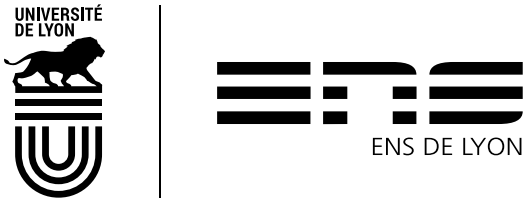
Lattice-based cryptography in a quantum setting : security proofs and attacks

Pouria Fallahpour



N° National de thèse : 2024ENSL0023

THÈSE

en vue de l'obtention du grade de Docteur, délivré par **l'École Normale Supérieure de Lyon**

École Doctorale N° 512
École Doctorale en Informatique et Mathématiques de Lyon

Discipline : Informatique

Soutenue publiquement le 05/07/2024, par

Pouria Fallahpour

Lattice-based cryptography in a quantum setting: security proofs and attacks

Cryptographie fondée sur les réseaux euclidiens dans un cadre quantique : preuves de sécurité et attaques

Devant le jury composé de :

Fouque Pierre-Alain, Professeur des universités, Université Rennes 1	Rapporteur
Tillich Jean-Pierre, Directeur de recherche, INRIA de Paris	Rapporteur
Debris-Alazard Thomas, Chargé de recherche, INRIA École Polytechnique	Examinateur
Kirshanova Elena, Personnalité scientifique,Technological Innovation Institute	Examinatrice
Roux-Langlois Adeline, Directrice de recherche, CNRS Normandie Université	Examinatrice
Stehlé Damien, Professeur des universités, CryptoLab	Examinateur
Villard Gilles, Directeur de recherche, CNRS ENS de Lyon	Directeur de thèse

To Mahshid

Abstract

The rise of quantum machines poses both challenges and opportunities for cryptography. In particular, security proofs may require revisions due to adversaries' quantum capabilities. This thesis presents two contributions in this respect: a positive result and a negative one.

The Fiat-Shamir transform with aborts is one of the major paradigms for designing post-quantum secure signature schemes. Part of this thesis consists of a detailed security analysis of this transform in the quantum random oracle model. It is worth noting that all previous works have neglected subtle details, jeopardizing the correctness of their proofs. Consequently, our security proof stands as the first of its kind that is correct. Moreover, we analyze the runtime and correctness of the signatures obtained from this transform.

The learning with errors (LWE) problem has been extensively utilized to construct cryptographic schemes that are secure against quantum adversaries. A knowledge assumption of the LWE problem states that obliviously sampling an LWE instance, namely without knowing its underlying secret, is hard for all polynomial-time algorithms. One can use this assumption to prove the security of some succinct non-interactive arguments of knowledge (SNARKs). While it seems a hard task for classical algorithms, we demonstrate a quantum polynomial-time oblivious LWE sampler. Consequently, our sampler breaks the security analysis of the mentioned SNARKs in the quantum setting.

Résumé

L'émergence des machines quantiques crée des défis et des opportunités pour la cryptographie. En particulier, les preuves de sécurité doivent être révisées en raison des capacités quantiques des adversaires. Cette thèse propose deux contributions à cet égard : un résultat positif et un résultat négatif.

La transformation de Fiat-Shamir avec des rejets est l'un des principaux paradigmes pour concevoir des schémas de signature post-quantiques. Une partie de cette thèse consiste en une analyse détaillée de cette transformation dans le modèle de l'oracle aléatoire quantique. Tous les travaux précédents proposant une analyse de sécurité de cette transformation ont négligé des détails subtils, compromettant la correction des preuves. Par conséquent, notre preuve de sécurité est la première de son genre à être correcte. De plus, nous analysons le temps d'exécution et la correction des signatures obtenues à partir de cette transformation.

Le problème learning with errors (LWE) a été largement utilisé pour construire des schémas cryptographiques sécurisés contre les adversaires quantiques. Une hypothèse liée à LWE stipule que la génération d'une instance LWE sans connaître son secret est difficile pour tous les algorithmes polynomiaux. On peut utiliser cette hypothèse pour prouver la sécurité de certains arguments de connaissance succints. Bien que cela semble être une tâche difficile pour les algorithmes classiques, nous présentons un algorithme quantique polynomial qui génère des instances LWE sans connaître le secret. Notre algorithme invalide ainsi les analyses de sécurité de ces arguments de connaissance succints dans le contexte quantique.

Acknowledgements

I was accompanied by many people throughout this journey. I would like to express my gratitude and share a few words about them.

First, thanks to Damien who taught me so many things. He was patient and very supportive of my ideas. He is excellent at what he does; focused, motivated, and with clear ideas and plans. I tried my best to learn such a spirit from him. And many thanks to Gilles. He is very thoughtful, and warmly accepted to supervise me. I wish we could do more research together.

I would like to thank Jean-Pierre and Pierre-Alain for taking the time to review my thesis. And more thanks to Adeline, Elena, and Thomas for accepting to be in my jury, as it is a great honor for me.

I am grateful that I had the opportunity to work with wonderful researchers. Thanks to Alain because he is a great teacher and researcher, and I learned many things from him. A special thanks to Thomas. He bore with my ambiguous and jumpy ideas, and gave me many useful comments. He also accepted to present our paper when I did not receive my visa in time.[1] Many thanks to Serge who hosted me for two months at CWI. Although it passed very quickly, it was a pleasant and productive experience for me. I would also like to thank Garazi, Julien, and Yu-Hsuan for the nice research collaboration and their great ideas.

I have always enjoyed working with the QInfo team. A superposition of thanks to Daniel and Omar. Omar was always supportive and kind. I had a delightful introduction to quantum theory during his course, and my enjoyable experience was then completed by being Daniel's TA. I hope that I can work more with them in the future.

I want to thank Dr. Khazaei and Dr. Eghlidos who introduced me to cryptography. My first experience in cryptographic thinking (and more generally TCS) was shaped by their kind and supportive attitudes, and for that I am greateful. I also want to thank ostadha Ahmadpour, Behzadi, and Sharifi. They are one of the main reasons that I decided to continue studying math and TCS.

My most enjoyable memories are the ones with my colleagues and friends at LIP. The journeys, the parties, and the games that we had. My best thanks to Alaa, to Arthur, to Calvin, to Chen, to Emily, to Joël, to Julien, to Fabrice, and to Octavie. I share unique memories with each of them that are beyond this acknowledgements. Thanks to them

[1] Yes, I am mentioning the long visa approval time here because it creates unnecessary difficulties for research!

all.[2] Je vais remercier beaucoup l'equipe de MALIP, en particulier les deux Maries et Chiraz. Chiraz est l'une des personnes les plus gentilles que je connaisse à l'ENS. Merci pour tous les efforts que tu as faits pour moi.

