if** $u \leq \min\left(\frac{f(\mathbf{z})}{M \cdot g(\mathbf{y})}, 1\right)$ **then**	5: **end if**
6: return $(\lceil \mathbf{z} \rfloor, \mathbf{c})$	
7: **else**	
8: go to Step 1	
9: **end if**	

Figure 4.2: Lyubashevsky's signature scheme with continuous distributions.

4.4.3 Comparison with other Distributions

Let $t = \max_{\mathbf{S},\mathbf{c}} \|\mathbf{Sc}\|$. In Table 4.1, we summarize the expected norm of signatures (up to a constant factor) for diverse distributions P and Q, and for a target expected number of iterations M. We consider three specific pairs of distributions, two of them being previously considered distributions (Gaussians and uniforms in hypercubes), and the last one being uniform distributions in hyperballs, introduced above. We consider three different scenarios:

- unimodal distributions and perfect rejection sampling, in column $\varepsilon = 0$;
- unimodal distributions and imperfect rejection sampling – we use approximations specific to the choice of $\varepsilon \geq 2^{-o(m)}$ and $\varepsilon = o(1/m)$;
- bimodal source distribution, perfect rejection sampling, in column "Bimodal".

Note that the second scenario relies on our improved analysis relying on the Rényi divergence for the imperfect case (see Section 3.4.1.3). This parameter range for ε is not appropriate when using the analysis relying on the statistical distance.

In the last column, we emphasize if the rejection test is simple or not. For hyperballs, it consists in comparing the norm of the sample with the radius of the target hyperball in the unimodal case. For the bimodal case, one needs to compute two norms, and if necessary, to flip a coin (as discussed after Lemma 4.11).

The entries in the table are approximations for $m \to \infty$, $t = \omega(1)$ and $M = 2^{o(m)}$, and for a given choice of P, we optimize the parametrization of Q (e.g., the radius in case of a hyperball) to minimize the signature norm.

The values of the table are obtained by computing the parameters for the underlying distributions (radii r, r' of the hypercubes or hyperballs and standard deviation σ of Gaussians) for our constraints M and t. This is done by computing their (smooth) Rényi Divergence, as done in Lemmas 4.10 and 4.11 for hyperballs. Proofs for hypercubes and Gaussians can be found in Section 4.4.4. Given these parameters, the expected norm immediately follows ($r\sqrt{m}$ for a hypercube of radius r, σr for a Gaussian of standard deviation σ, and r for a hyperball of radius r). To conclude this section, we emphasize the following points:

- Gaussians and Hyperballs are asymptotically equivalent and reach the lower bounds in the bimodal setting; Hyperballs further reach our lower bound in

Choices for P and Q	$\varepsilon = 0$	$\varepsilon \geq 2^{-o(m)}$ and $\varepsilon = o(1/m)$	Bimodal	Rejection Test
Hypercubes	$\frac{tm^{3/2}}{\log M}$	$\frac{tm^{3/2}}{\log M}$	$\frac{tm^{3/2}}{\log M}$	Simple
Gaussians	∞	$\frac{t\sqrt{m}\sqrt{\log \frac{1}{\varepsilon} + \log M}}{\log M}$	$\frac{t\sqrt{m}}{\sqrt{\log M}}$	Complex
Hyperballs	$\frac{tm}{\log M}$	$\frac{t\sqrt{m}\sqrt{\log \frac{1}{\varepsilon} + \log M}}{\log M}$	$\frac{t\sqrt{m}}{\sqrt{\log M}}$	Simple

Table 4.1: Expected norm of signatures depending on the choice of distributions and (im)perfectness of rejection sampling.

the exact unimodal setting as well;

- Hyperballs enjoy a significantly simpler rejection test compared to Gaussians;
- The bimodal setting (in both Gaussian and Hyperballs cases) leads to the most compact signatures.

4.4.4 Divergence of Usual Distributions

Lastly, we bound the (smooth) Rényi divergence for distributions classically used in Lyubashevsky's signatures. This lets us build Table 4.1.

4.4.4.1 Uniform Distribution in Hypercubes

For simplicity and ease of implementation, some applications rely on uniform distributions in hypercubes. The following result is implicit in [Lyu09].

Lemma 4.14 (Rényi Divergence)**.** *Let $m \geq 1$ and $\mathbf{v} \in \mathbb{Z}^m$. Let $r, r' \geq 1/2$ such that $r' \geq r + \|\mathbf{v}\|_\infty$. Then it holds that*

$$R_\infty\Big(U(\mathcal{B}_m^\infty(r) \cap \mathbb{Z}^m) \| U(\mathcal{B}_m^\infty(r', \mathbf{v}) \cap \mathbb{Z}^m)\Big) \leq \left(\frac{2r'+1}{2r-1}\right)^m.$$

Let $M > 1$. The above is $\leq M$ if $r \geq \frac{\|\mathbf{v}\|_\infty + (M^{1/m}+1)/2}{M^{1/m}-1}$ and $r' = r + \|\mathbf{v}\|_\infty$.

Proof. For the divergence to be defined, we need $\mathcal{B}_m^\infty(r) \cap \mathbb{Z}^m \subseteq \mathcal{B}_m^\infty(r', \mathbf{v}) \cap \mathbb{Z}^m$. This is ensured by the constraint $r' \geq r + \|\mathbf{v}\|_\infty$. In that case, the divergence is the ratio of the number of elements in each support, leading to the upper bound. The second claim follows by elementary calculations. □

The downside of the infinite norm is its lack of geometry: the use of the scalar product induced by the Euclidean norm is crucial to improve the bounds for the smooth divergence and the divergence with a bimodal version of the distribution, both for Gaussian distributions and uniforms in hyperballs. In contrary, these two setting do not bring significant improvements to the radius condition from Lemma 4.14 when considering the hypercube-uniform distribution.

4.4.4.2 Gaussian Distributions

By Lemma 2.9 and the fact that $\lim_{a\to+\infty} R_a(P\|Q) = R_\infty(P\|Q)$, we see that the Rényi divergence of infinite order between two Gaussian distributions with same standard deviation but different centers is infinite. However Lemma 2.6 gives us a finite upper bound on the smooth divergence. The result is of the same flavour as [Lyu12, Lemma 4.5], but our proof clearly states the influence of ε on the bound.

Lemma 4.15 (Smooth Rényi Divergence)**.** *Let $m > 0, \mathbf{v} \in \mathbb{R}^m, \varepsilon \in (0,1)$ and $\sigma > 0$. We have:*

$$R^\varepsilon_\infty\left(D_{\mathbb{Z}^m,\sigma}\|D_{\mathbb{Z}^m,\sigma,\mathbf{v}}\right) \leq \exp\left(\frac{\|\mathbf{v}\|^2}{2\sigma^2} + \frac{\|\mathbf{v}\|\sqrt{2\log\frac{1}{\varepsilon}}}{\sigma}\right),$$

Let $M > 1$. The above is $\leq M$ if

$$\sigma \geq \frac{\|\mathbf{v}\|}{\sqrt{2}\log(M)}\left(\sqrt{\log\frac{1}{\varepsilon}} + \sqrt{\log\frac{1}{\varepsilon} + \log M}\right).$$

Proof. Combining Lemmas 2.9 and 2.6, we obtain that for any $a \in (1, +\infty)$:

$$R^\varepsilon_\infty(P\|Q) \leq \exp\left(\frac{a\|\mathbf{v}\|^2}{2\sigma^2} + \frac{1}{a-1}\log\frac{1}{\varepsilon}\right).$$

Let $a = 1 + \frac{\sigma}{\|\mathbf{v}\|}\sqrt{2\log\frac{1}{\varepsilon}}$, which minimizes the above quantity. This yields

$$R^\varepsilon_\infty(P\|Q) \leq \exp\left(\frac{\|\mathbf{v}\|^2}{2\sigma^2} + \sqrt{2}\frac{\|\mathbf{v}\|}{\sigma}\sqrt{\log\frac{1}{\varepsilon}}\right).$$

To find when this is $\leq M$, we take the logarithm and multiply by σ^2 on both sides. Solving a degree-2 equation in σ leads to the second claim. □

The following is borrowed from [DDLL13]. We prove it for the sake of completeness.

Lemma 4.16 (Rényi Divergence with a Bimodal Gaussian)**.** *Let $m \geq 1$, $\mathbf{v} \in \mathbb{R}^m$ and $\sigma > 0$. Then the following holds:*

$$R_\infty\left(D_{\mathbb{Z}^m,\sigma}\|BD_{\mathbb{Z}^m,\sigma,\mathbf{v}}\right) \leq \exp\left(\frac{\|\mathbf{v}\|^2}{2\sigma^2}\right).$$

It is an equality if $\mathbf{v} \in \mathbb{Z}^m$. Let $M \geq 1$. The bound is $\leq M$ if $\sigma \geq \|\mathbf{v}\|/(2\sqrt{\log M})$.

Proof. Let $\mathbf{z} \in \mathbb{Z}^m$. We have:

$$\begin{aligned}
\frac{D_{\mathbb{Z}^m,\sigma}(\mathbf{z})}{BD_{\mathbb{Z}^m,\sigma,\mathbf{v}}(\mathbf{z})} &= \frac{\rho_{\sigma,\mathbf{v}}(\mathbb{Z}^m)}{\rho_\sigma(\mathbb{Z}^m)} \cdot \frac{\exp\left(\frac{-\|\mathbf{z}\|^2}{2\sigma^2}\right)}{\exp\left(\frac{-\|\mathbf{z}\|^2-\|\mathbf{v}\|^2}{2\sigma^2}\right)\cosh\left(\frac{|\langle\mathbf{z},\mathbf{v}\rangle|}{\sigma^2}\right)} \\
&= \frac{\rho_{\sigma,\mathbf{v}}(\mathbb{Z}^m)}{\rho_\sigma(\mathbb{Z}^m)} \cdot \frac{\exp\left(\frac{\|\mathbf{v}\|^2}{2\sigma^2}\right)}{\cosh\left(\frac{|\langle\mathbf{z},\mathbf{v}\rangle|}{\sigma^2}\right)} \\
&\leq \frac{\rho_{\sigma,\mathbf{v}}(\mathbb{Z}^m)}{\rho_\sigma(\mathbb{Z}^m)} \cdot \exp\left(\frac{\|\mathbf{v}\|^2}{2\sigma^2}\right),
\end{aligned}$$

	Hypercube-Uniform			Previous Gaussian		
	Medium	Recommended	Very High	Medium	Recommended	Very High
Ring dimension ℓ	256	256	256	256	256	256
q	8380417	8380417	8380417	918529	918529	918529
$(n, m-n)$	$(4,4)$	$(6,5)$	$(8,7)$	$(3,4)$	$(4,5)$	$(6,7)$
η	2	4	2	1	1	1
S	N/A	N/A	N/A	50	55	65
τ	39	49	60	39	49	60
$t = S \cdot \sqrt{\tau}$	N/A	N/A	N/A	312	385	503
B	N/A	N/A	N/A	386K	313K	457K
γ_2	$\frac{q-1}{88}$	$\frac{q-1}{32}$	$\frac{q-1}{32}$	$\frac{q-1}{256}$	$\frac{q-1}{256}$	$\frac{q-1}{128}$
d	13	13	13	12	11	11
M	4.25	5.1	3.85	4	4	4
BKZ block-size b to break SIS	423 (417)	638 (603)	909 (868)	408 (350)	639 (552)	1018 (887)
Best known classical bit-cost	123 (121)	186 (176)	265 (253)	119 (102)	186 (161)	297 (259)
Best known quantum bit-cost	108 (107)	163 (157)	233 (223)	104 (89)	164 (141)	261 (227)
BKZ block-size b to break LWE	422	622	860	471	619	934
Best known classical bit-cost	123	181	251	137	181	273
Best known quantum bit-cost	108	159	221	121	159	240
Expected signature size	2420	3293	4595	2009	2571	3706
Expected public key size	1312	1952	2592	800	1184	1760

Table 4.2: Parameters for Dilithium and updated Dilithium-G.

where the last inequality comes from the fact that $\cosh(x) \geq 1$ for any $x \in \mathbb{R}$. Note that for $\mathbf{z} \in \mathbb{Z}^m$ orthogonal to $\mathbf{v}$, this upper bound is reached. Finally, using [MR07, Lemma 2.9], we have that $\frac{\rho_{\sigma,\mathbf{v}}(\mathbb{Z}^m)}{\rho_\sigma(\mathbb{Z}^m)} \leq 1$. If $\mathbf{v} \in \mathbb{Z}^m$ this is actually an equality. □

As a side note, we observe that this result can be extended to any order and compared to standard results between two Gaussian distributions.

Corollary 4.17. *Let $m \geq 1$, $\mathbf{v} \in \mathbb{R}^m$ and $\sigma > 0$. Then the following holds:*

$$\forall a \in [1, +\infty], R_a(D_{\mathbb{Z}^m,\sigma} \| BD_{\mathbb{Z}^m,\sigma,\mathbf{v}}) \leq \exp\left(\frac{\|\mathbf{v}\|^2}{2\sigma^2}\right) = (R_a(D_{\mathbb{Z}^m,\sigma} \| D_{\mathbb{Z}^m,\sigma,\mathbf{v}}))^{\frac{1}{a}}.$$

Proof. The Rényi divergence is increasing in its order. Thus the upper bound from Lemma 4.16 is also an upper bound for any order $a \in [1, +\infty]$. □

4.4.5 Concrete Parameters

In this section, we use the *core-SVP* methodology introduced in [ADPS16], a conservative security estimation method in lattice cryptography.

To study the concrete impact of the choice of distributions on signature size, we consider Dilithium. The left side of Table 4.2 shows the parameters for three security levels of the round-3 documentation of the CRYSTALS-Dilithium submission to the NIST post-quantum project [BDK+20]. The right side of Table 4.2 gives updated parameters for Dilithium-G, which relies on Gaussian distributions whose description is available in the first version of the eprint version of [DKL+18]. For this update, we set the value of M to 4 and aim for security levels consistent with those of Dilithium. We update the quantum costs by plugging in the improvements from [CL21].