My greatest thanks to my mother. She is supportive, courageous, and a great mother. Many things would be impossible without her constant support during these years. Thank you. And finally, my shiniest and most vivid thanks to Mahshid. You are my friend, as eagle and serpent are Zarathustra's friends, as sophos is philosopher's friend, as joy is Loki's friend, and as Jerry is Tom's friend. What is all this sweet work worth if thou art not mine friend!

[2] Among you there are some who love riddles: 35 55? What dwells beneath Zürichsee?

Contents

Chapter 1

Introduction

1.1 Cryptography

Cryptography, in its most general sense, is the scientific discipline for constructing schemes that allow to securely manipulate information. An old example is the communication between two individuals while ensuring that no third eavesdropper party can understand the relayed messages. This protective measure is commonly referred to as encryption. Numerous attempts were made by many to construct secure encryptions schemes. However, the precise definition of what constituted security was ambiguous. The devised schemes were initially considered secure until subsequent efforts were made to break them. It was not until the seminal work of Shannon [Sha49] that cryptography was raised as a form of science where mathematics was hugely incorporated in its body. Shannon defined the notion of perfect secrecy for communication channels. Despite its innovation, Shannon's approach (known as information-theoretical) encountered significant challenges due its very hard-to-achieve conditions of perfect secrecy.

The celeberated work of Diffie and Hellman [DH76] paved the way to establish a complexity-theoretical foundation for cryptography. They could construct a secure encryption scheme based on the (conjectured) computational hardness of a number-theoretic problem. Besides the fact that the computational hardness seems a plausible assumption, the structure of the problem allowed the expansion of the cryptographic schemes to various functionalities with more advanced security properties. This success was soon followed by the RSA cryptosystem [RSA78]. In all mentioned works, the security definitions were mostly ad-hoc, lacking a precise and conclusive framework. This last piece of puzzle was solved by the seminal work of Goldwasser and Micali [GM82] in which they defined a flexible and formal definition for secure communication. Their work, together with [DH76, RSA78], laid a solid foundation to formalize and expand cryptography into much more advanced notions of security. This foundation is what we refer to as modern cryptography, and we begin by introducing some of its basic tools.

Syntax

A cryptographic scheme is defined as a tuple of algorithms with a set of specifications. Take the example of digital signatures. Let $\lambda \in \mathbb{N}^+$. A digital signature scheme consists of

a tuple of probabilistic polynomial-time algorithms $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Ver})$ with the following specifications:

- $\mathsf{KeyGen}(1^\lambda) \to (vk, sk)$: it takes a parameter λ and outputs a secret key sk and a verification key vk;
- $\mathsf{Sign}(sk, \mu) \to \sigma$: it takes a secret key sk and a message μ, and outputs a signature σ;
- $\mathsf{Ver}(vk, (\mu, \sigma)) \to b \in \{0, 1\}$: it takes a verification key vk and a message-signature pair (μ, σ), and ouputs a bit representing rejecting or accepting.

The parameter λ, known as the security parameter, allows to parameterize the constructions. The size of the parameters vk, sk, and ct are measured as a function of λ as well as the runtime of the algorithms. In practice, the quantity of λ and the description of all algorithms are public, only the keys can remain secret.

The above syntax captures real-world signatures on papers. First, the algorithm KeyGen is run by a party for some appropriately chosen security parameter λ, then the party broadcasts vk to everyone (for instance by writing it on a public bulletin) while keeps sk for herself/himself. Then the owner of sk can sign μ in a way that anyone, by having vk, can verify whether the signature is authentic or not.

The syntax merely captures the formal aspects of a real-world scenario, and not the correctness nor the security. For this system to actually work, a message signed by the true algorithm Sign and the secret key sk must be valid. More precisely, the algorithms must satisfy the following property: for every message μ, it holds that

$$\mathbb{P}_{(vk,sk)\leftarrow\mathsf{KeyGen}(1^\lambda)}[\mathsf{Ver}(vk, \mathsf{Sign}(sk, \mu)) = 1] = 1 \ .$$

Cryptographic game

What could go wrong in the presence of an adversary? The security definition models the adversarial behaviour; it encompasses properties such that if the scheme satisfies them, then it is secure against the adversarial behaviour. The appropriate definitions are rooted from paractical concerns and abstract analysis. One may also be interested in how various definitions imply or are separated from each other.

In the so-called game-based approach, the security is defined as a game between a challenger (modeling an honest party) and an adversarial entity. The adversary is allowed by the game to ask particular types of questions to the challenger, and the challenger responds honestly. After the game ends, an assessment determines whether the adversary wins or loses. The security requires that any efficient adversary loses. We refer to this game as cryptographic or security game.

As an example, we explain one of the principal security definitions for signature schemes. In the beginning of the game, the challenger generates a pair $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and sends vk to the adversary. During the game, the adversary is allowed to ask the following type of questions: it chooses an arbitrary message μ and ask the challenger to sign it. The challenger responds honestly. The goal of the adversary is to find a message-signature pair (μ^*, σ^*) such that $\mathsf{Ver}(vk, (\mu^*, \sigma^*)) = 1$. The message μ^* cannot be

one of the previously asked questions. This is called Existential-Unforgeability Chosen-Message Attacks (UF-CMA) security game. We say that a signature scheme is UF-CMA-secure if every polynomial-time (in λ) adversary cannot win with a non-negligible probability in λ.

Cryptographic assumption

The security of a cryptographic scheme, i.e., the adversary does not win the corresponding cryptographic game with significant probability, is mostly established via a computational reduction to a simple and well-studied computational problem. More explicitly, if $\mathcal{A}$ wins, one can turn it (in time polynomial in the security parameter λ) into an algorithm that solves the computational problem.

In practice, there are a few computational problems that are conjectured be hard and at the same time can be massaged into cryptographic constructions. Let us briefly explain this idea with the example of Lamport signature scheme [Lam10]. We consider the instantiation of the scheme based on the the Discrete Logarithm (DLog) problem defined as follows. Let $\mathbb{G}$ be a cyclic group of order p with a generator g. Given g^x where x is sampled uniformly from $\mathbb{Z}/p\mathbb{Z}$, find x. We say that the DLog assumption holds for $\mathbb{G}$ if no polynomial-time (in the security parameter λ which is usually set as $\log |G|$) algorithm can solve this problem with non-negligible probability. In the Lamport signature scheme, the message space is set to be $\{0,1\}^k$ for some positive integer k. For the sake of simplicity, we set $k = 1$. The secret key consists of two uniformly sampled elements (x_0, x_1) from $\mathbb{Z}/p\mathbb{Z}$. The verification key is then evaluated as (z_0, z_1) where $z_i = g^{x_i}$. One can sign a message $\mu \in \{0,1\}$ by outputting x_μ. For the verification, it suffices to check whether the equality $g^{x_\mu} = z_\mu$ holds or not, and accept if it holds.