In these schemes, the verification key is a module-LWE sample $\mathbf{Bs}_1 + \mathbf{s}_2$ where $\mathbf{s}_1$ and $\mathbf{s}_2$ have ℓ_∞-norms $\leq \eta$. For each coordinate, the lowest d bits are dropped. A parameter τ is used to control the ℓ_1-norm of any hashed value $\mathbf{c}$, so that $\mathbf{c}$ has sufficient min-entropy. In Dilithium-G, the bound t is $S\sqrt{\tau}$, where S is the median over

	Hyperball-Uniform			Improved Gaussian		
	Medium	Recommended	Very High	Medium	Recommended	Very High
Ring dimension ℓ	256	256	256	256	256	256
q	520193	520193	520193	758273	758273	758273
$(n, m-n)$	$(3,4)$	$(4,5)$	$(6,7)$	$(3,4)$	$(4,5)$	$(6,7)$
η	1	1	1	1	1	1
S	50	55	65	50	55	65
τ	39	49	60	39	49	60
$t = S \cdot \sqrt{\tau}$	312	385	503	312	385	503
B	197K	259K	246K	367K	265K	375K
γ_2	$\frac{q-1}{16}$	$\frac{q-1}{8}$	$\frac{q-1}{8}$	$\frac{q-1}{256}$	$\frac{q-1}{256}$	$\frac{q-1}{256}$
d	11	11	10	12	11	11
M	4	4	4	4	4	4
BKZ block-size b to break SIS	447 (381)	628 (541)	1091 (946)	404 (347)	650 (560)	1041 (906)
Best Known Classical bit-cost	130 (111)	183 (158)	319 (276)	118 (101)	190 (163)	304 (264)
Best Known Quantum bit-cost	114 (97)	161 (139)	280 (243)	103 (89)	167 (143)	267 (232)
BKZ block-size b to break LWE	494	650	977	478	629	948
Best Known Classical bit-cost	144	190	285	140	183	277
Best Known Quantum bit-cost	127	167	251	123	161	243
Expected signature size	1903	2473	3461	1921	2462	3553
Expected public key size	800	1056	1760	800	1184	1760

Table 4.3: Parameters for hyperball-uniform and improved Dilithium-G.

the key generation randomness of the largest singular value of $(\mathsf{rot}(\mathbf{s}_1)^\top, \mathsf{rot}(\mathbf{s}_2)^\top)^\top$. A rejection step is added in KeyGen to check that the key satisfies it. The value of the SIS bound for unforgeability is computed using [BDK+20, Equation (6)]. We multiply it by 2 to get the strong unforgeability bound. The security is estimated using block-size optimized BKZ to break the module-SIS or module-LWE instances.

For Dilithium, i.e., the hypercube version, we take $t_\infty = \tau\eta$ as a bound on the ℓ_∞-norm of the secret key, which drives the radius of the hypercube and subsequently the unforgeability SIS bound (in ℓ_∞-norm).

It was argued in [DKL+18] that it seems difficult for BKZ to solve SIS with ℓ_∞-norm bound close to q, i.e., ℓ_2-norm above q. To analyze the runtime of BKZ in the case of an ℓ_2-norm bound $B \geq q$, one can remove the trivial vectors of the input basis (i.e., the vectors with coordinates in $q\mathbb{Z}$) by some randomizing step. This approach was however not considered for Dilithium-G and q was chosen such that $B < q$, leading to bigger parameters overall. We keep this constraint, as it is difficult to analyze the effectiveness of the attack presented in [DKL+18] when $B > q$ for Euclidean norm.

Finally, the computation of the verification key and signature sizes (in bytes) is performed as in [BDK+20] and [DKL+18], respectively, with a different encoding. Namely, to compute signature sizes for the Gaussian version, we rely on a strategy explained in [ETWY22, Section 5], called range Asymmetric Numeral System, which allows one to encode the signature with an average bit-length reaching its entropy plus some constant overhead that we ignore. This technique is used to obtain the sizes in the right hand side of Table 4.2.

Next, we apply to Dilithium-G two modifications introduced in this work and introduce the Hyperball variant. In Table 4.3 (right side), we show the improvements we obtain when the standard deviation σ is computed using our refined bound from Lemma 4.15 on the smooth Rényi divergence between two Gaussians and instantiated with $\varepsilon = 2^{-64}$ (we set $Q_s = 2^{64}$) instead of $\varepsilon = 2^{-\lambda}$, as allowed by the use of Rényi divergence (as discussed in Section 3.1.2 and Section 3.4.1.3). Keeping $M = 4$, the

standard deviation σ drops from $11t$ to $6.85t$ and leads to an additional saving on the signature size. When compared to Dilithium, if we consider the sum of signature and verification key sizes, we obtain up to $\approx 30\%$ savings for the 'Recommended' parameter set, and $\approx 25\%$ for the others.

Finally, we explore the use of the continuous uniform distributions in hyperballs. We take the algorithms from Dilithium-G, which are adapted to radial distributions and replace the Gaussians with the continuous uniform distributions in hyperballs, adding coefficient-wise rounding to integers when computing commitments. To set parameters, the bound B is computed using the radius of the hyperball instead of the probabilistic upper bound on the norm of a Gaussian vector. In Table 4.3 (left side), we provide the instantiations that we obtained. We note that the signature sizes are very similar to the ones obtained with Gaussians.

Chapter 5

HAETAE: Hyperball bimodAl modulE rejecTion signAture schemE

This chapter presents HAETAE, a signature implementation relying on modules, bimodal hyperballs and Fiat-Shamir with aborts, whose complete specification can be found in [CCD+23]. HAETAE is a candidate to the Korean post-quantum competition[1] and NIST post-quantum signature competition[2]. In Section 5.2, we discuss the various optimisations we considered in the implementation. We give the full description of the scheme in Section 5.3 and discuss its performances in Section 5.4.

5.1 Additional Preliminaries

Before describing the scheme, we introduce a few additional notations.

5.1.1 Additional Notations

Let $\mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$ where n is a power of 2. For any $q > 0$ let $\mathcal{R}_q = \mathcal{R}/q\mathcal{R} = \mathbb{Z}_q[x]/(x^n+1)$. We identify $\mathcal{R}_2$ with the set of elements in $\mathcal{R}$ with binary coefficients. Let $\mathcal{R}_{\mathbb{R}} = \mathbb{R}[x]/(x^n + 1)$. Given $\mathbf{y} = (\sum_{0\le i<n} y_i x^i, \ldots, \sum_{0\le i<n} y_{nk-n+i} x^i)^\top \in \mathcal{R}_{\mathbb{R}}^k \supset \mathcal{R}^k$, we define its ℓ_2-norm as the ℓ_2-norm of its corresponding coefficient vector, i.e. we let $\|\mathbf{y}\|_2 = \|(y_0, \cdots, y_{nk-1})^\top\|_2$. For an integer η, we let the set of polynomials of degree less than n with coefficients in $[-\eta, \eta] \cap \mathbb{Z}$ be denoted by S_η. Let $\mathcal{B}_{\mathcal{R},m}(r, \mathbf{c}) = \{\mathbf{x} \in \mathcal{R}_{\mathbb{R}}^m | \|\mathbf{x} - \mathbf{c}\|_2 \le r\}$ (resp. $\mathcal{B}_{(1/N)\mathcal{R},m}(r, \mathbf{c}) = (1/N)\mathcal{R}^m \cap \mathcal{B}_{\mathcal{R},m}(r, \mathbf{c})$) denote the continuous (resp. discretized) hyperball with center $\mathbf{c} \in \mathcal{R}^m$ and radius $r > 0$ in dimension $m > 0$ (resp. for some integer N). When $\mathbf{c} = \mathbf{0}$, we omit the center.

5.1.2 High and Low Bits

An important compression technique introduced by [BG14] and implemented by the Dilithium team [DKL+18], consists in seding only the high bits of the lower part of

[1] `https://www.kpqc.or.kr/`

[2] `https://csrc.nist.gov/projects/pqc-dig-sig`

the signature vector $\mathbf{z}$. Moreover, we also rely on bit decomposition to encode the signature. We first recall the Euclidean division with a centered remainder.

Lemma 5.1. *Let $a \geq 0$, $b > 0$. Writing $a = bq + r$, $r \in [-b/2, b/2)$ is unique and:*

$$a = \left\lfloor \frac{a + b/2}{b} \right\rfloor \cdot b + (a \bmod^{\pm} b).$$

We define our base decomposition function.

Definition 5.1 (High and low bits)**.** *Let $r \in \mathbb{Z}$ and α be a power of two. Successively define $r_1 = \lfloor (r + \alpha/2)/\alpha \rfloor$ and $r_0 = r \bmod^{\pm} \alpha$. Finally, define the tuple:*

$$(\mathsf{LowBits}(r, \alpha), \mathsf{HighBits}(r, \alpha)) \;=\; (r_0, r_1).$$

We extend these definitions to vectors by applying them component-wise. This decomposition lets us recover the original element and we bound its components.

Lemma 5.2. *Let $\alpha = 2^d, d > 0$, $q = 2^{d-1} \cdot p + 1$ be a prime and $r \in \mathbb{Z}$. It holds that*

$$\begin{aligned} & r = \alpha \cdot \mathsf{HighBits}(r, \alpha) + \mathsf{LowBits}(r, \alpha), \\ & \mathsf{LowBits}(r, \alpha) \in [-\alpha/2, \alpha/2), \\ r \in [0, 2q-1] \implies & \mathsf{HighBits}(r, \alpha) \in [0, (2q-1)/\alpha]\,. \end{aligned}$$

Proof. By Lemma 5.1, there exists a unique representation

$$r = \lfloor (r + \alpha/2)/\alpha \rfloor \alpha + (r \bmod^{\pm} \alpha).$$

Identifying $\mathsf{HighBits}(r, \alpha)$ and $\mathsf{LowBits}(r, \alpha)$ above yields the first result.

Next, by definition of $\bmod^{\pm}$, we have that $r' \in [-\alpha/2, \alpha/2)$.

For the second range, since $\lfloor (r + \alpha/2)/\alpha \rfloor$ is a non-decreasing function, we only show that $\lfloor (2q - 1 + \alpha/2)/\alpha \rfloor \leq \lfloor (2q-1)/\alpha \rfloor$. We have $(2q - 1 + \alpha/2) \leq \lfloor (2q - 1)/\alpha \rfloor \alpha + \alpha - 1$ by assumption on q. Dividing by α and taking the floor yields the result. □

We define $\mathsf{HighBits}^{z1}(r) = \mathsf{HighBits}(r, 256)$ and $\mathsf{LowBits}^{z1}(r) = \mathsf{LowBits}(r, 256)$.

5.1.2.1 High and low bits for hint

To produce the hint that we send instead of the lower part of $\mathbf{z}$, we could use the previous bit decomposition. However, as noted in a preliminary version of [DKL+18, Appendix B], a modification allows to further reduce the entropy of the hint.

We pack the high bits in the range $[0, 2(q-1)/\alpha_{\mathsf{h}})$. This is possible if we use the range $[-\alpha_{\mathsf{h}}/2 - 2, 0)$ to represent the integers that are close to $2q - 1$.

Definition 5.2 (High and low bits for hint)**.** *Let q be a prime and $\alpha_{\mathsf{h}} | 2(q-1)$ be a power of two. Let $m = 2(q-1)/\alpha_{\mathsf{h}}$ and $r \in \mathbb{Z}$. Define*

$$r_1 = \mathsf{HighBits}(r \bmod^{+} 2q, \alpha_{\mathsf{h}}) \quad \textit{and} \quad r_0 = \mathsf{LowBits}(r \bmod^{+} 2q, \alpha_{\mathsf{h}}).$$

If $r_1 = m$, let $(r'_0, r'_1) = (r_0 - 2, 0)$. Else, $(r'_0, r'_1) = (r_0, r_1)$. We define:

$$(\mathsf{LowBits}^{\mathsf{h}}(r), \mathsf{HighBits}^{\mathsf{h}}(r)) \;=\; (r'_0, r'_1).$$

As before, we extend these definitions to vectors by applying them componentwise. This decomposition lets us recover the original element and we bound its components.

Lemma 5.3. *Let $r \in \mathbb{Z}$. Let q be a prime, $\alpha_{\mathsf{h}} | 2(q-1)$ be a power of two and define $m = 2(q-1)/\alpha_{\mathsf{h}}$. It holds that*

$$\begin{aligned} &r = \alpha_{\mathsf{h}} \cdot \mathsf{HighBits}^{\mathsf{h}}(r) + \mathsf{LowBits}^{\mathsf{h}}(r) \mod 2q, \\ &\mathsf{LowBits}^{\mathsf{h}}(r) \in [-\alpha_{\mathsf{h}}/2 - 2, \alpha_{\mathsf{h}}/2), \\ &\mathsf{HighBits}^{\mathsf{h}}(r) \in [0, m-1]\,. \end{aligned}$$

Proof. Let $r \in [0, 2q-1]$. Let r_0, r_1, r'_0, and r'_1 defined as in Definition 5.2. The equality $r'_0 + r'_1 \cdot \alpha_{\mathsf{h}} = r_0 + r_1 \cdot \alpha_{\mathsf{h}} \bmod 2q$ holds vacuously if $r'_0 = r_0$ and $r'_1 = r_1$.

If not, then $r'_0 = r_0 - 2$ and $r'_1 = r_1 - 2(q-1)/\alpha_{\mathsf{h}}$ and $r'_0 + r'_1\alpha_{\mathsf{h}} = r_0 + r_1\alpha_{\mathsf{h}} - 2q$. By Lemma 5.2, we get the first equality.

The second property stems from the second property in Lemma 5.2. The modifications to r_0 make r'_0 lie in the range $[-\alpha_{\mathsf{h}}/2 - 2, \alpha_{\mathsf{h}}/2)$.

The last property stems from the third property in Lemma 5.2 and the fact that if $r_1 = m$, then we have $r'_1 = 0$. □

5.2 Design Specifications

This section details all optimisations applied to HAETAE as well as the sampling algorithm used to sample $\mathbf{y}$ during signing.

5.2.1 Key Generation

The bimodal rejection sampling relies on having a key pair $(\mathbf{A}, \mathbf{s}) \in \mathcal{R}_p^{k\times(k+\ell)} \times \mathcal{R}_p^{k+\ell}$ such that $\mathbf{As} = -\mathbf{As} \bmod p$. To generate such a pair, following [DDLL13], we choose $p = 2q$ and aim at $\mathbf{As} = q\mathbf{j} \bmod 2q$ for $\mathbf{j} = (1, 0, \ldots, 0)^\top$.

5.2.1.1 Key Generation and Encoding

To build a key pair, we start from an MLWE sample $\mathbf{b} - \mathbf{a} = \mathbf{A}_0\mathbf{s}_0 + \mathbf{e}_0 \bmod q$, where $\mathbf{A}_0 \leftarrow U(\mathcal{R}_q^{k\times(\ell-1)})$, $\mathbf{a} \leftarrow U(\mathcal{R}_q^k)$ and $(\mathbf{s}_0, \mathbf{e}_0) \leftarrow U(S_\eta^{\ell-1} \times S_\eta^k)$. Let $\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_0$, we define $\mathbf{A} = (2(\mathbf{a} - \mathbf{b}_1) + q\mathbf{j} | 2\mathbf{A}_0 | 2\mathbf{I}_k)$ as well as $\mathbf{s} = (1 | \mathbf{s}_0 | (\mathbf{e}_0 - \mathbf{b}_0))$. One sees that $\mathbf{As} = q\mathbf{j} \bmod 2q$. The verification key is comprised of $\mathbf{b}_1$ and the seed that allows generating $\mathbf{A}_0$ and $\mathbf{a}$. The secret key is the seed used to generate $\mathbf{s}$ and $(\mathbf{A}_0, \mathbf{a})$.

It remains to choose the decomposition of $\mathbf{b}$, that we see as an nk-dimensional vector with coordinates in $[0, q-1]$. We choose $\mathbf{b}_0$ with coordinates in $\{-1, 0, 1\}$ such that if a coordinate of $\mathbf{b}$ is odd, then it is rounded to the nearest multiple of 4. We can then write $\mathbf{b} = \mathbf{b}_0 + 2\mathbf{b}_1$, where $\mathbf{b}_1$ is encoded using $\lceil \log_2(q) - 1 \rceil$ bits per coordinate. This is computed coordinate-wise with $\mathbf{b}_0 = (-1)^{\lfloor \mathbf{b}/2 \rfloor \bmod 2}\mathbf{b} \bmod 2$, i.e. one less bit than $\mathbf{b}$. In all of the following, we let $(\mathsf{LowBits}^{\mathsf{vk}}(\mathbf{b}), \mathsf{HighBits}^{\mathsf{vk}}(\mathbf{b}))$ denote $(\mathbf{b}_0, \mathbf{b}_1)$. When $\mathbf{b}$ is uniform, we notice that the coordinates of $\mathbf{b}_0$ roughly follow a (centered) binomial law with parameters $(2, 1/2)$, which experimentally leads to smaller choices for β, which we discuss and introduce now.

5.2.1.2 Rejection Sampling on the Key

A critical step of our scheme is bounding $\|\mathbf{s}c\|_2$, where $\mathbf{s}$ is generated as before and $c \in \mathcal{R}$ is a polynomial with coefficients in $\{0, 1\}$ and has less than or equal to τ nonzero coefficients. The lower this bound is, the smaller the signature is, which in turn leads to harder forging. In the key generation algorithm, we apply the following rejection condition for some heuristic value β:

$$\tau \cdot \sum_{i=1}^{m} \max_{j}^{i\text{-th}} \|\mathbf{s}(\omega_j)\|_2^2 + r \cdot \max_{j}^{(m+1)\text{-th}} \|\mathbf{s}(\omega_j)\|_2^2 \leq \frac{n\beta^2}{\tau},$$

where $m = \lfloor n/\tau \rfloor$ and $r = n \bmod \tau$. We argue that the left hand side is a bound on $\frac{n}{\tau} \cdot \|\mathbf{s}c\|_2^2$ and that this condition leads to asserting $\|\mathbf{s}c\|_2 \leq \beta$.

Lemma 5.4. *Let $n, \tau > 0$ and $m = \lfloor n/\tau \rfloor$ and $r = n \bmod \tau$. For any $c \in \{0,1\}^n$ with hamming weight τ and any secret $\mathbf{s} \in S_\eta^{k+\ell}$, the quantity $n\|c\mathbf{s}\|_2^2$ is bounded by*

$$\tau^2 \cdot \sum_{i=1}^{m} \max_{j}^{i\text{-th}} \|\mathbf{s}(\omega_j)\|_2^2 + r \cdot \tau \cdot \max_{j}^{(m+1)\text{-th}} \|\mathbf{s}(\omega_j)\|_2^2.$$

Proof. We first rewrite $\|\mathbf{s}c\|^2$ as:

$$\|\mathbf{s}c\|_2^2 = \frac{\sum_i |c(\omega_j)|^2 \cdot \|\mathbf{s}(\omega_j)\|_2^2}{n},$$

where $\mathbf{s}(\omega_j) = (\mathbf{s}_1(\omega_j), \cdots, \mathbf{s}_{k+\ell}(\omega_j))$, and ω_j's are the primitive $2n$-th roots of unity. For $n = m \cdot \tau + r$, let $m = \lfloor n/\tau \rfloor$ and $r = n \bmod \tau$. Since $\sum_{j=1}^{n} |c(\omega_j)|^2 = n\tau$ and

$$|c(\omega_j)|^2 = |\ \omega_{j,1} + \cdots + \omega_{j,\tau}|^2 \leq \tau^2,$$

we bound $\sum_{j=1}^{n} |c(\omega_j)|^2 \cdot \|\mathbf{s}(\omega_j)\|_2^2$ by rearrangement: let $m = \lfloor n/\tau \rfloor$ be the maximum number of $|c(\omega_j)|^2$'s that can be τ^2. By sorting $\|\mathbf{s}(\omega_j)\|_2$ in decreasing order,

$$\|\mathbf{s}(\omega_{\sigma(1)})\|_2 \geq \|\mathbf{s}(\omega_{\sigma(2)})\|_2 \geq \cdots \geq \|\mathbf{s}(\omega_{\sigma(n)})\|_2,$$

where σ is a permutation for the indices, we have

$$\sum_{j=1}^{n} |c(\omega_j)|^2 \cdot \|\mathbf{s}(\omega_j)\|_2^2 \leq \sum_{j=1}^{m} |c(\omega_{\sigma(j)})|^2 \cdot \|\mathbf{s}(\omega_{\sigma(j)})\|_2^2 + \sum_{j=m+1}^{n} |c(\omega_{\sigma(j)})|^2 \cdot \|\mathbf{s}(\omega_{\sigma(m+1)})\|_2^2.$$

Its maximum is reached when the m largest $\|\mathbf{s}(\omega_j)\|_2^2$'s are multiplied with τ^2's, i.e.

$$\begin{aligned}\sum_{j=1}^{n} |c(\omega_j)|^2 \cdot \|\mathbf{s}(\omega_j)\|_2^2 &\leq \sum_{j=1}^{m} \tau^2 \cdot \|\mathbf{s}(\omega_{\sigma(j)})\|_2^2 + \Big(\sum_{j=1}^{n} |c(\omega_j)|^2 - m\tau^2\Big) \cdot \|\mathbf{s}(\omega_{\sigma(j)})\|_2^2 \\ &= \tau^2 \cdot \sum_{j=1}^{m} \|\mathbf{s}(\omega_{\sigma(j)})\|_2^2 + r \cdot \tau \cdot \|\mathbf{s}(\omega_{\sigma(j)})\|_2^2.\end{aligned}$$

□

5.2.2 Fix-point-friendly Bimodal Hyperball Rejection Sampling

Rejected values are highly sensitive data. We adapt rejection sampling in order to make that step easier to mask and implement in constant-time.

5.2.2.1 Rejection Sampling

A first step towards this goal is to use discrete hyperball samples instead of continuous ones. This allows for a removal of floating-point arithmetic in the rejection step and replace it with an exact computation of the squared norm.

Lemma 5.5 (Bimodal Hyperball Rejection Sampling)**.** *Let n be the degree of $\mathcal{R}$, $c > 1$, $r, t, m > 0$, and $r' \geq \sqrt{r^2 + t^2}$. Define $M = 2(r'/r)^{mn}$ and set*

$$N \geq \frac{1}{c^{1/(mn)} - 1} \frac{\sqrt{mn}}{2} \cdot \left(\frac{c^{1/(mn)}}{r} + \frac{1}{r'} \right).$$

Let $\mathbf{v} \in \mathcal{R}^m \cap \mathcal{B}_{(1/N)\mathcal{R},m}(t)$. Let $p : \mathbb{R}^m \to \{0, 1/2, 1\}$ be defined as follows

$$p(\mathbf{z}) = \begin{cases} 0 & \text{if } \|\mathbf{z}\| \geq r, \\ 1/2 & \text{else if } \|\mathbf{z} - \mathbf{v}\| < r' \wedge \|\mathbf{z} + \mathbf{v}\| < r', \\ 1 & \text{otherwise.} \end{cases}$$

Then there exists $M' \leq cM$ such that the output distributions of the two algorithms from Figure 5.2 are identical.

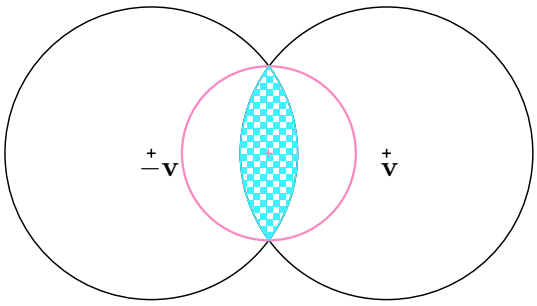

Figure 5.1: The HAETAE eyes

Figure 5.1 illustrates this rejection sampling technique. The black circles have radii equal to r' and the pink circle has radius r. We sample a vector $\mathbf{z}$ uniformly inside one of the black circles (with probability 1/2 for each) and keep $\mathbf{z}$ if it lies inside the pink circle but not in the blue zone. We reject it if it outside the circle, and keep it with probability 1/2 if it lies in the blue zone.

5.2.2.2 Discrete Hyperball-uniform Sampling

We first explain how to get continuous hyperball-uniform samples, using a folklore technique which yields such a n-dimensional sample using only $n + 2$-dimensional continuous Gaussians.

$\underline{\mathcal{A}(\mathbf{v}):}$
1: $\mathbf{y} \hookleftarrow U(\mathcal{B}_{(1/N)\mathcal{R},m}(r'))$
2: $\mathbf{b} \hookleftarrow U(\{0,1\})$
3: $\mathbf{z} \leftarrow \mathbf{y} + (-1)^b\mathbf{v}$
with probability $p(\mathbf{z})$
4: **return z**
5: **return** $\perp$

$\underline{\mathcal{B}:}$
1: $\mathbf{z} \hookleftarrow U(\mathcal{B}_{(1/N)\mathcal{R},m}(r))$
with probability $1/M'$
2: **return z**
3: **return** $\perp$

Figure 5.2: Bimodal hyperball rejection sampling

$\underline{\mathbf{y} \hookleftarrow U(\mathcal{B}_{\mathcal{R},k}(r'')):}$
1: $y_i \hookleftarrow \mathcal{N}(0,1)$ for $i = 0, \ldots, nk+1$
2: $L \leftarrow \|(y_0, \ldots, y_{nk+1})^\top\|_2$
3: $\mathbf{y} \leftarrow r''/L \cdot (\sum_{i=0}^{n-1} y_i\, x^i, \ldots, \sum_{i=0}^{n-1} y_{nk-n+i}\, x^i)^\top$ $\triangleright\ \mathbf{y} \in \mathcal{R}_{\mathbb{R}}^k$
4: **return y**

Figure 5.3: Continuous hyperball uniform sampling

Lemma 5.6 ([VGS17]). *The distribution of the output of the algorithm in Figure 5.3 is $U(\mathcal{B}_{\mathcal{R},k}(r''))$.*

Then, from a continuous sample, we get a discretized one by rounding it, and slightly reducing its radius.

$\underline{\mathbf{y} \hookleftarrow U(\mathcal{B}_{(1/N)\mathcal{R},m}(r')):}$
1: $\mathbf{y} \hookleftarrow U(\mathcal{B}_{\mathcal{R},m}(Nr' + \sqrt{mn}/2))$
if $\|\lfloor \mathbf{y} \rceil\|_2 \leq Nr'$
2: **return** $\lfloor \mathbf{y} \rceil / N$
else
3: **restart**

Figure 5.4: Discrete hyperball uniform sampling

Lemma 5.7. *Let n be the degree of $\mathcal{R}$, $M_0 \geq 1$, $r', m > 0$ and set*

$$N \geq \frac{\sqrt{mn}}{2r'} \cdot \frac{M_0^{1/(mn)} + 1}{M_0^{1/(mn)} - 1}.$$

At each iteration, the algorithm from Figure 5.4 succeeds with probability $\geq 1/M_0$. Moreover, the distribution of the output is $U(\mathcal{B}_{(1/N)\mathcal{R},m}(r'))$.

Finally, in the complete specification of HAETAE [CCD+23], we explain how to replace the continuous Gaussian with a discrete one as well as the necessary precision for those steps to succeed, using only fix-point arithmetic and standard techniques.

5.2.3 Challenge Sampling

The challenges we use are polynomials $c \in \mathcal{R}_2$ with τ nonzero coefficients. The challenge space has size $\binom{n}{\tau}$. To sample them, we rely on the SampleInBall algorithm from Dilithium, which we recall in Fig. 5.5.

SampleInBall(ρ, τ):

1: $\mathbf{c} \leftarrow c_0 c_1 \dots c_{255} = 00 \dots 0$
2: **For** $i = 256 - \tau$ to 255
3: $\quad j \leftarrow \{0, \dots, i\}$
4: $\quad c_i = c_j$
5: $\quad c_j = 1$
6: **return c**

Figure 5.5: Challenge sampling algorithm

For the highest security, however, we require 255 bits of entropy for the challenge, which cannot be reached with $\binom{256}{\tau}$. To achieve it, we replace the challenge sampling for the parameter set with the following. Given a 256-bits hash $w_0 \dots w_{255}$ with Hamming weight w, do the following. If $w < 128$, return $\sum_{i=0}^{255} w_i x^i$. If $w = 128$, return $\sum_{i=0}^{255} w_i \otimes w_0 x^i$. Otherwise, return $\sum_{i=0}^{255} w_i \otimes 1 x^i$. Exactly half of all binary polynomials are reachable this way and the challenge set has size 2^{255} as desired.

5.2.4 Signature Encoding

A signature is comprised of three elements $(\mathbf{h}, c, \mathbf{z}_1)$. We first split $\mathbf{z}_1$ into low and high bits such that the low bits are distributed almost uniformly. The high bits however can be compressed using the range Asymmetric Numeral System from [Dud13], which allows for a better compression rate than Huffman coding when encoding a stream of data. Furthermore, it is possible to avoid arithmetic operations altogether and realize high-speed implementations using lookup tables (tANS).

Definition 5.3 (Range Asymmetric Numeral System (rANS) Coding)**.** *Let $n > 0$ and $S \subseteq [0, 2^n - 1]$. Let $g : [0, 2^n - 1] \to \mathbb{Z} \cap (0, 2^n]$ such that $\sum_{x \in S} g(x) \leq 2^n$ and $g(x) = 0$ for all $x \notin S$. We define the following:*

- $\mathsf{CDF} : S \to \mathbb{Z}$*, defined as* $\mathsf{CDF}(s) = \sum_{y=0}^{s-1} g(y)$.
- $\mathsf{symbol} : \mathbb{Z} \to S$*, where* $\mathsf{symbol}(y) \mapsto s$ *such that* $\mathsf{CDF}(s) \leq y < \mathsf{CDF}(s+1)$.
- $C : \mathbb{Z} \times S \to \mathbb{Z}$*, defined as* $C(x, s) = \left\lfloor \frac{x}{g(s)} \right\rfloor \cdot 2^n + (x \bmod^+ g(s)) + \mathsf{CDF}(s)$.

The rANS encoding/decoding for S and frequency $g/2^n$ is defined in Figure 5.6.

Lemma 5.8 (Adapted from [Dud13])**.** *The rANS coding is correct, and the size of the rANS code is asymptotically equal to Shannon entropy of the symbols. That is, for any choice of $\mathbf{s} = (s_1, \cdots, s_m) \in S^m$, $\mathsf{Decode}(\mathsf{Encode}(\mathbf{s})) = \mathbf{s}$. Moreover, for any positive x and any probability distribution p over S, it holds that*

$$\sum_{s \in S} p(s) \log(C(x, s)) \leq \log(x) + \sum_{s \in S} p(s) \log\left(\frac{g(s)}{2^n}\right) + \frac{2^n}{x}.$$

Last, the cost of encoding the first symbol is $\leq n$ i.e. for any $s \in S, \log(C(0, s)) \leq n$.

We determine the frequency of the symbols experimentally, by executing the signature computation and collecting several million samples. Finally, we apply some rounding strategy to compute g such that the average extra cost per coordinate caused by this rounding is almost negligible.