This scheme is One-Time (OT) secure under the DLog assumption. The OT security is similar to the UF-CMA security with the exception that the adversary is restricted to one query. We sketch the proof as follows. Assume that there exists a polynomial-time adversary breaking the OT security of the Lamport signature when $k = 1$. Let z be an instance of the DLog problem. To solve z, one can trick the adversary in the following way. First, sample a uniform element x from $\mathbb{Z}/p\mathbb{Z}$ and a uniform bit $b \in \{0,1\}$. Then, set

$$vk = \begin{cases} (z, g^x) & \text{if } b = 0 \ , \\ (g^x, z) & \text{if } b = 1 \ . \end{cases}$$

The distribution of vk is correctly chosen according to the Lamport's construction. However, here the discrete logarithm of z is not known while it is known in the construction. This may raise some issues. On the negative side, if $b = 0$ and the adversary asks for a signature of the message $\mu = 0$, then one cannot output a correct signature without knowing the discrete logarithm of z. Similar failure occurs when $b = \mu = 1$. On the positive side, if for instance $b = 0$ and $\mu = 1$, then x would be a valid signature. In this case, the adversary wins if it generates a valid signature for $\mu^* = 0$, i.e., the discrete logarithm of z. Note that the probability of $b = \mu$ is $1/2$ since the choice of μ is independent of b. Therefore, if the adversary wins with a non-negligible probability $\nu(\lambda)$, one can solve z with probability $\nu(\lambda)/2$, which is still non-negligible. This contradicts the DLog assumption.

Random oracle model

Some cryptographic constructions may use a function that is publicly accessible to all parties including the adversary. An example is a hash function such as $\mathsf{SHA3} : \{0,1\}^* \to \{0,1\}^m$. It is designed to satisfy a key property: the output of the function on an arbitrary input must look uniformly random. Consequently, it is computationally difficult for practical algorithms to find a collision, namely, two strings x and x' such that $\mathsf{SHA3}(x) = \mathsf{SHA3}(x')$. Note that such a collision necessarily exists. In practice, a description of $\mathsf{SHA3}$ is public. An adversary can use the description of H to evaluate it by itself.

As mentioned above, $\mathsf{SHA3}$ is designed to look uniformly random on its evaluations. Although this is not correct (the function is deterministic and thus not randomized), such a property is very helpful in security analyses. In a widely used paradigm, known as the Random Oracle Model (ROM), the hash function is modeled as a uniformly sampled function. The parties can issue oracle queries to evaluate the function.

> *Random Oracle Model: In the cryptographic game, all parties have oracle access to a uniformly sampled function* $H : \{0,1\}^* \to \{0,1\}^m$ *for some integer* $m \geq 1$.

Note that H cannot be described nor sampled. In fact, no Turing machine can evaluate H since every input is mapped to a uniformly sampled output. The Turing machine must encode infinite amount of information to be able to evaluate H.

In a standard method, the random oracle can be simulated *on the fly*, also known as the lazy sampling method. Every time the oracle is queried on a fresh input, it returns a fresh uniformly sampled element from $\{0,1\}^m$; if it is queried twice on the same input, it outputs the same value.

The random oracle model was first used by [FS86, BR93] to argue the security of cryptographic protocols. The constructions designed in this model often enjoy more optimized features, such as runtime or compactness. The Fiat-Shamir [FS86] and Fujisaki-Okamoto [FO99] transforms are two well-known examples of how hash functions help to optimize cryptographic designs. Moreover, some protocols are not known to exist from certain type of assumptions (falsifiable) in the standard model, while they exist in the ROM (see, e.g., [Mic00]).

1.2 Quantum Computation

The idea of building quantum computers began in 1980's by Manin and Feynman. They put forward the question of how quantum models of computation look like. The goal was to simulate quantum phenomena, which was beyond the reach of classical (based on integrated circuits) computers. The quantum superposition of matter, roughly speaking, provides the capability of having many classical states at the same time which allows manipulating larger amount of information in smaller memory/time capacity compared to classical computers. The intuition is that such computers are sufficiently strong to simulate the huge space of possible states of a quantum system. Below, we briefly recall basic tools of quantum computation.

The state of a quantum system is represented using a unit vector (up to scalar multiplication) in a Hilbert space (mostly a $\mathbb{C}$-vector space). A special case is a quantum

system, such as a photon, whose state belongs to $\mathbb{C}^2$. The *computational basis* of $\mathbb{C}^2$ is the orthonormal basis $|0\rangle := (1\ 0)^\intercal$ and $|1\rangle := (0\ 1)^\intercal$. Any state is a linear combination $\alpha|0\rangle + \beta|1\rangle$ for some $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$. Such a state is called a qubit (quantum bit) since it can be regarded as a superposition of classical bits 0 and 1, being represented by $|0\rangle$ and $|1\rangle$, respectively. The joint state of a many-body quantum system is always a unit vector in the tensor product space obtained by tensoring the Hilbert spaces of all partial systems. Consequently, the state of n qubits is a unit vector in the span of $\{|x_1\rangle \otimes \cdots \otimes |x_n\rangle\}_{x_1,\cdots,x_n \in \{0,1\}}$. For the sake of convenience, we sometimes drop the tensor product notation, or we define $|x_1, \cdots, x_n\rangle := |x_1\rangle \otimes \cdots \otimes |x_n\rangle$. We also let $\langle x|$ denote the complex conjugate of $|x\rangle$.

The unitarity principle states that the evolution of an isolated quantum system is reversible. The evolution $|\psi\rangle \mapsto |\psi'\rangle$ can be represented by a unitary map, acting on the corresponding Hilbert space of the system. As long as the system is isolated and is not measured, the evolution is reversible. To revert the system, the inverse of $\mathbf{U}$, denoted by $\mathbf{U}^\dagger$, is applied. A major difference of quantum theory of physics with its classical counterpart occurs when one decides to measure/observe the system. The unitarity principle does not hold in this case since the system is not isolated. Measuring a quantum system possibly changes the state of the system. A quantum measurement can be formalized by a set of projections over the Hilbert space that sum up to identity. Then by measuring a quantum system with respect to a given set of projections, the post-measurement state (up to normalization) can be computed by projecting the current state of the system under a randomly chosen projection in the set. The probability that a given projection is applied is proportional to the norm of the image of the state under the projection. For instance by measuring $\alpha|0\rangle + \beta|1\rangle$ with the two projections into the spans of $|0\rangle$ and $|1\rangle$, one observes $|0\rangle$ as the outcome with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$.

To build a simple prototype of a quantum computer, one can implement a physical apparatus in the lab that simulates very basic unitary and projective operators. If the set of the unitary operators are chosen to be universal (e.g., see [NC11]), then one can simulate any arbitrary quantum evolution. Once having this machinery, we can encode the binary description of any computational problem into quantum states and look for a quantum evolution that maps this state into a one that encodes the solution of the problem. If a solution exists, such a quantum evolution necessarily exists.

The quantum circuit model of computation is widely used to measure the runtime of quantum algorithms. A quantum circuit operates on some number of qubits, using a universal set of one-qubit or two-qubit unitary gates (maps) and one-qubit quantum measurements. The measurements consist of projections onto the spans of the computational basis vectors. The outcomes of some of the final measurements are flagged as the output of the algorithm. An algorithm may use ancilla qubits, i.e., extra quantum registers initialized to $|0\rangle$. We say that a sequence of quantum circuits $(Q_i)_i$ is Quantum Polynomial-Time (QPT) if there exists a classical deterministic polynomial-time algorithm that takes i in unary as input and outputs the description of Q_i with gates and measurements.

Assume that a Boolean-circuit implementation of a function $f : \{0,1\}^n \to \{0,1\}^m$ is given. What is an appropriate model for quantum evaluation of f? One model that is widely used is a unitary map $\mathbf{U}_f$ that acts on the computational basis as follows:

$$\forall x \in \{0,1\}^n, y \in \{0,1\}^m : \ |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle \ ,$$

which can be uniquely extended to the whole space by linearity. The above unitary allows to evaluate f over a superposition of a wide range of inputs. Moreover, given the Boolean circuit description, one can implement $\mathbf{U}_f$ using a basic set of one-qubit or two-qubit unitaries (see, e.g., [NC11]). The number of unitaries is linear in the size of the Boolean circuit. Another model of quantum evaluation of f is the so-called phase oracle. When $m = 1$, one can use $\mathbf{U}_f$ to implement a unitary acting on the computational basis as follows:

$$\forall x \in \{0,1\}^n : \ |x\rangle \mapsto (-1)^{f(x)} |x\rangle \ ,$$

which can be extended by linearity to the whole space. The choice of the oracle depends on the application. In some cases the phase oracle is more helpful such as Grover's algorithm [Gro96], while in some other cases the oracle $\mathbf{U}_f$ is more frequent such as in algorithms designed for solving the Hidden Subgroup Problem (see, e.g, [NC11]).

The ability of building a superposition of inputs is crucial in quantum computation. Let $\mathbf{H}$ be the Hadamard transform with the matrix representation in the computational basis as follows:

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

It is a unitary map. The reader may note that by applying it to the computational basis, we obtain

$$\mathbf{H}\,|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad \text{and} \quad \mathbf{H}\,|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \ .$$

1.3 Cryptographic Impacts of Quantum Computation

Some computational assumptions that have been used to analyze the security of cryptographic schemes break against quantum adversaries. Shor's algorithm [Sho94, Sho97] provided a first example by solving the DLog problem in any cyclic group in polynomial-time (with respect to the size of the instance). So far, no classical polynomial-time algorithm is known that solves DLog in arbitrary cyclic groups. For prime order subgroups of the multiplicative group of finite fields with large characteristics, the best known algorithms achieve superpolynomial runtime (see, e.g., [Gor93,Mat03,JLSV06,BP14,BGK15]). The DLog problem has been extensively used in the classical-setting cryptography. As a consequence of Shor's algorithm, all constructions based on DLog are insecure against quantum polynomial-time adversaries.

To briefly explain what is pivotal in Shor's idea, we use the Deutsch-Josza algorithm [DJ92]. Assume that $f : \{0,1\} \to \{0,1\}$ is a function that is either constant or balanced, namely, either $f(0) = f(1)$ or $f(0) \neq f(1)$, respectively. The Deutsch-Josza problem asks, given oracle access to f, to tell which one is the case. Clearly, any classical algorithm requires 2 queries to the oracle. However, one can use a single quantum oracle query to the phase oracle of f to solve the problem due to Deutsch and Josza [DJ92]. The principal idea is to evaluate f over all possible inputs using the phase oracle. The

obtained state is called the phase state of f. To do so, it suffices to first prepare the following superposition of all inputs:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} ,$$

and then apply the phase oracle of f to obtain

$$\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(0)}|1\rangle}{\sqrt{2}} .$$

Note that the phase state is equal to $(|0\rangle + |1\rangle)/\sqrt{2}$ if f is constant, and $(|0\rangle - |1\rangle)/\sqrt{2}$ if f is balanced. These two states are orthogonal, therefore, the measurement that consists of projections into the spans of these states can perfectly tell them apart. This solves the problem with one quantum oracle query to f.

The measurement above is performed using the projections into spans of the states $\mathbf{H}|0\rangle$ and $\mathbf{H}|1\rangle$. This is known as the *Fourier basis* as opposed to the computational basis $\{|0\rangle, |1\rangle\}$. With this terminology, the Deutsch-Josza algorithm is simplified as follows. First, build a superposition of all inputs using the Hadamard/Fourier transform. Second, apply the classical function in superposition. Finally, analyze or measure the final state in the Fourier basis.

The power of Fourier: the phase state of a classical function reveals useful information when analyzed in the Fourier basis.

Shor's algorithm for solving DLog is more involved and its details are beyond the scope of this thesis. However, the principal idea follows similar techniques to the Deutsch-Josza algorithm. The role of Fourier analysis on a particular set of states constructed out of the DLog instance is significant.

In addition to the promising directions for designing more efficient algorithms, quantum computers have some drawbacks for cryptography; most notably for security reductions. Recall that in a cryptographic game, an adversary makes queries to the challenger during the game. The security reduction that transforms a winning adversary into a solver for the underlying computational problem can observe the queries of the adversary. The success of the reduction may depend on the values on which the adversary makes queries. Many schemes require this type of reductions for their security proofs (we will provide a concrete example later).

With classical adversaries, the reduction can read, measure, or copy the queries of the adversary, or simply can inspect it during the execution without changing its behaviour. However, extracting information from a quantum adversary is quite challenging. The first issue is the quantum measurement effect. As discussed above, measuring a quantum state, being the query or being the state of the adversary, potentially changes the state. This can destroy the correct execution of the adversary such that there is no more promise that the adversary wins despite being measured.

Quantum measurement effect: measuring a quantum state changes the state itself.

The second issue is raised when one attempts to copy the state of the adversary. There is no quantum algorithm, that given an arbitrary quantum state, generates a copy of it. Consequently, the security reduction cannot make multiple copies of the state of the adversary and measure some of them to gain information, while keeping one copy untouched.

No cloning theorem: arbitrary quantum states cannot be cloned.

More issues could be encountered by the security reduction in the presence of quantum adversaries. There is no general lifting theorem saying that if a scheme is secure (against classical adversaries) under a quantum-secure assumption, then it is also secure against quantum adversaries. On the contrary, there exist counterexamples [BDF+11, YZ21]. For these reasons, the security analyses are mostly case-dependent.

Quantum random oracle model

The description of the hash function $H : \{0,1\}^* \to \{0,1\}^m$ is public. Therefore, a quantum adversary is able to evaluate it in a superposition of a wide range of inputs. More precisely, an adversary can evaluate H as follows. It first builds a state of the form

$$\sum_{x \in X, y \in Y} \alpha_{x,y} |x\rangle |y\rangle \ ,$$

where $X \subset \{0,1\}^*$ and $Y \subseteq \{0,1\}^m$. Then it applies the quantum evaluation of H, say $\mathbf{U}_H$, to obtain

$$\sum_{x \in X, y \in Y} \alpha_{x,y} |x\rangle |y \oplus H(x)\rangle \ .$$

Note that for a polynomial-time adversary, the quantity of $\log |X|$ must be polynomial. This stems from the fact that the required number of qubits to represent the elements of X must be polynomial. Without loss of generality, one can assume that X is $\{0,1\}^n$ for a sufficiently large n.