$\underline{\mathsf{Encode}((s_1, \ldots, s_m) \in S^m)}$:

$x_0 = 0$
for $i = 0, \ldots, m-1$ **do**
  $x_{i+1} = C(x_i, s_{i+1})$
return x_m

$\underline{\mathsf{Decode}(x \in \mathbb{Z})}$:

$y_0 = x$
$i = 0$
while $y_i > 0$ **do**
  $t_{i+1} = \mathsf{symbol}(y_i \bmod^+ 2^n)$
  $y_{i+1} = \lfloor y_i / 2^n \rfloor \cdot g(t_{i+1}) + (y_i \bmod^+ 2^n) - \mathsf{CDF}(t_{i+1})$
  $i \leftarrow i + 1$
$m = i - 1$
return $(t_m, \ldots, t_1) \in S^m$

Figure 5.6: rANS encoding and decoding procedures

5.3 Description of HAETAE

We give the description of the signature scheme HAETAE in Figure 5.7 with the following building blocks, implemented with symmetric primitives:

- Hash function H_{gen} to generate the seeds and hashing the messages,
- Hash function H to sign, returning ρ, a seed for challenge sampling,
- Extendable output function expandA to derive $\mathbf{a}$ and $\mathbf{A}_{\mathsf{KeyGen}}$ from $\mathsf{seed}_{\mathbf{A}}$,
- Extendable output function expandS to derive $(\mathbf{s}_{\mathsf{gen}}, \mathbf{e}_{\mathsf{gen}})$ from $(\mathsf{seed}_{\mathsf{sk}}, \mathsf{counter}_{\mathsf{sk}})$,
- Extendable output function expandYbb to derive $(\mathbf{y}, b, b')$ from $(\mathsf{seed}_{ybb}, \mathsf{counter})$,

For the rest of this chapter, we let $\mathbf{j} = (1, 0, \ldots, 0) \in \mathcal{R}^k$. The parameters ρ_0 and α_{h} refer to the size of the seed and the compression factor, respectively. The parameter β is the bound for $\|c\mathbf{s}\|$, which will be checked by bounding

$$f(\mathbf{s}) \coloneqq \tau \cdot \sum_{i=1}^{m} \max_{j}^{i\text{-th}} \|\mathbf{s}(\omega_j)\|_2^2 + r \cdot \max_{j}^{(m+1)\text{-th}} \|\mathbf{s}(\omega_j)\|_2^2$$

by $n\beta^2/\tau$. The parameters B, B', and B'' refer to the radii of hyperballs. At Step 2 of the Sign algorithm, the variable $y_0 \in \mathcal{R}_{\mathbb{R}}$ refers to the first component of the vector $\mathbf{y} \in \mathcal{R}_{\mathbb{R}}^{k+\ell}$. At Step 3 of the Sign algorithm, the vector $\mathbf{z} \in \mathcal{R}_{\mathbb{R}}^{k+\ell}$ is decomposed as $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2)$ with $\mathbf{z}_1 \in \mathcal{R}_{\mathbb{R}}^{\ell}$ and $\mathbf{z}_2 \in \mathcal{R}_{\mathbb{R}}^{k}$. At Step 4 Verify, the variable $\tilde{z}_0 \in \mathcal{R}$ refers to the first component of the vector $\tilde{\mathbf{z}} \in \mathcal{R}^{k+\ell}$. We assume that α_{h} divides $2q - 1$. At Step 6 of Verify, the division by 2 is over $\mathbb{Z}$ and well-defined as the operand is even and defined modulo $2q$.

We also give a randomized signing of HAETAE in Figure 5.8. We observe that in the randomized version signing process, significant part of signing including the hyperball sampling algorithms for $\mathbf{y}$ can be performed "off-line", i.e., before receiving

$\underline{\mathsf{KeyGen}(1^\lambda)}$:

1: $\mathsf{seed} \leftarrow \{0,1\}^{\rho_0}$
2: $(\mathsf{seed}_{\mathbf{A}}, \mathsf{seed}_{\mathsf{sk}}, K) = H_{\mathsf{gen}}(\mathsf{seed})$
3: $(\mathbf{a}|\ \mathbf{A}_{\mathsf{gen}}) \in \mathcal{R}_q^{k\times\ell} \coloneqq \mathsf{expandA}(\mathsf{seed}_{\mathbf{A}})$
4: $\mathsf{counter}_{\mathsf{sk}} = 0$
5: $(\mathbf{s}_{\mathsf{KeyGen}}, \mathbf{e}_{\mathsf{KeyGen}}) \coloneqq \mathsf{expandA}(\mathsf{seed}_{\mathsf{sk}}, \mathsf{counter}_{\mathsf{sk}})$
6: $\mathbf{b} = \mathbf{a} + \mathbf{A}_{\mathsf{gen}} \cdot \mathbf{s}_{\mathsf{gen}} + \mathbf{e}_{\mathsf{gen}} \bmod q$ // $\mathbf{b} \in \mathcal{R}_q^k$
7: $(\mathbf{b}_0, \mathbf{b}_1) = (\mathsf{LowBits}^{\mathsf{vk}}(\mathbf{b}), \mathsf{HighBits}^{\mathsf{vk}}(\mathbf{b}))$
8: $\mathbf{A} = (2(\mathbf{a} - \mathbf{b}_1) + q\mathbf{j}|\ 2\mathbf{A}_{\mathsf{KeyGen}}|\ 2\mathbf{Id}_k) \bmod 2q$ // $\mathbf{A} \in \mathcal{R}_{2q}^{k\times(k+\ell)}$
9: $\mathbf{s} = (1, \mathbf{s}_{\mathsf{gen}}, \mathbf{e}_{\mathsf{gen}} - \mathbf{b}_0)$ // $\mathbf{s} \in S_\eta^{k+\ell}$
10: **if** $f(\mathbf{s}) > n\beta^2/\tau$ then go to 5
11: return $\mathsf{sk} = (\mathbf{s}, K)$, $\mathsf{vk} = (\mathsf{seed}_{\mathbf{A}}, \mathbf{b}_1)$

$\underline{\mathsf{Sign}(\mathsf{sk}, M)}$:

1: $\mu = H_{\mathsf{gen}}(\mathsf{seed}_{\mathbf{A}}, \mathbf{b}_1, M)$
2: $\mathsf{seed}_{ybb} = H_{\mathsf{gen}}(K, \mu)$
3: $\mathsf{counter} = 0$
4: $(\mathbf{y}, b, b') \coloneqq \mathsf{expandYbb}(\mathsf{seed}_{ybb}, \mathsf{counter})$
5: $\mathbf{w} = \mathbf{A}\,\lfloor \mathbf{y} \rceil$
6: $\rho = H(\mathsf{HighBits}^{\mathsf{h}}(\mathbf{w}), \mathsf{LSB}(\lfloor y_0 \rceil), \mu)$
7: $c = \mathsf{SampleInBall}(\rho, \tau)$
8: $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2) = \mathbf{y} + (-1)^b c \cdot \mathbf{s}$
9: $\mathbf{h} = \mathsf{HighBits}^{\mathsf{h}}(\mathbf{w}) - \mathsf{HighBits}^{\mathsf{h}}(\mathbf{w} - 2\,\lfloor \mathbf{z}_2 \rceil) \bmod^+ \frac{2(q-1)}{\alpha_{\mathsf{h}}}$
10: **if** $\|\mathbf{z}\|_2 \geq B'$, then $\mathsf{counter}{+}{+}$ and go to 4
11: **else if** $\|2\mathbf{z} - \mathbf{y}\|_2 < B$ **and** $b' = 0$, then $\mathsf{counter}{+}{+}$ and go to 4
12: **else** return $\sigma = (\mathsf{Encode}(\mathsf{HighBits}^{z1}(\lfloor \mathbf{z}_1 \rceil)), \mathsf{LowBits}^{z1}(\lfloor \mathbf{z}_1 \rceil), \mathsf{Encode}(\mathbf{h}), c)$

$\underline{\mathsf{Verify}(\mathsf{vk}, M, \sigma = (x, \mathbf{v}, h, c))}$:

1: $\tilde{\mathbf{z}}_1 \leftarrow \mathsf{Decode}(x) \cdot a + \mathbf{v}$ and $\tilde{\mathbf{h}} = \mathsf{Decode}(h)$
2: $(\mathbf{a}|\ \mathbf{A}_{\mathsf{KeyGen}}) = \mathsf{expandA}(\mathsf{seed}_{\mathbf{A}})$
3: $\mathbf{A}_1 = (2(\mathbf{a} - 2\mathbf{b}_1) + q\mathbf{j}|\ 2\mathbf{A}_{\mathsf{KeyGen}})$
4: $\mathbf{w}_1 = \tilde{\mathbf{h}} + \mathsf{HighBits}^{\mathsf{h}}(\mathbf{A}_1\tilde{\mathbf{z}}_1 - qc\mathbf{j}) \bmod^+ \frac{2(q-1)}{\alpha_{\mathsf{h}}}$
5: $w' = \mathsf{LSB}(\tilde{z}_0 - c)$
6: $\tilde{\mathbf{z}}_2 = [\mathbf{w}_1 \cdot \alpha_{\mathsf{h}} + w'\mathbf{j} - (\mathbf{A}_1\tilde{\mathbf{z}}_1 - qc\mathbf{j})]/2 \bmod^{\pm} q$
7: $\tilde{\mathbf{z}} = (\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2)$
8: $\tilde{\mu} = H_{\mathsf{KeyGen}}(\mathsf{seed}_{\mathbf{A}}, \mathbf{b}_1, M)$
9: Return $(c = \mathsf{SampleInBall}(H(\mathbf{w}_1, w', \tilde{\mu}), \tau)) \wedge (\|\tilde{\mathbf{z}}\| < B'')$

Figure 5.7: Deterministic version of HAETAE

a message M to be signed. It holds for computations such as $\mathbf{w} = \mathbf{A}\,\lfloor \mathbf{y} \rceil$ and $\mathsf{HighBits}^{\mathsf{h}}(\mathbf{w})$. In the "on-line" phase of signing, we can use $\mathbf{y}$ and the corresponding pre-computed components by choosing them randomly among the pre-sampled list.

Lemma 5.9. *We borrow the notations from Figure 5.7. If we run* $\mathsf{Verify}(\mathsf{vk}, M, \sigma)$ *on the signature* σ *returned by* $\mathsf{Sign}(\mathsf{sk}, M)$ *for an arbitrary message* M *and an arbitrary key-pair* $(\mathsf{sk}, \mathsf{vk})$ *returned by* $\mathsf{KeyGen}(1^\lambda)$*, then the following relations hold:*

$\underline{\mathsf{Sign}(\mathsf{sk}, M)}$:

// can be done off-line: using vk, make a list $\mathcal{L}$ of $(\mathbf{y}, \mathbf{w}, \mathbf{w}_1)$

$\mathbf{y} \leftarrow U(\mathcal{B}_{(1/N)\mathcal{R},(k+\ell)}(B))$

$\mathbf{w} = \mathbf{A} \lfloor \mathbf{y} \rceil$

$\mathbf{w}_1 = \mathsf{HighBits}^{\mathsf{h}}(\mathbf{w})$

// can be done on-line: using sk, M and pre-computed $(\mathbf{y}, \mathbf{w}, \mathbf{w}_1)$ sampled from $\mathcal{L}$

$\mu = H_{\mathsf{gen}}(\mathsf{seed}_{\mathbf{A}}, \mathbf{b}_1, M)$

$b, b' \leftarrow \{0, 1\}$

$c = \mathsf{SampleInBall}(H(\mathbf{w}_1, \mathsf{LSB}(\lfloor y_0 \rceil), \mu), \tau)$

$\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2) = \mathbf{y} + (-1)^b c \cdot \mathbf{s}$

$\mathbf{h} = \mathbf{w}_1 - \mathsf{HighBits}^{\mathsf{h}}(\mathbf{w} - 2 \lfloor \mathbf{z}_2 \rceil) \bmod^{+} \frac{2(q-1)}{\alpha_{\mathsf{h}}}$

if $\|\mathbf{z}\|_2 \geq B'$, then

    go to 5 with resampled $(\mathbf{y}, \mathbf{w}, \mathbf{w}_1)$ // resample $(\mathbf{y}, \mathbf{w}, \mathbf{w}_1) \leftarrow \mathcal{L}$

else if $(\|2\mathbf{z} - \mathbf{y}\|_2 < B) \wedge (b' = 0)$, then

    go to 5 with resampled $(\mathbf{y}, \mathbf{w}, \mathbf{w}_1)$ // resample $(\mathbf{y}, \mathbf{w}, \mathbf{w}_1) \leftarrow \mathcal{L}$

else return $\sigma = (\mathsf{Encode}(\mathsf{HighBits}^{z1}(\lfloor \mathbf{z}_1 \rceil)), \mathsf{LowBits}^{z1}(\lfloor \mathbf{z}_1 \rceil), \mathsf{Encode}(\mathbf{h}), c)$

Figure 5.8: Randomized signing of HAETAE. On/offline signing can accelerate the signing process. Note that the signing can also be accelerated even if $\mathbf{y}$ is sampled offline alone.

1. $\mathbf{w} = \mathsf{HighBits}^{\mathsf{h}}(\mathbf{w})$,
2. $w'\mathbf{j} = \mathsf{LSB}(\lfloor y_0 \rceil) \cdot \mathbf{j} = \mathsf{LSB}(\mathbf{w}) = \mathsf{LSB}(\mathbf{w} - 2 \lfloor \mathbf{z}_2 \rceil)$.
3. $2\lfloor \mathbf{z}_2 \rceil - 2\tilde{\mathbf{z}}_2 = \mathsf{LowBits}^{\mathsf{h}}(\mathbf{w}) - \mathsf{LSB}(\mathbf{w})$ *assuming* $B' + \alpha_{\mathsf{h}}/4 + 1 \leq B'' < q/2$,

Proof. Let $m = 2(q-1)/\alpha_{\mathsf{h}}$. Let us prove the first statement. By definition of $\mathbf{h}$, it holds that $\mathbf{w}_1 = \mathsf{HighBits}^{\mathsf{h}}(\mathbf{w}) \mod m$. However, both parts of the equality already lie in $[0, m-1]$ by definition and Lemma 5.3. Hence, the equality stands over $\mathbb{Z}$ too.