In the Quantum Random Oracle Model (QROM), the parties are granted quantum access to a uniformly sampled function. The description of the function is not public.

> *Quantum Random Oracle Model: In the cryptographic game, all parties have quantum oracle access to a uniformly sampled function $H : \{0,1\}^n \to \{0,1\}^m$ for some integers $m, n \geq 1$.*

As opposed to the classical case, the random oracle can be (inefficiently) described or sampled. Therefore, every oracle query can be answered by the quantum evaluation of the sampled function.

In many scenarios, an efficient simulation of the random oracle similar to the on-the-fly sampling is required. Various techniques have been introduced for this purpose, such as using k-wise independent hash functions [Zha12b] or compressed oracles [Zha19]. We do not discuss the details since it is beyond the scope of this thesis.

1.4 A Conjectural Quantum-Hard Problem

Shor's algorithm gave a motivation to study computational problems that are hard-to-solve for QPT algorithms and are simultaneously useful for cryptography. There are a few candidates based on Euclidean lattices, error-correcting codes, elliptic-curve isognies, and multivariate polynomials. In this thesis, we only discuss the lattice-based ones.

A Euclidean lattice L is a discrete subgroup of $\mathbb{R}^n$. There always exists a basis $\mathbf{B} = (\mathbf{b}_1 \mid \cdots \mid \mathbf{b}_k)$ with $k \leq n$ of linearly independent vectors such that L is the integer span of $\{\mathbf{b}_i\}_i$. In other words, we have $L = \mathbf{B}\mathbb{Z}^k$. The choice of basis is not unique. In fact, for every integer matrix $\mathbf{U}$ with determinant 1, the two bases $\mathbf{B}$ and $\mathbf{BU}$ define the same lattice. For a lattice L, we let $\lambda_1(L)$ denote the minimum ℓ_2-norm of the elements in $L \setminus \{\mathbf{0}\}$. The Approximate Decisional Shortest Vector Problem (GapSVP) for the parameter $\gamma > 0$ is defined as follows:

> GapSVP_γ: *Given a lattice L and a distance threshold $r > 0$, with the promise that either $\lambda_1(L) < r$ or $\lambda_1(L) > \gamma r$, decide which one is the case.*

Some results about this problem have been discovered depending on the quantity of γ as a function of the dimension of the lattice, i.e. n. The main hardness result states that $\mathsf{GapSVP}_{O(1)}$ is NP-hard under randomized reductions [Ajt98, CN98, Mic98, Kho03, Kho05]. There are barriers to extend the hardness to larger quantities of γ. For instance, when $\gamma = \sqrt{n}$, the problem falls in co-NP (see, e.g., [AR05]) and it is unlikely to be NP-hard. On the other hand, the best known polynomial-time algorithms succeed when γ is roughly as large as $\exp(n \log\log n / \log n)$ [LLL82, Sch87, AKS01]. Taking into account both the hardness and the algorithmic results, the problem is conjectured to be quantum worst-case hard when γ is chosen to be polynomial in n.

GapSVP and several other lattice-based problems establish a foundation for post-quantum cryptography where schemes are devised to be secure against quantum adversaries. Particularly, the Learning With Errors (LWE) problem, which was introduced by [Reg09], is widely used in cryptographic designs.

> LWE: *Let m, n, q be positive integers, and χ be a distribution over $\mathbb{Z}/q\mathbb{Z}$. Let $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, $\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n$ be sampled uniformly and $\mathbf{e} \in (\mathbb{Z}/q\mathbb{Z})^m$ from $\chi^{\otimes m}$. The search $\mathsf{LWE}_{m,n,q,\chi}$ problem asks to find $\mathbf{s}$ and $\mathbf{e}$ given the pair $(\mathbf{A}, \mathbf{As} + \mathbf{e})$.*

The vectors $\mathbf{s}$ and $\mathbf{e}$ are respectively called the secret and the noise.

In most cases, the dimension m is polynomial in a security parameter λ and the modulus q ranges from polynomial to exponential in λ; the distribution χ is often set as an integer Gaussian of standard deviation parameter $\sigma \in [\Omega(\sqrt{n}), O(q/\sqrt{n})]$ that is folded modulo q, which will be subsequently denoted by $\vartheta_{\sigma,q}$. We have $\vartheta_{\sigma,q}(e) = \sum_{k\in\mathbb{Z}} \exp(-|e + qk|^2/\sigma^2)$ for all $e \in \mathbb{Z}$, up to a normalization factor. For sufficiently small values of σ, for example $\sigma = O(q^{(m-n)/m}/\sqrt{\lambda})$, one can show that the valid LWE instances are sparse in $(\mathbb{Z}/q\mathbb{Z})^m$: a uniformly sampled vector $\mathbf{b}$ is unlikely a valid LWE instance.

The LWE problem has profound connections with Euclidean lattices. The quantum hardness of the LWE problem for various distributions of the noise and the secret has been extensively studied (see, e.g., [Reg09, Pei09, GKPV10, MM11, BLP+13, BD20]) and it is known that LWE is no easier than worst-case GapSVP for certain parameters [Reg09].

More precisely, it was shown in [Reg09] that the LWE problem with parameters $m, n, q, \vartheta_{\sigma,q}$ is quantumly at least as hard as worst-case GapSVP_γ in dimension n when $\sigma \geq \Omega(\sqrt{n})$ and $\gamma = \widetilde{O}(qn/\sigma)$. The reduction was later "dequantized" by [Pei09] under the condition that $q \geq 2^{\Omega(n)}$. Moreover, the authors of [BLP+13] showed that the LWE problem with parameters $m, n^2, q, \vartheta_{\sigma,q}$ is classically at least as hard as worst-case GapSVP in dimension n when σ is larger than some polynomial in n. From an algorithmic viewpoint, there is no known solver for LWE with runtime lower than $\exp\Big(\Omega(n \log n \log q / \log^2(q/\sigma))\Big)$, when m is polynomially large (see, e.g., [HKM18]).

The subdiscipline of cryptography that mostly exploits the problems related to Euclidean lattices including the LWE problem is known as lattice-based cryptography. The security of most elementary lattice-based schemes, such as signatures and encryptions, translates to the hardness of GapSVP_γ with γ being a low-degree polynomial. Consequently, they are conjectured to be secure against quantum adversaries. In this thesis, we do not use lattices anymore; the LWE problem and its black-box hardness results are sufficient to present our contributions.

Hardness of oblivious sampling

Recall that in the LWE problem an instance consists of a pair $(\mathbf{A}, \mathbf{b}) \in (\mathbb{Z}/q\mathbb{Z})^{m\times n} \times (\mathbb{Z}/q\mathbb{Z})^m$. A formal definition of LWE samplers is as follows.

> LWE *sampler: A polynomial-time algorithm $\mathcal{S}$, that takes as input a uniform matrix $\mathbf{A}$ and outputs a correctly distributed $\mathbf{b}$:*
>
> $$\mathcal{S}_{m,n,q,\chi} : \ \mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n} \quad \longrightarrow \quad \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \in (\mathbb{Z}/q\mathbb{Z})^m \ .$$

A naive way of sampling an LWE instance is as follows: sample $\mathbf{s}$ and $\mathbf{e}$ with the correct distributions, and then compute $\mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$. However, in this way of sampling the pair $(\mathbf{s}, \mathbf{e})$ is exposed to the sampler. Intuitively, it already knows the underlying secret. How can one mathematically formalize the notion of knowing the underlying secret?

If the sampler is classical, then it is also given a polynomial-length (in n) uniformly random bit-string besides $\mathbf{A}$. In this case, the notion of knowing the underlying secret can be roughly defined as follows. We say that a classical sampler $\mathcal{S}_{m,n,q,\chi}$ knows the underlying secret if there exists a polynomial-time (in n) extractor algorithm, that given the randomness of $\mathcal{S}_{m,n,q,\chi}$, extracts the secret. Note that once the randomness is fixed, the generated instance is fixed. Note that by having the randomness, the extractor can run the sampler or observe and copy the state of the sampler at any step.

A way of sampling LWE instances without knowing the underlying secrets could be carried out by sampling a vector $\mathbf{b} \in (\mathbb{Z}/q\mathbb{Z})^m$ from the uniform distribution and hoping that $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$ for some vectors $\mathbf{s}$ and $\mathbf{e}$, with high probability. The probability is taken over the distributions of $\mathbf{s}$ and $\mathbf{e}$. For the typical parametrizations of the LWE problem in cryptography, the distribution of LWE instances is considerably sparse over the vectors in $(\mathbb{Z}/q\mathbb{Z})^m$ and a correctly sampled one most likely admits a unique witness pair $(\mathbf{s}, \mathbf{e})$. Therefore, such vectors $\mathbf{b}$ are far from the correct distribution of LWE.

To our knowledge, no candidate oblivious sampler for LWE has been proposed. Furthermore, it is conjectured that no such sampler exists. This conjecture has been used to analyze the security of several cryptographic schemes. An early occurrence was [LMSV12], to build a homomorphic encryption scheme. The precise algebraic framework was different and led to a quantum polynomial-time attack in [CDPR16], but the usefulness of the assumption can be explained in the LWE context as follows. Assume a ciphertext corresponds to an LWE instance $\mathbf{b} = \mathbf{A}\mathbf{s}+\mathbf{e}$ belonging to the ciphertext space $(\mathbb{Z}/q\mathbb{Z})^m$, and that the plaintext of a well-formed ciphertext is a function of $\mathbf{s}$ (the matrix $\mathbf{A}$ is publicly known, and could for example be part of the public key). In the context of chosen-ciphertext security, the attacker is allowed to query a decryption oracle on any element in the ciphertexts space to extract useful information. In the scheme, if the query is not a well-formed ciphertext, the challenger will be able to detect it and reply with a failure symbol. The oblivious sampling hardness assumption ensures that if the adversary makes a decryption query on a well-formed $\mathbf{b} = \mathbf{A}\mathbf{s}+\mathbf{e}$, then the reply to the query does not give it anything more than it already knows. The oblivious sampling hardness assumption was used more recently in a series of works building Succinct Non-interactive Arguments of Knowledge (SNARKs) from lattice assumptions [GMNO18, NYI+20, ISW21, SSEK22, CKKK23, GNSV23].

1.5 Fiat-Shamir Transform

We illustrate how a random oracle can be used in cryptography. Our example is centered around constructing secure signature schemes. We consider the UF-CMA security for the signature schemes that can be briefly described via the following game. First, the challenger generates $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and sends vk to the adversary. During the game, the adversary is allowed to ask signatures of arbitrary messages from the challenger. Finally, the adversary wins by outputting a valid message-signature pair for a message that was not queried before.

Let us first recall some background. An interactive proof for a language $L \subseteq \{0,1\}^*$ is a two-party system wherein one party (the prover) is determined to convince the other party (the verifier) that a public instance x belongs to the language. Further, the protocol must satisfy the following properties:

- It is complete, i.e, for every instance $x \in L$, the verifier is convinced;
- It is sound, i.e, for every instance $x \notin L$, the verifier is not convinced;
- It is zero-knowledge, i.e, for every $x \in L$, the verifier will not learn anything beyond the membership $x \in L$.

More precisely, the zero-knowledgeness requires the existence of an algorithm Sim that generates a fake transcript of the whole communication by only using x with the following property: the statistical difference between the distributions of the fake transcript and the honest transcript is negligible in the size of x.

We are interested in the special case of 3-round interactive protocols where the message of the verifier is chosen uniformly at random from a fixed set $\mathcal{C}$ of challenges. This is known as Σ-protocol. See Figure 1.1.

When L is an NP language, the special-soundness of a Σ-protocol is more desirable than the soundness property. It states that by having two successful transcripts (w, c, z)

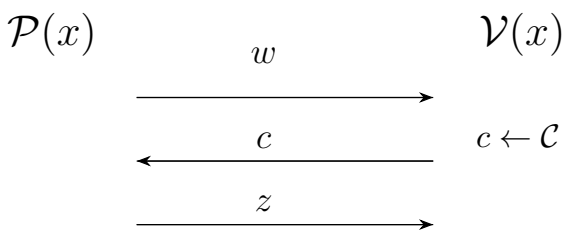


Figure 1.1: The interaction in a Σ-protocol between the prover and the verifier.

and (w, c', z') such that $c \neq c'$, one can find a witness for x. In fact, special-soundness implies soundness. In the following, we consider special-sound protocols.

The Fiat-Shamir transform [FS86] turns a Σ-protocol into a signature scheme, by replacing the challenge of the verifier with hash function evaluations. For signing a message μ, the prover replaces c as $H(w\|\mu)$, runs the protocol, and outputs (w, z) as the signature. To verify a signature, it suffices to check $\mathcal{V}(w, H(w\|\mu), z) = 1$. Intuitively, if H is a random oracle with codomain $\mathcal{C}$, then its outcome has the same distribution as the verifier's challenge. The obtained signature from the Fiat-Shamir transform is indeed UF-CMA secure. The proof in the random oracle model consists of two steps. First, it is proven that the adversary gains roughly no advantage by asking signature queries from the challenger, according to the zero-knowledge property of the Σ-protocol. Second, assuming that the adversary makes no signature queries (without loss of generality due to the first step), it is shown that any successful adversary can also break the special-soundness of the Σ-protocol. We briefly explain the two steps below.