We move on to the second statement. By considering only the first component of $\mathbf{z} = \mathbf{y} + (-1)^b c \cdot \mathbf{s}$, we obtain, modulo 2:

$$\tilde{z}_0 = \lfloor z_0 \rceil = \lfloor y_0 \rceil + (-1)^b c = \lfloor y_0 \rceil + c.$$

This yields the result. Moreover, by reducing $\mathbf{A}$ mod 2, we obtain that

$$\mathbf{w} = \mathbf{A}_1 \lfloor \mathbf{z}_1 \rceil - qc\mathbf{j} = (\lfloor z_0 \rceil - c)\mathbf{j} \bmod 2.$$

For the last statement, let us use the two preceding results. In particular, we note

$$\mathbf{w}_1 \cdot \alpha_{\mathsf{h}} + w'\mathbf{j} = \mathbf{w} - \mathsf{LowBits}^{\mathsf{h}}(\mathbf{w}) + \mathsf{LSB}(\mathbf{w}).$$

We note that the last two elements have same parity, as the former one has the same parity as $\mathsf{LowBits}(\mathbf{w}, \alpha_{\mathsf{h}})$. By Lemma 5.3 their sum has infinite norm $\leq \alpha_{\mathsf{h}}/2 + 2$. Hence from its definition, it holds that

$$2\tilde{\mathbf{z}}_2 = 2\lfloor \mathbf{z}_2 \rceil - \mathsf{LowBits}^{\mathsf{h}}(\mathbf{w}) + \mathsf{LSB}(\mathbf{w}) \bmod^{\pm} 2q.$$

Finally, this identity holds over the integers as the right-hand side has infinite norm at most $2B' + \alpha_{\mathsf{h}}/2 + 2 < q$. □

Theorem 5.10 (Completeness)**.** *Let* $B'' = B' + \sqrt{n(k+\ell)}/2 + \sqrt{nk}(\alpha_{\mathsf{h}}/4 + 1) < q/2$*. Then the signature schemes of Figures 5.7 and 5.8 are complete, i.e., for every message* M *and every key-pair* $(\mathsf{sk}, \mathsf{vk})$ *returned by* $\mathsf{KeyGen}(1^\lambda)$*, we have:*

$$\mathsf{Verify}(\mathsf{vk}, M, \mathsf{Sign}(\mathsf{sk}, M)) = 1.$$

Proof. We use the notations of the algorithms. We focus on the deterministic version in Fig. 5.7, since Fig. 5.8 also has almost the same proof. The two first equations from Lemma 5.9 state that $\rho = \tilde{\rho}$ and thus $c = \mathsf{SampleInBall}(\rho, \tau)$. On the other hand, we use the last equation from Lemma 5.9 to bound the size of $\tilde{\mathbf{z}}$. We have:

$$\begin{aligned}
\|\tilde{\mathbf{z}}\| &\leq \|\mathbf{z}\| + \|\mathbf{z} - \lfloor \mathbf{z} \rceil\| + \|\lfloor \mathbf{z} \rceil - \tilde{\mathbf{z}}\| \\
&\leq B' + \sqrt{n(k+\ell)} \cdot \|\mathbf{z} - \lfloor \mathbf{z} \rceil\|_\infty + \|\lfloor \mathbf{z}_2 \rceil - \tilde{\mathbf{z}}_2\| \\
&\leq B' + \frac{\sqrt{n(k+\ell)}}{2} + \sqrt{nk} \cdot \|\mathsf{LowBits}^{\mathsf{h}}(\mathbf{w})\|_\infty \\
&\leq B' + \frac{\sqrt{n(k+\ell)}}{2} + \sqrt{nk} \cdot \left(\frac{\alpha_{\mathsf{h}}}{4} + 1\right) = B''.
\end{aligned}$$

□

5.4 Parameters and Performance Analysis

Finally, we give our choice of parameters and the performance of the implementation.

5.4.1 Parameter Sets

We instantiate the HAETAE signature scheme to reach the NIST PQC security levels 2, 3, and 5. The instantiations are set to be at least as secure as the corresponding parameter sets for Dilithium and Falcon. We keep the methodology from Section 4.4.5. Namely, we set $B'' \ll q$ and use the *core-SVP* methodology. The parameters are provided in Table 5.1. The figures between parentheses are for the strong unforgeability security in the case of the randomized signing version of HAETAE (in the deterministic version, strong and weak unforgeability are the same). More details on the practical attacks and theoretical security reduction are provided in the complete specification [CCD+23]. They are similar to the ones from Dilithium [BDK+20] and the reduction from Chapter 3.

5.4.2 Performance Analysis

The C reference implementation of HAETAE can be found on team HAETAE website[3].

In Table 5.2, we give the performance results of the reference implementation and the sizes. All benchmarks were obtained on one core of an Intel Core i7-10700k, with TurboBoost and hyperthreading disabled. All cycle counts reported are the median and average of the cycle counts of 1,000 executions of the respective functions.

Due to the key and the signature rejection steps, the median and the average values for KeyGen and Sign differ clearly. The two values are much closer for Verify.

[3] `https://kpqc.cryptolab.co.kr/haetae`

Parameter set NIST Security level	HAETAE-120 2	HAETAE-180 3	HAETAE-260 5
q	64513	64513	64513
M	6.0	5.0	6.0
Key Rate	0.1	0.25	0.1
β	354.82	500.88	623.72
B	9388.97	17773.21	22343.66
B'	9382.26	17766.15	22334.95
B''	12320.79	21365.10	24441.49
(k, ℓ)	(2,4)	(3,6)	(4,7)
η	1	1	1
τ	58	80	128
α_{h}	512	512	256
d	1	1	0
Forgery			
BKZ block-size b	409 (333)	617 (512)	878 (735)
Classical hardness	119 (97)	180 (149)	256 (214)
Quantum hardness	105 (85)	158 (131)	225 (188)
Key-Recovery			
BKZ block-size b	428	810	988
Classical hardness	125	236	288
Quantum hardness	109	208	253
Signature size	1463	2337	2908
Public key size	992	1472	2080
Sum	2455	3809	4988
Private key size	1376	2080	2720

Table 5.1: Parameter choices for 120, 180, 260 bits of core-SVP hardness

Parameter set		KeyGen	Sign	Verify
HAETAE-120	*med*	1,384,274	6,253,166	387,594
	ave	1,832,973	8,903,852	388,377
HAETAE-180	*med*	2,333,614	9,472,724	718,010
	ave	3,464,004	11,763,246	719,400
HAETAE-260	*med*	1,693,776	8,989,980	913,378
	ave	2,129,737	12,459,046	914,336

Table 5.2: Median and average cycle counts of 1000 executions for HAETAE.

Based on the profiling and benchmarking of subcomponents, we here discuss the most expensive parts during key generation and signing. During the key generation, the complex Fast Fourier Transformation, used for computing $f(\mathbf{s})$, consumes nearly 50% of the total cycles. Among the components of the signing process, we remark that the hyperball sampling is the most significant part, using almost 80% of the total signing cost, mainly because of randomness sampling via the extendable output function.

In addition, we expect that the on/offline approach will reduce the (online) signing time by 12% to 20%, except for the time spent reading from the list.

Chapter 6

G + G: Compact Lattice-Based Fiat-Shamir Signatures

The previous chapters were focused on lattice-based Fiat-Shamir *with aborts* signatures. In this chapter, we propose an identification scheme based on the same lattice assumptions as before, except that it has an aborting probability of zero. While this was already achieved in previous works (see [ASY22] for example), our work departs from these as this comes at no cost in terms of signature compactness. On the contrary, the asymptotic size of the signature is lower by a factor $\sqrt{m}/\sqrt{\log Q_s}$, where m is the dimension of the vector acting as the signature. We describe the scheme in Section 6.1 and show that it can be turned into an unforgeable signature scheme. In Section 6.2, we give concrete parameters for an optimized implementation relying on modules and using the compression techniques from [BG14].

6.1 The G + G Identification Protocol

In this section, we first describe the G + G identification protocol, then prove the required properties to compile it into a signature using the Fiat-Shamir heuristic, and then discuss asymptotic parameters.

6.1.1 Description of the Scheme

Let us first introduce the parameters of the scheme as well as some notations. Let $m \geq \ell > 0$, $k > m + \ell$ and $\mathbf{J} = \mathbf{J}_{m,\ell}$. Let χ be a distribution over $\mathbb{Z}$. Let $\mathcal{C} \subseteq \mathbb{Z}^\ell$ be the challenge space, which we assume to be finite. Let $\sigma, s \geq 0$ and define $\mathbf{\Sigma} : \mathbb{Z}^{k\times\ell} \to \mathbb{R}^{k\times k}$ as

$$\mathbf{\Sigma} : \mathbf{S} \mapsto \sigma^2\mathbf{I}_k - s^2\mathbf{S}\mathbf{S}^\top.$$

The scheme is also parametrized by an odd modulus q and an acceptance bound γ.

The G + G identification protocol is described in Figure 6.1. The instance generation algorithm samples a verification key $\mathbf{A} \in \mathbb{Z}_{2q}^{m\times k}$ and a secret key $\mathbf{S} \in \mathbb{Z}^{k\times\ell}$ with small-magnitude coefficients such that $\mathbf{A} \cdot \mathbf{S} = q\mathbf{J} \bmod 2q$. In the first phase of the interaction, the prover samples a vector $\mathbf{y}$ with well-crafted covariance matrix, and sends the commitment $\mathbf{w} = \mathbf{A}\mathbf{y} \bmod 2q$ to the verifier. The protocol is public-coin,

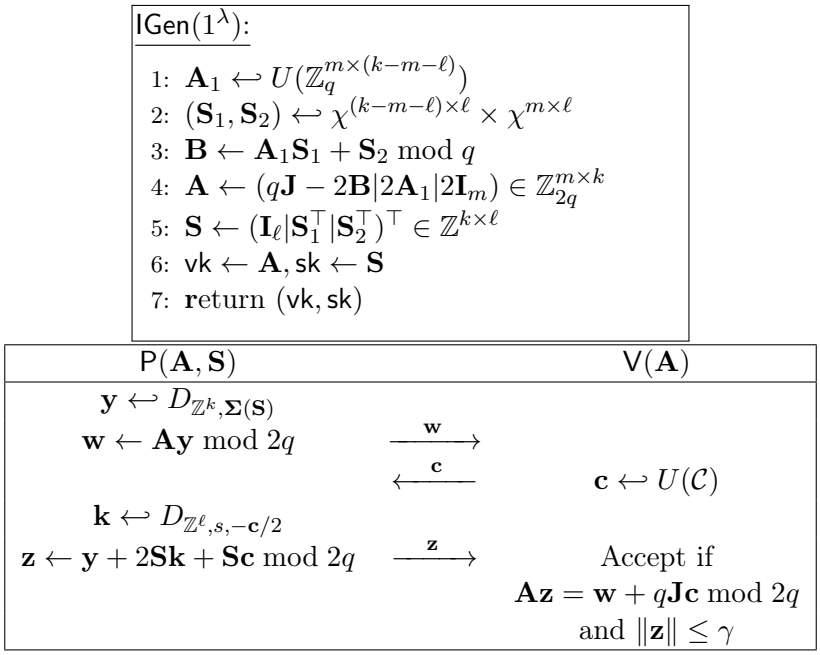

Figure 6.1: The G + G Identification Protocol.

i.e., the verifier just samples $\mathbf{c}$ uniformly in the challenge space and sends it to the prover. After receiving $\mathbf{c}$, the prover samples a Gaussian vector $\mathbf{k}$ over the lattice coset $2\mathbf{S}\mathbb{Z}^\ell + \mathbf{c}$. The covariance matrices of $\mathbf{y}$ and $\mathbf{k}$ are set so that the Gaussian plus Gaussian sum is statistically close to a spherical Gaussian distribution.

The first sampling that the prover has to perform is well-defined only if $\mathbf{\Sigma}(\mathbf{S})$ is definite positive, which we show in Lemma 2.14. The first sampling is implemented using Lemma 2.11, which requires $\sigma^2 - s^2\sigma_1(\mathbf{S})^2 \geq \sqrt{\ln(2\ell+4)/\pi}$, where we let $\sigma_1(\mathbf{S})$ denote the largest singular value of $\mathbf{S}$. The protocol can then be executed in polynomial time.

Combining this identification protocol with the Fiat-Shamir (without aborts) paradigm, we then obtain a lattice-based signature $\mathsf{FS}[\mathsf{G}+\mathsf{G}, H]$, as stated in the following Theorem. The correctness and security of the scheme are inherited from the properties of the underlying identification protocol.

Theorem 6.1. *Let $m \geq \ell > 0$, $k > m + \ell$, $\varepsilon \in (0, 1/2]$, $s \geq \sqrt{8\ln(\ell - 1 + 2\ell/\varepsilon)/\pi}$ and $\sigma \geq \sqrt{2}\sigma_1(\mathbf{S}) \cdot s$ for all $\mathbf{S} \in \mathbb{Z}^{k\times\ell}$ in the range of* IGen*. Let γ and ε_c be such that $\Pr_{\mathbf{z}\hookleftarrow D_{\mathbb{Z}^k,\sigma}}[\|\mathbf{z}\| > \gamma] \leq \varepsilon_c/3$. Let $q > \max(2\gamma, \sigma \cdot \eta_\varepsilon(\mathbb{Z}^m))$ be an odd modulus.*

Then the signature scheme $\mathsf{FS}[\mathsf{G}+\mathsf{G}, H]$ is:

- *ε_c-correct;*
- *sEU-CMA-secure in the ROM under the $\mathsf{SIS}_{m,k,q,2\gamma}$ assumption;*
- *sEU-CMA-secure in the QROM under the $\mathsf{LWE}_{k-m-\ell,m,\ell,\chi,q}$ assumption, assuming that $1/|\mathcal{C}| + (|\mathcal{C}|^2(2\gamma+1)^{2k})/q^m$ is negligible.*

The proof of Theorem 6.1 follows from Corollaries 6.3, 6.5, 6.7, and 6.8, which are derived from the properties of the underlying identification protocol proved in Sections 6.1.2, 6.1.3, and 6.1.4, by applying the Fiat-Shamir transform.

6.1.2 Completeness and Commitment Recoverability

We first show that the $\mathsf{G}+\mathsf{G}$ protocol is complete and commitment recoverable. As a corollary, we obtain that the resulting Fiat-Shamir signature scheme $\mathsf{FS}[\mathsf{G}+\mathsf{G}, H]$ is correct.

Theorem 6.2. *Let $m \geq \ell > 0$, $k > m + \ell$, $\varepsilon \in (0, 1/2]$, $s \geq \sqrt{8\ln(\ell - 1 + 2\ell/\varepsilon)/\pi}$ and $\sigma \geq \sqrt{2}\sigma_1(\mathbf{S}) \cdot s$ for all $\mathbf{S} \in \mathbb{Z}^{k\times\ell}$ in the range of IGen. Let γ and ε_c be such that $\Pr_{\mathbf{z}\hookleftarrow D_{\mathbb{Z}^k,\sigma}}[\|\mathbf{z}\| > \gamma] \leq \varepsilon_c/3$. Let $q > 2\gamma$ be an odd modulus. Then the $\mathsf{G}+\mathsf{G}$ identification protocol is ε_c-complete and achieves commitment-recoverability.*

Proof. First, we note that $\mathbf{AS} = q\mathbf{J} \bmod 2q$ holds for any matrix pair output by IGen. Then, in order to pass the first verification step, a transcript $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ must satisfy:

$$\mathbf{Az} = \mathbf{A}(\mathbf{y} + 2\mathbf{Sk} + \mathbf{Sc}) = \mathbf{w} + \mathbf{0} + q\mathbf{Jc} \bmod 2q \ . \tag{6.1}$$

In particular, this implies a commitment $\mathbf{w} = \mathbf{Az} - q\mathbf{Jc} \bmod 2q$, which is unique, such that $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ can be a valid transcript, and $\mathbf{w}$ is efficiently recoverable, by defining Rec as $\mathsf{Rec}(\mathbf{A}, \mathbf{c}, \mathbf{z}) := \mathbf{Az} - q\mathbf{Jc} \bmod 2q$.

Now, we note that a honestly generated transcript $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ always satisfies Equation 6.1. The probability preservation property of the Rényi divergence (Equation 2.2) and Lemma 2.14 then immediately imply that the probability that a honest transcript $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ be rejected at most $\leq 3 \cdot \Pr_{\mathbf{z}\hookleftarrow D_{\mathbb{Z}^k,\sigma}}[\|\mathbf{z}\| > \gamma]$. □

We then obtain the following corollary.

Corollary 6.3. *Using the same assumptions as in Theorem 6.2, the resulting signature scheme $\mathsf{FS}[\mathsf{G}+\mathsf{G}, H]$ is ε_c-correct.*

Note that correctness of $\mathsf{FS}[\mathsf{G}+\mathsf{G}, H]$ does not require to assume that H is modeled as a random oracle. As Lemma 2.14 holds without relying on the randomness of $\mathbf{c}$. This is in contrast to Lemma 2.15 that generically considers completeness of signatures obtained using the Fiat-Shamir transform.

6.1.3 Honest-Verifier Zero-Knowledge and Commitment Min-Entropy

We now show that the $\mathsf{G}+\mathsf{G}$ protocol is HVZK and has large commitment min-entropy. As a corollary, we obtain that the signature scheme $\mathsf{FS}[\mathsf{G}+\mathsf{G}, H]$ is $\mathsf{EU\text{-}CMA}$-secure provided it is $\mathsf{EU\text{-}NMA}$-secure.

Theorem 6.4. *Let $m \geq \ell > 0$, $k > m + \ell$, $\varepsilon \in (0, 1/2]$, $s \geq \sqrt{8\ln(\ell - 1 + 2\ell/\varepsilon)/\pi}$ and $\sigma \geq \sqrt{2}\sigma_1(\mathbf{S}) \cdot s$ for all $\mathbf{S} \in \mathbb{Z}^{k\times\ell}$ in the range of IGen. Let $q > \sigma \cdot \eta_\varepsilon(\mathbb{Z}^m)$ be an odd modulus. Then the $\mathsf{G}+\mathsf{G}$ identification protocol satisfies:*

- $(1+\varepsilon)/(1-\varepsilon)$*-divergence HVZK,*
- $2\varepsilon/(1-\varepsilon)$*-HVZK.*

In addition, its commitment min-entropy is $\geq m \cdot \log(s\sigma_1(\mathbf{S}))/3)$.

Proof. We prove both properties separately. We start by proving HVZK, which is inherited from Lemma 2.14 and then focus on commitment min-entropy.

HVZK. The simulator on input a challenge $\mathbf{c} \in \mathcal{C}$ and a public matrix $\mathbf{A}$ samples $\mathbf{z} \hookleftarrow D_{\mathbb{Z}^k,\sqrt{2}\sigma}$, sets $\mathbf{w} = \mathbf{A}\mathbf{z} - q\mathbf{J}\mathbf{c}$ and returns $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ as a transcript. As everything here is a function of $\mathbf{z}$ and $\mathbf{c}$, we can rely on Lemma 2.14. The bounds from the above claim are immediately inherited from the latter lemma by applying the data processing inequalities (which we recall in Equation 2.1 for the Rényi divergence – the same inequality holds replacing the Rényi divergence by the statistical distance). This concludes the zero-knowledge analysis.

Commitment Min-Entropy. Let $\mathbf{w} \in \mathbb{Z}_{2q}^m$ and $(Y_1^\top, Y_2^\top)^\top \hookleftarrow D_{\mathbb{Z}^k,\mathbf{\Sigma}(\mathbf{S})}$, where Y_1 takes values in $\mathbb{Z}^{k-m}$. Given a matrix $\mathbf{A} = (\mathbf{A}_0|2\mathbf{I}_m) \in \mathbb{Z}_{2q}^{m\times k}$, it holds that

$$\begin{aligned}\Pr_{(Y_1,Y_2)}[\mathbf{A}_0Y_1 + 2Y_2 = \mathbf{w} \bmod 2q] &= \Pr_{(Y_1,Y_2)}[2Y_2 = \mathbf{w} - \mathbf{A}_0Y_1 \bmod 2q] \\ &\leq \Pr_{(Y_1,Y_2)}[Y_2 = (\mathbf{w} - \mathbf{A}_0Y_1)\zeta \bmod q] \ ,\end{aligned}$$

where ζ is the modular inverse of 2 mod q. Hence, the min-entropy of the commitment is $\geq H_\infty(Y_2 \bmod q|Y_1)$ and we move on to bounding the latter quantity from below. Note that there exist $\sigma \geq \sigma_1 \geq \cdots \geq \sigma_m \geq (\sigma^2 - s^2\sigma_1(\mathbf{S})^2)^{1/2}$ and $\mathbf{Q} \in \mathbb{R}^{m\times m}$ orthogonal such that

$$\mathbf{\Sigma}(\mathbf{S}) = \mathbf{Q}\begin{pmatrix}\sigma_1^2 & & \\ & \ddots & \\ & & \sigma_m^2\end{pmatrix}\mathbf{Q}^\top.$$

Let $\mathbf{y}_1 \in \mathbb{Z}^{k-m}$ be fixed. The distribution of Y_2 conditioned on $Y_1 = \mathbf{y}_1$ is exactly $D_{\mathbb{Z}^m,\overline{\mathbf{\Sigma}},\overline{\mathbf{c}}}$, as defined in Lemma 2.10 (with $\mathbf{c} = \mathbf{0}$). Let $\overline{\sigma}_1^2$ (resp. $\overline{\sigma}_m^2$) be the largest (resp. smallest) eigenvalue of $\overline{\mathbf{\Sigma}}$ and $\overline{\mathbf{c}} = (\overline{c}_1, \ldots, \overline{c}_m)^\top$. We are interested in obtaining an upper bound on $\rho_{\overline{\mathbf{\Sigma}},\overline{\mathbf{c}}}(\mathbf{z} + q\mathbb{Z}^m)/\rho_{\overline{\mathbf{\Sigma}},\overline{\mathbf{c}}}(\mathbb{Z}^m)$ for all $\mathbf{z} \in (-q/2, q/2]^m$.

As $\overline{\mathbf{\Sigma}}^{-1}$ is the bottom right submatrix of $\mathbf{\Sigma}^{-1}$ of size $m \times m$, it holds that for any $\mathbf{y} \in \mathbb{R}^m$, we have $\mathbf{y}^\top\overline{\mathbf{\Sigma}}^{-1}\mathbf{y} \in \|\mathbf{y}\|^2 \cdot [1/\sigma_1^2, 1/\sigma_m^2]$. Hence all singular values $\overline{\sigma}_i$ of $\overline{\mathbf{\Sigma}}$ lie in $[(\sigma^2 - s^2\sigma_1(\mathbf{S})^2)^{1/2}, \sigma]$. Thanks to the theorem assumptions, we obtain that all $\overline{\sigma}_i$'s are above $\eta_\varepsilon(\mathbb{Z}^m)$. Using Lemma 2.13, it holds that

$$\rho_{\overline{\mathbf{\Sigma}},\overline{\mathbf{c}}}(\mathbb{Z}^m) \geq (1-\varepsilon)\cdot\sqrt{\det\overline{\mathbf{\Sigma}}} \geq (1-\varepsilon)\cdot\left(\sigma^2 - s^2\sigma_1(\mathbf{S})^2\right)^{m/2} \ .$$

The latter is $\geq (1-\varepsilon)\cdot(s\sigma_1(\mathbf{S}))^m$, by assumption on σ. For the numerator, we first use Lemma 2.13 once more, to obtain:

$$\rho_{\overline{\mathbf{\Sigma}},\overline{\mathbf{c}}}(\mathbf{z} + q\mathbb{Z}^m) \leq \rho_{\overline{\mathbf{\Sigma}}}(q\mathbb{Z}^m) = 1 + \rho_{\overline{\mathbf{\Sigma}}}(q\mathbb{Z}^m \setminus \{\mathbf{0}\}) \leq 1 + \rho_\sigma(q\mathbb{Z}^m \setminus \{\mathbf{0}\}) \ .$$

By assumption on q, the latter is $\leq 1 + \varepsilon$. The result follows.

□

We then obtain the following corollary as an application of Theorem 2.16.

Corollary 6.5. *Using the same assumptions as in Theorem 6.4 the resulting signature scheme* $\mathsf{FS}[\mathsf{G}+\mathsf{G}, H]$ *is* EU-CMA*-secure (and* sEU-CMA*-secure) in the QROM, provided it is* EU-NMA*-secure.*

6.1.4 Special Soundness and Lossy Soundness

To complete the analysis, we show that (i) $\mathsf{G}+\mathsf{G}$ is special-sound, and that (ii) $\mathsf{G}+\mathsf{G}$ is a lossy identification scheme with lossy-soundness. As a corollary, we obtain that the signature scheme $\mathsf{FS}[\mathsf{G}+\mathsf{G}, H]$ is $\mathsf{EU\text{-}NMA}$-secure in the ROM, and in the QROM under some parameters constraint.

Theorem 6.6. *Let $m \geq \ell > 0$, $k > m + \ell$, $\varepsilon \in (0, 1/2]$, $s \geq \sqrt{8\ln(\ell - 1 + 2\ell/\varepsilon)/\pi}$ and $\sigma \geq \sqrt{2}\sigma_1(\mathbf{S}) \cdot s$ for all $\mathbf{S} \in \mathbb{Z}^{k\times\ell}$ in the range of IGen. Let $\gamma > 0$ and $q > 2\gamma$ be an odd modulus. Then the $\mathsf{G}+\mathsf{G}$ identification protocol is:*

- *special-sound, under the $\mathsf{SIS}_{m,k,q,2\gamma}$ assumption,*
- *lossy, under the $\mathsf{LWE}_{k-m-\ell,m,\ell,\chi,q}$ assumption,*
- *$\varepsilon_{\mathsf{ls}}$-lossy sound for*

$$\varepsilon_{\mathsf{ls}} \;=\; \frac{1}{|\mathcal{C}|} + \frac{|\mathcal{C}|^2(2\gamma+1)^{2k}}{q^m} \;\;.$$

Proof. We first prove $\mathsf{G}+\mathsf{G}$ achieves special soundness, and then explain how to set our identification scheme in lossy mode.

Special soundness. Assume there exists a PPT adversary $\mathcal{A}$ which, given the verification key $\mathsf{vk} = \mathbf{A}$, produces two valid transcripts $(\mathbf{w}, \mathbf{c}_0, \mathbf{z}_0), (\mathbf{w}, \mathbf{c}_1, \mathbf{z}_1)$ with $\mathbf{c}_0 \neq \mathbf{c}_1$. It can be turned into an $\mathsf{SIS}_{m,k,q,2\gamma}$ solver. Indeed, by definition, such transcripts satisfy $\mathbf{A}(\mathbf{z}_0 - \mathbf{z}_1) = q\mathbf{J}(\mathbf{c}_1 - \mathbf{c}_0) \bmod 2q$.

Notice that we have $\mathbf{A}(\mathbf{z}_0 - \mathbf{z}_1) = \mathbf{0} \bmod q$, which implies that $\mathbf{z}_0 - \mathbf{z}_1$ is a solution to the SIS instance defined by $\mathbf{A}$. In addition, when reducing modulo 2, we also have $\mathbf{A}(\mathbf{z}_0 - \mathbf{z}_1) = \mathbf{J}(\mathbf{c}_1 - \mathbf{c}_0) \bmod 2$, which implies that $\mathbf{z}_0 \neq \mathbf{z}_1$. Finally, note that the condition on γ implies that $\|\mathbf{z}_0 - \mathbf{z}_1\| \leq 2\gamma$ (as transcript validity implies $\|\mathbf{z}\| \leq \gamma$), and that $\mathbf{z}_0 - \mathbf{z}_1 \neq \mathbf{0} \bmod q$.

Hence, there exists an adversary $\mathcal{B}$ against the $\mathsf{SIS}_{m,k,q,2\gamma}$ problem such that:

$$\mathsf{Adv}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{SIS}_{m,k,q,2\gamma}}(\mathcal{B}) \;\;.$$

Let us now focus on lossy-soundness. We first define a lossy key generation algorithm, and then argue about lossy-soundness.

Lossiness. The lossy key generation algorithm $\mathsf{LossyIGen}$ only modifies the generation of $\mathbf{B}$. Let us recall that in IGen, the latter is defined as $\mathbf{B} \leftarrow \mathbf{A_1}\mathbf{S_1} + \mathbf{S_2}$, with $\mathbf{A}_1 \hookleftarrow U(\mathbb{Z}_q^{m\times(k-m-\ell)})$ and $(\mathbf{S}_1, \mathbf{S}_2) \hookleftarrow \chi_\eta^{(k-m-\ell)\times\ell} \times \chi_\eta^{m\times\ell}$. The lossy key generation algorithm $\mathsf{LossyIGen}$ samples it as $\mathbf{B} \hookleftarrow U(\mathbb{Z}_q^{m\times\ell})$. Lossy verification keys are computationally indistinguishable from non-lossy ones, under the $\mathsf{LWE}_{k-m-\ell,m,\ell,\eta,q}$ assumption.

$\varepsilon_{\mathsf{ls}}$-lossy Soundness. First note that, if the lossy verification key $\mathbf{A}$ is such that, for all commitment $\mathbf{w}$, there exists at most one challenge $\mathbf{c}$ such that there exists $\mathbf{z}$ with $(\mathbf{w}, \mathbf{c}, \mathbf{z})$ passing verification, then, as the challenge is sampled uniformly and independently of $\mathbf{w}$, an (unbounded) prover cannot pass verification, except with probability at most $1/|\mathcal{C}|$.

We focus on proving that the above holds with overwhelming probability over the choice of the lossy key $\mathbf{A}$. By contradiction, assume there exists $\mathbf{w}, \mathbf{c}_0, \mathbf{c}_1, \mathbf{z}_0, \mathbf{z}_1$

with $\|\mathbf{z}_0\|, \|\mathbf{z}_1\| \leq \gamma$ and $\mathbf{c}_0 \neq \mathbf{c}_1 \in \mathcal{C}$, such that we have both $\mathbf{A}\mathbf{z}_0 = \mathbf{w} + q\mathbf{J}\mathbf{c}_0 \bmod 2q$ and $\mathbf{A}\mathbf{z}_1 = \mathbf{w} + q\mathbf{J}\mathbf{c}_1 \bmod 2q$. Then, we have:

$$\mathbf{A}(\mathbf{z}_0 - \mathbf{z}_1) = q\mathbf{J}(\mathbf{c}_1 - \mathbf{c}_0) \bmod 2q \ .$$

Recall that $\mathbf{A}$ is of the form $(q\mathbf{J} - 2\mathbf{B}|2\mathbf{A}_1|2\mathbf{I}_m)$, with $\mathbf{A}_1, \mathbf{B}$ uniform over $\mathbb{Z}_q$. Hence, the matrix $\mathbf{A} \bmod q$ is of the form $(\mathbf{B}|\mathbf{A}_1|\mathbf{I}_m)$, since q is odd. Then the above implies that $(\mathbf{B}|\mathbf{A}_1|\mathbf{I}_m)(\mathbf{z}_0 - \mathbf{z}_1) = \mathbf{0} \bmod q$ with $\mathbf{z}_0 - \mathbf{z}_1 \neq \mathbf{0} \bmod q$. This happens with probability at most $1/q^m$.