1. When the adversary asks for a signature of a message μ the challenger must return a correct signature of the form (w, z) such that $\mathcal{V}(x, w, H(w\|\mu), z) = 1$. Consider a cheating challenger that signs as follows. It samples a fake transcript (w', c', z') using the zero-knowledge simulator and outputs (w', z') as the signature. Note that the fake transcript does not necessarily satisfy $\mathcal{V}(x, w', H(w'\|\mu), z') = 1$. However, when the hash query $w'\|\mu$ is queried by the adversary, the challenger can simply answer by c'. Note that c' must have a distribution very close to uniform because of the zero-knowledgeness. Therefore, answering the hash query by c' does not significantly change the distribution of the random oracle. The response is most likely consistent with the previous hash queries of the adversary if w' has high entropy from the viewpoint of the adversary (hence unlikely that the adversary queried it before). Based on the zero-knowledgeness property, the fake answer by the challenger is not detectable by the adversary. Therefore, the adversary gains almost no knowledge by signature queries. This technique is known as *reprogramming the random oracle* since it reprograms $H(w'\|\mu) := c'$. Note that this is not possible if H was a fixed hash function: we use the properties of the random oracle model.

2. Assume that the adversary outputs a successful forgery (w^*, z^*) for some message μ^* with probability ε. It must hold that $\mathcal{V}(x, w^*, H(w^*\|\mu^*), z^*) = 1$. Intuitively, the adversary must have queried the hash function on input $w^*\|\mu^*$, otherwise since the outcome of H is uniformly sampled, it is highly unlikely the forgery gets accepted by the verifier. Now, we rewind the state of the adversary back to the step exactly before the hash query $w^*\|\mu^*$. We let the adversary run again but with

reprogramming $H(w^*\|\mu^*) := c'$ where c' is uniformly sampled independent of c^*. The adversary must win with the same success probability since the distribution of c' does not change the distribution of H. Due to a probabilistic argument known as the *rewinding lemma* (see, e.g., [BS23]), the adversary outputs a forgery of the form (w^*, z') for the same message μ^* with probability at least $\varepsilon^2 - \varepsilon/N$ where N is the size of the challenge space (the range of H). Note that if the challenge space is sufficiently large, this breaks the special-soundness since $c^* \neq c'$ and both transcripts get accepted by the verifier.

However, one encounters multiple issues when analyzing the security in the QROM.

- In the first part of the reduction above, the reprogramming cannot be carried out as before due to the fact that the reduction cannot observe the value of the input without possibly destroying it.
- In the second part, one cannot resort to the rewinding lemma. More precisely, after one execution of the adversary and obtaining the first successful forgery, the second execution is not guaranteed to succeed anymore. This is because the adversary may perform quantum measurements for outputting the forgery. These measurements possibly destroy the reversibility of the execution. For instance, the adversary may use an auxiliary quantum state, which is not cloneable in general, such that after the measurement cannot be restored. Therefore, it is not clear how one can run the adversary twice and obtain two accepting transcripts.

These issues have been addressed by [Unr17, KLS18, DFMS19, LZ19, GHHM21]. The authors of [GHHM21] showed how to successfully perform the reprogramming of the random oracle for the first part of the reduction. They showed that the same naive reprogramming of the random oracle, as in the classical case, works via a more sophisticated analysis. Their approach exploits the compressed oracle technique that was first developed by [Zha19]. The authors of [DFMS19] showed how to adaptively reprogram a quantum random oracle at one input. More precisely, for any adversary $\mathcal{A}$ that makes quantum queries to the random oracle H and outputs a pair (w, z) such that $\mathcal{V}(w, H(w\|\mu), z) = 1$, they construct the following extractor: it measures one of the queries of $\mathcal{A}$ randomly to obtain $w'\|\mu'$, reprograms $H(w'\|\mu')$ to c' for some uniformly sampled c', and outputs (w', z') that likely satisfies $\mathcal{V}(w', c', z') = 1$. This measure-and-reprogram technique can be used to perform the second part of the reduction. We note that the approach of [KLS18] has some flaws, which will be discussed later.

Fiat-Shamir with aborts

By applying the Fiat-Shamir transform to an *aborting* identification scheme, where the prover can reject or stop responding by outputting a special symbol $\perp$, one also obtains a signature scheme. However, the correctness of the signature is no longer guaranteed. To amplify the correctness, one can simply repeat the protocol many times and hope for finding a non-aborting transcript with higher probability. Then the transform replaces the challenge of the non-aborting transcript with a hash function evaluation. We refer to this modified transform as Fiat-Shamir with Aborts (FSwA).

FSwA has been used to construct post-quantum secure signature schemes. In [Lyu09, Lyu12], Lyubashevsky proposed a lattice-based signature scheme that can be regarded as the first application of FSwA. The underlying interactive proof system has a non-negligible probability of aborting. Aborting allows to make the signature distribution independent of the signing key and is necessary to avoid attacks against the signature schemes (see [ASY22, Section 4.1]).

There are two variants of FSwA: with bounded or with unbounded number of aborts. In the Fiat-Shamir with Bounded Aborts (FSwBA), a parameter B restricts the number of repetitions. The signing algorithm stops after B number of repetitions, even if the signing algorithm does not succeed to find a non-aborting transcript. With FSwBA, the runtime analysis is trivial. In the security proof, the upper bound on the number of iterations is technically convenient as it provides a bound on how many random oracle values are being programmed by the challenger, which eases the analysis of the random oracle programming impact on the adversary's view. The most detailed security analyses are provided in [AFLT16] for the ROM, and in [KLS18] for the QROM. An alternative proof strategy in the QROM is suggested in [GHHM21], but not detailed. In the Fiat-Shamir with Unbounded Aborts (FSwUA), the repetition does not stop until a non-aborting transcript is obtained. This variant is more desirable in practice since it is simpler to implement and it enjoys better correctness (see, e.g, [DKL+18]). However, it is more difficult to analyze, as arbitrarily many hash values may be reprogrammed by the challenger in the security proof.

1.6 Our Results

We provide two independent results in the scope of security proofs and attacks in the quantum setting. In both cases, we leverage the power of quantum tools and show how they can be manipulated to obtain positive and negative results.

Oblivous LWE sampling

One encounters multiple issues when studying oblivious samplers in the quantum setting. The first one is to generalize the definition of oblivious sampling to include quantum samplers. This is not a trivial task. In the classical setting, the generated instance is determined by the randomness. When the computation is repeated with the same randomness, it consistently produces identical results. In a quantum sampler, each execution potentially yields a different instance due to internal quantum measurements. The randomness induced by quantum measurements cannot be fully determined, as in the classical case, by a fixed bit-string. On the other hand, observing the sampler using quantum measurements possibly destroys its state and consequently its outcome distribution, changing the sampler to something different. These issues prompt a question: what specific information must be given to the extractor, or what kind of operations the extractor is allowed to perform on the sampler?