To conclude, note that there are at most $(2\gamma + 1)^{2k} \cdot |\mathcal{C}|^2$ choices for $\mathbf{z}_0, \mathbf{z}_1, \mathbf{c}_0$ and $\mathbf{c}_1$. A union bound therefore implies that the probability over $\mathbf{A}$ that there is a commitment with at least two challenges permitting valid transcripts is at most $|\mathcal{C}|^2(2\gamma + 1)^{2k}/q^m$. Our lossy identification scheme is then $\varepsilon_{\mathsf{ls}}$-lossy-sound, with

$$\varepsilon_{\mathsf{ls}} \ \leq \ \frac{1}{|\mathcal{C}|} + \frac{|\mathcal{C}|^2(2\gamma + 1)^{2k}}{q^m} \ ,$$

which completes the proof of the theorem. □

We then obtain the following corollary as an application of Lemma 2.19.

Corollary 6.7. *Using the same assumptions as in Theorem 6.6, the resulting signature scheme* $\mathsf{FS}[\mathsf{G} + \mathsf{G}, H]$ *is EU-NMA-secure, in the ROM.*

We also obtain the following corollary as an application of Theorem 2.17.

Corollary 6.8. *Using the same assumptions as in Theorem 6.6, and if* $\varepsilon_{\mathsf{ls}}$ *is negligible, the signature scheme* $\mathsf{FS}[\mathsf{G} + \mathsf{G}, H]$ *is EU-NMA-secure, in the QROM.*

To conclude this section, we introduce an additional assumption of a similar flavour as the SelfTargetMSIS assumption [KLS18], which allows to directly prove EU-NMA-security of $\mathsf{FS}[\mathsf{G} + \mathsf{G}, H]$ in the QROM as it is (up to LWE) the EU-NMA security game of the resulting signature. As for SelfTargetMSIS, this problem can be related in the ROM to SIS, using the special soundness property of the scheme.

Definition 6.1 (GpGSelfTargetSIS)**.** *Let* $m \geq \ell > 0$, $k > m + \ell$. *Let* $\gamma > 0$ *and* $q > 2\gamma$ *be an odd modulus. The* $\mathsf{GpGSelfTargetSIS}_{m,k,\ell,\gamma,q}$ *assumption states that given a matrix* $\mathbf{A} := (q\mathbf{J} - 2\mathbf{B}|2\mathbf{A}_1|2\mathbf{I}_m) \in \mathbb{Z}_{2q}^{m \times k}$, *where the matrices* $\mathbf{A}_1 \hookleftarrow U(\mathbb{Z}_q^{m \times (k-m-\ell)})$ *and* $\mathbf{B} \hookleftarrow U(\mathbb{Z}_q^{m \times \ell})$, *and oracle access to a hash function* H, *it is computationally hard to find* $\mathbf{c} \in \mathcal{C}$, $\mathbf{z} \in \mathbb{Z}^k$ *and* $\mu \in \{0,1\}^\star$ *such that* $H(\mathbf{A}\mathbf{z} - q\mathbf{J}\mathbf{c}, \mu) = \mathbf{c}$ *and* $\|\mathbf{z}\| \leq \gamma$.

6.1.5 Asymptotic Parameters Analysis

Our analysis above is applicable to the following instantiation of parameters, as a function of the security parameter λ and the number of signature queries Q_S. We assume Q_S to be a large polynomial in λ. We consider k, ℓ, m linear in λ. We set χ as $D_{\mathbb{Z},\sqrt{k}}$ with tail-cutting to get samples in $\{-k, \ldots, 0, \ldots, k\}$ with overwhelming probability. We let $\varepsilon = 1/Q_s$.

We make the security of the $\mathsf{G} + \mathsf{G}$ scheme rely on the following two assumptions. First, the $\mathsf{LWE}_{m-k-\ell,k,\ell,q,\chi}$ assumption, where $\sqrt{k} = \alpha q$. This LWE parametrization is compatible with the reduction from worst-case lattice problems from [Reg09].

$\mathsf{KeyGen}(1^\lambda)$:	$\mathsf{Sign}(\mathbf{A}, \mathbf{s}, \mu)$:	$\mathsf{Verify}(\mathbf{A}, \mu, \mathbf{z}, c)$:
1: $\mathbf{A}_0 \hookleftarrow U(\mathcal{R}_q^{m \times k-m-1})$	1: $\mathbf{y} \hookleftarrow D_{\mathcal{R}^k, \mathbf{\Sigma}(\mathbf{s})}$	1: $\mathbf{w} \leftarrow \mathbf{Az} - qc\mathbf{j} \bmod 2q$
2: **do** $(\mathbf{s}_1, \mathbf{s}_2) \hookleftarrow S_\eta^{k-m-1} \times S_\eta^m$	2: $\mathbf{w} \leftarrow \mathbf{Ay} \bmod 2q$	2: **if** $\mathbf{c} = H(\mathbf{w}, \mu)$
3: $\quad \mathbf{s} \leftarrow (1 \vert \mathbf{s}_1^\top \vert \mathbf{s}_2^\top)^\top \in \mathcal{R}_{2q}^k$	3: $c \leftarrow H(\mathbf{w}, \mu)$	3: and $\Vert\mathbf{z}\Vert \leq \gamma$ **then**
4: **while** $\Vert\zeta\mathbf{s}\Vert \geq S$	4: $u \hookleftarrow D_{\mathcal{R}, s, -c/2}$	4: $\quad$ **r**eturn 1
5: $\mathbf{b} \leftarrow \mathbf{A}_0\mathbf{s}_1 + \mathbf{s}_2 \bmod q$	5: $\mathbf{z} \leftarrow \mathbf{y} + (\zeta u + c)\mathbf{s}$	5: **end if**
6: $\mathbf{A} \leftarrow (-2\mathbf{b} + q\mathbf{j} \vert 2\mathbf{A}_0 \vert 2\mathbf{I}_m)$	6: **r**eturn $(\mathbf{z}, c)$	6: **r**eturn 0
7: **r**eturn $(\mathsf{vk}, \mathsf{sk}) = (\mathbf{A}, \mathbf{s})$		

Figure 6.2: The Module $\mathsf{G} + \mathsf{G}$ Signature Scheme.

Second, the $\mathsf{SIS}_{m,k,\beta}$ assumption, where $\beta = O(\sqrt{k}\sigma)$. The SIS parametrization is compatible with the reductions from worst-case lattice problems from [MR07,GPV08] when $q \geq \Omega(\sqrt{k}\beta)$. Finally, the hardness of both problems is balanced out when $\alpha \approx 1/\beta$.

The distribution of $\mathbf{z}$ is Gaussian with standard deviation $\sigma = 2\eta_\varepsilon(2\mathbf{S}\mathbb{Z}^\ell)$, which is $O(\sigma_1(\mathbf{S})\sqrt{\log(Q_s\lambda)})$. As $\sigma_1(\mathbf{S}) = O(\lambda)$, the norm of $\mathbf{z}$ is (almost) always at most $\beta = O(\lambda^{3/2}\log^{1/2} Q_s)$. Finally, we set $q = \Theta(\lambda^2 \log^{1/2} Q_s)$.

Verification keys and signatures have bit-sizes $O(\lambda^2 \log \lambda)$ and $O(\lambda \log \lambda)$.

6.2 Optimizations and Concrete Parameters

In order to decrease the sizes of a lattice-based scheme, a common approach is to replace $\mathbb{Z}$ with a cyclotomic polynomial ring of the form $\mathcal{R} = \mathbb{Z}[x]/(1 + x^n)$, where n is a power of 2, and to rely on the intractability of the module versions of SIS and LWE [BGV12, LS15]. Gaussian distributions are extended by considering the coefficients of the polynomials.

6.2.1 Description of the Module-Based Scheme

In this section, we propose parameters for an optimized, module version of the $\mathsf{G} + \mathsf{G}$ signature, that we present in Figure 6.2.

As in Section 6.1, let $m > 0$, $k > m + 1$ and $\ell = 1$. Let $\mathbf{j} = (\zeta^*, 0, \ldots, 0) \in \mathcal{R}^m$, where $\zeta = 1 + x^{n/2}$ and $\zeta^* = x^{n/2} - 1$ satisfy $\zeta^*\zeta = 2 \bmod 1 + x^n$. The challenge space is $\mathcal{R}/\zeta^*\mathcal{R}$. We let $\eta > 0$ and $\chi_\eta = U(\{y \in \mathcal{R} \mid \Vert y\Vert_\infty \leq \eta\})$. Given an element $s \in \mathcal{R}$, we define $\mathsf{rot}(s)$ as the $n \times n$ matrix whose (i, j)-th entry is the coefficient of degree $n - 1 - j$ of $x^i \cdot s \bmod 1 + x^n$. This matrix maps the coefficient embedding of a polynomial c to the coefficient embedding of sc. We extend this definition to vectors coordinate-wise and we define $\mathbf{\Sigma}(\mathbf{s}) = \mathbf{\Sigma}(\mathsf{rot}(\mathbf{s}))$, where $\mathbf{\Sigma}$ is borrowed from Section 6.1. This gives rise to the signature scheme presented in Figure 6.2.

Beyond relying on polynomial rings, we consider various improvements and optimizations, which we discuss now.

KeyGen: The key generation step includes a rejection sampling step. The threshold S will be set such that about 50% of the keys will be rejected. This helps

Target Security	120	180	260
n	256	256	256
q	95233	48640	202753
S	23.33	27.59	32.97
s	14.22	14.22	14.22
σ	331.91	392.57	469.12
γ	13885.1830	18857.9404	33367.4202
$(m, k-m)$	(2,4)	(3,5)	(4,7)
η	1	1	1
α	128	128	1024
BKZ block-size b to break SIS	415 (338)	616 (510)	924 (777)
Best Known Classical bit-cost	121 (98)	180 (149)	270 (227)
Best Known Quantum bit-cost	106 (86)	158 (131)	237 (199)
BKZ block-size b to break LWE	411	617	895
Best Known Classical bit-cost	120	180	261
Best Known Quantum bit-cost	105	158	230
Signature size with rANS	1542	2033	2518
Expected public key size	1120	1568	2336
Sum	2662	3601	4854
Signature size (Chapter 4)	1903	2473	3461
Public Key size	800	1056	1760
Sum	2703	3529	5221

Table 6.1: Parameter sets for the Optimized G + G Signature Scheme. Numbers in parentheses for SIS security are for strong-unforgeability.

controlling the upper bound on the smoothing parameter of the secret lattice.

Sign: Instead of computing $\mathbf{z} = \mathbf{y} + (2u + c)\mathbf{s}$, we compute $\mathbf{z} = \mathbf{y} + (\zeta u + c)\mathbf{s}$. Still, as $\mathbf{As} = \mathbf{j} \bmod 2q$, we have $\zeta\mathbf{As} = \mathbf{0} \bmod 2q$ by definition of $\mathbf{j}$. Thus, the identity $\mathbf{Az} - qc\mathbf{j} = \mathbf{Ay} \bmod 2q$ still holds. The main advantage of this modification is that the secret lattice is now $\zeta\mathbf{s}\mathcal{R}$ instead of $2\mathbf{s}\mathcal{R}$, whose smoothing parameter is a factor $\sqrt{2}$ smaller.

Verify: The verification bound is set to $\gamma = 1.01 \cdot \sqrt{nk}\sigma$, and the signer may verify that its signature is accepted before outputting it, up to restarting in the somewhat rare event that it is not.

6.2.2 Concrete Parameters

We now give concrete parameters and estimates of the public key and signature sizes resulting from these optimizations in Table 6.1. This gives rise to the following estimates. We note that the compression techniques from Chapter 5 can be used in this setting too. Namely, the high and low bits decomposition from Section 5.1.2, which increases the verification to $\gamma = 1.01 \cdot \sqrt{nk}\sigma + \sqrt{nm}(1 + \alpha/4)$, as well as the rANS technique described in Section 5.2.4.

For comparison, we include in Table 6.1 a reminder on estimated sizes of optimized Lyubashevsky signatures from Table 4.3. As far as we are aware of, these

$\mathsf{KeyGen}(1^\lambda)$:

1: **do** $(f, g) \hookleftarrow U(\{\mathbf{x} \in \mathbb{R}^2[\|\mathbf{x}\|_\infty \leq \eta\})$
2: **while** $\|(\zeta f | 2x^{n/2}g + \zeta)\| \geq S$ or f non-invertible mod q
3: $\mathbf{h} \leftarrow [\zeta g + 1]/f \bmod q$
4: $\mathbf{A} \leftarrow (\zeta^*(q-1)h \mid \zeta^*) \bmod 2q$
5: $\mathbf{s} \leftarrow (f \mid \zeta g + 1)^\top$
6: **return** $\mathsf{vk} = \mathbf{A}$ and $\mathsf{sk} = (\mathbf{A}, \mathbf{s})$

Figure 6.3: NTRU KeyGen for $\mathsf{G} + \mathsf{G}$

are the lowest signatures and key sizes provided in the literature for Lyubashevsky's signatures (when using the core-SVP hardness methodology to estimate security). We note that the resulting signature sizes are 20% to 30% lower than the ones from Chapter 4. The asymptotic gain of our signature is observable when comparing the signature sizes with Haetae, as the trade-off is first in its favor but ends up in favor of $\mathsf{G} + \mathsf{G}$ for the higher security level, up to 16% of savings. However, the sum of the public key and the signature sizes is somewhat similar across the three signatures. This is due to the fact that in the non-bimodal setting, a practical optimization due to [DKL+18] consists in truncating the low bits of the public key, at the cost of increasing the verification bound. While such a technique is also implemented in Haetae, its efficiency is relative in this setting, and we chose not to incorporate it in $\mathsf{G} + \mathsf{G}$ for the sake of simplicity.

6.2.3 Optimized NTRU Key Generation Algorithm

We can alternatively use the NTRU-based key generation algorithm from [DDLL13]. In our setting, it is possible to improve it, by relying on the aforementioned technique based on the divisibility of 2 by $(1+x^{n/2})$. This leads to the key generation algorithm presented in Figure 6.3.

The algorithm outputs keys $\mathbf{A}$ and $(\mathbf{A}, \mathbf{s})$ satisfying $\mathbf{As} = \zeta^* q \bmod 2q$ as it holds that $(q-1)hf = (q-1)(\zeta g+1) \bmod 2q$ since $(q-1)$ is even. It that $\zeta\mathbf{As} = 0 \bmod 2q$, and the lattice in needs of smoothing is $\zeta\mathbf{s}\mathcal{R}$ where $\zeta\mathbf{s}^\top = (\zeta f | 2x^{n/2}g + \zeta)$. We then propose two sets of parameters in Table 6.2, for ring dimensions 512 and 1024. The former leads to only around 90 bits of security, but the latter allows to reach NIST security level III. Focusing on the latter, while the sum $|\mathsf{vk}|+|\mathsf{sig}|$ is similar to those of the other schemes, we note that the signature size is further decreased, compared to module $\mathsf{G} + \mathsf{G}$. The resulting signature is 40% smaller than in Section 4.4.5 and 55% smaller than Dilithium.

Target Security	90	180
n	512	1024
q	32257	45569
S	43.73	36.11
KeyGen acceptance rate	0.25	0.5
s	14.32	14.42
σ	626.49	520.75
B	21719.152	40218.387
η	2	1
α	256	2048
BKZ block-size b to break SIS	314 (238)	740 (622)
Best Known Classical bit-cost	91 (69)	216 (181)
Best Known Quantum bit-cost	80 (61)	190 (159)
BKZ block-size b to break LWE	305	616
Best Known Classical bit-cost	89	180
Best Known Quantum bit-cost	78	158
Signature size with rANS	974	1497
Expected public key size	992	2080
Sum	1966	3577

Table 6.2: Parameter Sets for NTRU G + G.

Chapter 7

Conclusion

In this thesis, we explored many aspects of the Fiat-Shamir (with Aborts) paradigm in the lattice setting. By a careful analysis, we discovered that the previous generic security reductions missed subtle issues, which underlines that the paradigm had yet to be fully understood. We fixed the proofs and thankfully recovered similar security loss bounds. This also let us clean up a missing link, in the sense that we proved the first reduction for the Fiat-Shamir with Unbounded Aborts transform. Moreover, our simulator for both aborting and non-aborting transcripts may prove useful for more advanced applications. In particular, any multiparty setting where multiple players have to agree during a first round on a challenge by each outputting their commitments.

In a second time, we aimed at making rejection sampling more efficient. It turned out that the generic technique is already essentially optimal in terms of expected runtime, which means that there was nothing to gain from this side. However, when we consider the specific case of the Lyubashevsky and BLISS identification protocols, we found that using uniform distributions in hyperballs led to the smallest expected norm of the answer, given a target number of repetitions, in both unimodal and bimodal setting. As a side bonus, the rejection condition consists of roughly one Euclidean norm computation and comparison. This is in stark contrast with the rejection condition for the discrete Gaussian distribution, which requires transcendental function evaluations.

However, optimized implementations such as Dilithium rely on more techniques to compress even further the signature size such as the Bai-Galbraith compression [BG14]. For comparison, we proposed HAETAE, which relies on the above findings as well as adaptations of the standard compression techniques. It further shows that relying on hyperball-uniforms is realistic, as we implement such a sampler.

Finally, we showed how to depart from rejection sampling, while retaining similar concrete sizes and improved asymptotic sizes. The $\mathsf{G}+\mathsf{G}$ signature scheme instead relies on adding randomness to the reply to the verifier's challenge, similarly to BLISS, but with more randomness involved. This paves the way for improvement of more advanced protocols, whose efficiency may have been limited due to rejection sampling. These findings raise multiple questions.

1. *Can we improve the hyperball sampler for HAETAE?*

Namely, we measured that hyperball sampling could take up to 80% of the signing process. Any improvement to the sampler would have a huge impact on the performance of HAETAE, as it clearly currently drives the signing cost. One of its downsides is its randomness consumption due to the necessary precision. Alternatively, improving the sampler could also come in the form of reducing the necessary precision, via a refined precision analysis. A third path could be to re-use randomness. Indeed, strategies exist for Dilithium that allow to reuse some part of the randomness (see, e.g. [SW20]). They however cannot be directly applied to our setting, as our distribution has a shape less compatible with coordinate-wise sampling, contrary to hypercubes.

2. What concrete performance could $\mathsf{G}+\mathsf{G}$ *achieve?*

The main concern is the first Gaussian sampling, as the covariance matrix is skewed. Making it competitive with other schemes in terms of efficiency is far from trivial. Moreover, it has been a long standing problem to efficiently protect Gaussian sampling against side-channel attacks via masking. However, we note that if we consider an offline/online scenario, then we could sample the two Gaussian samples beforehand: we would need to sample the second gaussian twice, the first time assuming that $c = 0^\ell$ and the other times assuming that it is $c = 1^\ell$. Indeed, we may not be aware of the challenge yet, but when we obtain it, we can compose the appropriate sample from these two samples, coordinate-wise. Moreover, the leftover randomness is not lost and it can be used for the next signature.

3. What impact can $\mathsf{G}+\mathsf{G}$ *have on advanced protocols?*

The Fiat-Shamir with aborts paradigm is more versatile than just compiling identification protocols into signature schemes. More advanced cryptographic primitives have been built from Dilithium, such as multisignatures [DOTT21], blind signatures [ASY22], advanced proof systems [LNP22], to cite a few. These proof systems rely on the unimodal setting, where one does not have to tweak the public key. Our $\mathsf{G}+\mathsf{G}$ scheme is closer to the bimodal setting, for which it is not immediate to adapt these constructions. This downside is thus inherited by $\mathsf{G}+\mathsf{G}$.

4. Is there a link between $\mathsf{G}+\mathsf{G}$ *and hash-and-sign signatures?*

Hash-and-Sign is another paradigm of signatures, out of which Falcon [FHK+17] was created. Falcon is the other winner of the NIST PQC and the hash-and-sign paradigm is also very successful in terms of further applications. It recently appeared in [CLMQ21] that the first lattice based hash-and-sign signature [GPV08] could be seen as a Fiat-Shamir signature with a very specific instantiation of the hash function. This raises the question of whether there exists an equivalent of $\mathsf{G}+\mathsf{G}$ in the hash-and-sign world, maybe obtained through the same hash function instantiation.

Bibliography

[ABB+17] E. Alkim, N. Bindel, J. Buchmann, Ö. Dagdelen, E. Eaton, G. Gutoski, J. Krämer, and F. Pawlega. Revisiting TESLA in the quantum random oracle model. In *PQCrypto*, 2017.

[ADPS16] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange - A new hope. In T. Holz and S. Savage, editors, *USENIX Security 2016*, pages 327–343. USENIX Association, August 2016.

[AFLT16] M. Abdalla, P.-A. Fouque, V. Lyubashevsky, and M. Tibouchi. Tightly secure signatures from lossy identification schemes. *Journal of Cryptology*, 29(3):597–631, July 2016.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.

[ASY22] S. Agrawal, D. Stehlé, and A. Yadav. Round-optimal lattice-based threshold signatures, revisited. In M. Bojanczyk, E. Merelli, and D. P. Woodruff, editors, *ICALP 2022*, volume 229 of *LIPIcs*, pages 8:1–8:20. Schloss Dagstuhl, July 2022.

[BDF+11] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011.

[BDK+20] S. Bai, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-DILITHIUM round-3 candidate to the NIST post-quantum cryptography standardisation project, 2020. Avalailable at `https://pq-crystals.org/dilithium/`.

[BF11] D. Boneh and D. M. Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 1–16. Springer, Heidelberg, March 2011.

[BG14] S. Bai and S. D. Galbraith. An improved compression technique for signatures based on learning with errors. In J. Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 28–47. Springer, Heidelberg, February 2014.

[BGMN05] F. Barthe, O. Guédon, S. Mendelson, and A. Naor. A probabilistic approach to the geometry of the ℓ_p^n-ball. *Ann Probab*, 2005.

[BGV12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In S. Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, January 2012.

[BLP+13] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *45th ACM STOC*, pages 575–584. ACM Press, June 2013.

[BLR+18] S. Bai, T. Lepoint, A. Roux-Langlois, A. Sakzad, D. Stehlé, and R. Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. *Journal of Cryptology*, 31(2):610–640, April 2018.

[BP98] K. Ball and I. Perissinaki. The subindependence of coordinate slabs in ℓ_p^n balls. *Israel J Math*, 1998.

[BP02] M. Bellare and A. Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 162–177. Springer, Heidelberg, August 2002.

[CCD+23] J. H. Cheon, H. Choe, J. Devevey, T. Güneysu, D. Hong, M. Krausz, G. Land, M. Möller, D. Stehlé, and M. Yi. Haetae: Shorter lattice-based fiat-shamir signatures. Cryptology ePrint Archive, Paper 2023/624, 2023. `https://eprint.iacr.org/2023/624`.

[CL21] A. Chailloux and J. Loyer. Lattice sieving via quantum random walks. In M. Tibouchi and H. Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 63–91. Springer, Heidelberg, December 2021.

[CLMQ21] Y. Chen, A. Lombardi, F. Ma, and W. Quach. Does fiat-shamir require a cryptographic hash function? In T. Malkin and C. Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 334–363, Virtual Event, August 2021. Springer, Heidelberg.

[Dat09] N. Datta. Min-and max-relative entropies and a new entanglement monotone. *T. Inform. Theory*, 2009.

[DDLL13] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal Gaussians. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 40–56. Springer, Heidelberg, August 2013.

[Dev86] L. Devroye. *Non-Uniform random variate generation*. Springer New York, NY, 1986.

[DFMS19] J. Don, S. Fehr, C. Majenz, and C. Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 356–383. Springer, Heidelberg, August 2019.

[DFPS22] J. Devevey, O. Fawzi, A. Passelègue, and D. Stehlé. On rejection sampling in lyubashevsky's signature scheme. In S. Agrawal and D. Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 34–64. Springer, Heidelberg, December 2022.

[DFPS23] J. Devevey, P. Fallahpour, A. Passelègue, and D. Stehlé. A detailed analysis of fiat-shamir with aborts. Cryptology ePrint Archive, Report 2023/245, 2023. `https://eprint.iacr.org/2023/245`.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DKL+18] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR TCHES*, 2018(1):238–268, 2018. `https://tches.iacr.org/index.php/TCHES/article/view/839`.

[DLN+21] J. Devevey, B. Libert, K. Nguyen, T. Peters, and M. Yung. Non-interactive CCA2-secure threshold cryptosystems: Achieving adaptive security in the standard model without pairings. In J. Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 659–690. Springer, Heidelberg, May 2021.

[DLP22] J. Devevey, B. Libert, and T. Peters. Rational modular encoding in the DCR setting: Non-interactive range proofs and paillier-based naor-yung in the standard model. In G. Hanaoka, J. Shikata, and Y. Watanabe, editors, *PKC 2022, Part I*, volume 13177 of *LNCS*, pages 615–646. Springer, Heidelberg, March 2022.

[DOTT21] I. Damgård, C. Orlandi, A. Takahashi, and M. Tibouchi. Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices. In J. Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 99–130. Springer, Heidelberg, May 2021.

[DPSZ12] I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias. Multiparty computation from somewhat homomorphic encryption. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 643–662. Springer, Heidelberg, August 2012.

[DSSS21] J. Devevey, A. Sakzad, D. Stehlé, and R. Steinfeld. On the integer polynomial learning with errors problem. In J. Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 184–214. Springer, Heidelberg, May 2021.

[Duc14] L. Ducas. Accelerating bliss: the geometry of ternary polynomials. Cryptology ePrint Archive, Report 2014/874, 2014. https://eprint.iacr.org/2014/874.

[Dud13] J. Duda. Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding, 2013. ArXiv preprint, available at https://arxiv.org/abs/1311.2540.

[EFG+22] T. Espitau, P.-A. Fouque, F. Gérard, M. Rossi, A. Takahashi, M. Tibouchi, A. Wallet, and Y. Yu. Mitaka: A simpler, parallelizable, maskable variant of falcon. In O. Dunkelman and S. Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 222–253. Springer, Heidelberg, May / June 2022.

[ETWY22] T. Espitau, M. Tibouchi, A. Wallet, and Y. Yu. Shorter hash-and-sign lattice-based signatures. In Y. Dodis and T. Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 245–275. Springer, Heidelberg, August 2022.

[FHK+17] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. Falcon: Fast-Fourier lattice-based compact signatures over NTRU, 2017. Submission to the NIST post-quantum cryptography standardization process.

[FS87] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987.

[GHHM21] A. B. Grilo, K. Hövelmanns, A. Hülsing, and C. Majenz. Tight adaptive reprogramming in the QROM. In M. Tibouchi and H. Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 637–667. Springer, Heidelberg, December 2021.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[HJMR07] P. Harsha, R. Jain, D. McAllester, and J. Radhakrishnan. The communication complexity of correlation. In *CCC*, 2007.

[Kat21] S. Katsumata. A new simple technique to bootstrap various lattice zero-knowledge proofs to QROM secure NIZKs. In T. Malkin and C. Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 580–610, Virtual Event, August 2021. Springer, Heidelberg.

[Kle00] P. N. Klein. Finding the closest lattice vector when it's unusually close. In D. B. Shmoys, editor, *11th SODA*, pages 937–941. ACM-SIAM, January 2000.

[KLS18] E. Kiltz, V. Lyubashevsky, and C. Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 552–586. Springer, Heidelberg, April / May 2018.

[LNP22] V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Y. Dodis and T. Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 71–101. Springer, Heidelberg, August 2022.

[LS15] A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 2015.

[LSS14] A. Langlois, D. Stehlé, and R. Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 239–256. Springer, Heidelberg, May 2014.

[Lyu09] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Heidelberg, December 2009.

[Lyu12] V. Lyubashevsky. Lattice signatures without trapdoors. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Heidelberg, April 2012.

[Lyu16] V. Lyubashevsky. Digital signatures based on the hardness of ideal lattice problems in all rings. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 196–214. Springer, Heidelberg, December 2016.

[MKMS22] J. M. B. Mera, A. Karmakar, T. Marc, and A. Soleimanian. Efficient lattice-based inner-product functional encryption. In G. Hanaoka, J. Shikata, and Y. Watanabe, editors, *PKC 2022, Part II*, volume 13178 of *LNCS*, pages 163–193. Springer, Heidelberg, March 2022.

[MM11] D. Micciancio and P. Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 465–484. Springer, Heidelberg, August 2011.

[MR07] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *J. Comput.*, 2007.

[MV10] D. Micciancio and P. Voulgaris. Faster exponential time algorithms for the shortest vector problem. In M. Charika, editor, *21st SODA*, pages 1468–1480. ACM-SIAM, January 2010.

[Pre17] T. Prest. Sharper bounds in lattice-based cryptography using the Rényi divergence. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017,*

Part I, volume 10624 of *LNCS*, pages 347–374. Springer, Heidelberg, December 2017.

[Reg09] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 2009.

[Ren05] R. Renner. *Security of quantum key distribution.* PhD thesis, ETH Zurich, 2005.

[RW04] R. Renner and S. Wolf. Smooth Rényi entropy and applications. In *ISIT*, 2004.

[Sch91] C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, January 1991.

[SW20] A. Sprenkels and B. Westerbaan. Don't throw your nonces out with the bathwater. Cryptology ePrint Archive, Report 2020/1158, 2020. `https://eprint.iacr.org/2020/1158`.

[vEH14] T. van Erven and P. Harremos. Rényi divergence and Kullback-Leibler divergence. *T. Inform. Theory*, 2014.

[VGS17] A. R. Voelker, J. Gosmann, and T. C. Stewart. Efficiently sampling vectors and coordinates from the n-sphere and n-ball. *Centre for Theoretical Neuroscience-Technical Report*, 01 2017.

[ZXZ18] Z. Zheng, G. Xu, and C. Zhao. Discrete Gaussian measures and new bounds of the smoothing parameter for lattices. Cryptology ePrint Archive, Report 2018/786, 2018. `https://eprint.iacr.org/2018/786`.

List of Figures

List of Tables