As a first result, we extend the definition of oblivious sampling to quantum algorithms. A prior definition was put forward in [LMZ23]. We propose an alternative definition that, in our opinion, better models what an extractor should be allowed. For the class of quantum algorithms that first perform a unitary and then a measurement, we show that

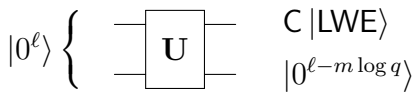


Figure 1.2: The circuit of the oblivious sampler in Theorem 1.

these two definitions are equivalent. We believe that our new definition is valuable as it provides further insight on oblivious sampling.

Our main result is a polynomial-time quantum LWE sampler that we prove oblivious under the assumption that LWE is intractable, under very mild parameter restrictions.

Theorem 1. *Let $m \geq n \geq 1$ and $q \geq 3$ be integers and $\sigma \geq 2$ be a real number. The parameters m, n, q, σ are functions of the security parameter λ with $m, \log q \leq \mathsf{poly}(\lambda)$ and q prime. Assume that the parameters satisfy the following conditions:*

$$m \geq n\sigma \cdot \omega(\log \lambda) \quad \text{and} \quad 2 \leq \sigma \leq \frac{q}{\sqrt{8m \ln q}} \ .$$

Then there exists a $\mathsf{poly}(\lambda)$-time quantum oblivious $\mathsf{LWE}_{m,n,q,\vartheta_{\sigma,q}}$ instance sampler, under the assumption that $\mathsf{LWE}_{m,n,q,\vartheta_{\sigma,q}}$ is hard.

The proof technique and result are quite flexible. For example, the secret $\mathbf{s}$ can have any efficiently sampleable distribution. Moreover, only some mild conditions restrict the shape of the matrix $\mathbf{A}$ and the error $\mathbf{e}$. This allows us to extend Theorem 1 to oblivious samplers for LWE with more algebraic structures such as to the module version of LWE [BGV12, LS15]. Also, we will show that obliviousness is preserved through randomized Karp reductions. Then, by using reductions from LWE with a parametrization satisfying the conditions of Theorem 1 to LWE with a second parametrization, we obtain the existence of an efficient quantum oblivious LWE sampler for the second parametrization, under the assumed hardness of LWE for the first parametrization. We can notably throw away superfluous samples (i.e., decrease m), take an arbitrary arithmetic shape for q and choose larger values for σ, by using modulus-dimension switching [BLP+13].

The oblivious sampler in Theorem 1 can be roughly illustrated as in Figure 1.2, where the quantum state $\mathsf{C}\,|\mathsf{LWE}\rangle$ is

$$\sum_{\mathbf{s},\mathbf{e}} \left(\prod_i f(e_i) \right) |\mathbf{A}\mathbf{s} + \mathbf{e}\rangle \ , \tag{1.1}$$

and ℓ is a sufficiently large polynomial. When disposing the ancilla and measuring the final state, the outcome of the measurement is indeed an $\mathsf{LWE}_{m,n,q,|f|^2}$ instance for a uniformly distributed $\mathbf{A}$. We prove that such a circuit is oblivious, whatever the description of $\mathbf{U}$, assuming the hardness of $\mathsf{LWE}_{m,n,q,|f|^2}$.

We use an algorithm from [CLZ22] as a framework to obtain a candidate for the unitary above. The analysis is mostly centered around the following set of states:

$$\forall j \in \mathbb{Z}/q\mathbb{Z}, \quad |\psi_j\rangle := \sum_{e=0}^{q-1} f\left(e\right) |j + e \bmod q\rangle \ . \tag{1.2}$$

Roughly speaking, the success probability of the framework relies on designing good measurements for identifying a given state $|\psi_j\rangle$ whose index is unknown. The measurement of [CLZ22] imposes the following condition for the algorithm to succeed:

$$m \geq \frac{nq}{\min_x |\widehat{f}(x)|^2} \cdot \omega(\log \lambda) \ . \tag{1.3}$$

This result has two limitations. First, the lower bound on m and the runtime both grow at least polynomially with q (note that $\min |\widehat{f}| \leq 1$), which prevents us from choosing an exponential q. Second, the quantity $\min |\widehat{f}|$ is extremely small for a wide range of amplitude functions, notably $f = \sqrt{\vartheta_{\sigma,q}}$ up to a normalization factor (recall that $\vartheta_{\sigma,q}$ denotes the folded discrete Gaussian distribution). Indeed, we prove that if $\sigma \geq 1$, we have

$$q \cdot \min_x |\widehat{f}(x)|^2 \leq 32\sigma \cdot \exp\left(-\min\left(\frac{\pi\sigma^2}{4}, \frac{q^2}{4\sigma^2}\right)\right) \ .$$

This expression is most often extremely small. For example, for $\sigma = \Omega(\sqrt{n})$ and $q = \Omega(\sqrt{n}\sigma)$, the expression is $2^{-\Omega(n)}$.

We enhance the result of [CLZ22]. Our enhancement is three-fold. First, we note that a proposed quantum measurement in [CB98] can substantially increase the probability of success in the above task. However, it is not clear how to implement this measurement in time polynomial in m and $\log q$. Note that this is necessary to handle exponentially-large moduli. As the second enhancement, we show how to implement this measurement in time polynomial in m and $\log q$. Overall, the result of our analysis imposes the following condition:

$$m \geq \frac{n}{q \cdot \min_x |\widehat{f}(x)|^2} \cdot \omega(\log \lambda) \ .$$

While this is q^2 times better than Equation (1.3), it is still not sufficient to achieve Theorem 1. The quantity of $q \cdot \min_x |\widehat{f}(x)|^2$ is still extremely small for $f = \sqrt{\vartheta_{\sigma,q}}$ (up to a normalization factor). To increase this quantity, we introduce the last enhancement. We observe that for the purpose of oblivious LWE sampling for a distribution χ, we do not need to set $f = \sqrt{\chi}$ but can set $f = \sqrt{\chi} \cdot u$ for any function $u : \mathbb{Z}/q\mathbb{Z} \to \mathbb{C}$ taking values on the unit circle. Indeed, the new phases disappear when we measure the $\mathsf{C}|\mathsf{LWE}\rangle$ state to obtain the LWE sample. Interestingly, we show that the phases can greatly help to increase $\min |\widehat{f}|^2$. We suggest to set u as the sign function:

$$\forall x \in \mathbb{Z} \cap [0, q/2] : u(x) = 1 \quad \text{and} \quad \forall x \in \mathbb{Z} \cap (-q/2, 0) : u(x) = -1 \ .$$

For $f = \sqrt{\vartheta_{\sigma,q}}$ (up to a normalization factor), one can show that $q \cdot \min_x |\widehat{f \cdot u}(x)|^2 \geq 1/\sigma$. This yields Theorem 1.

Finally, we consider the application of our result to the security analyses of SNARK constructions. In particular, this requires to adapt our analysis of the oblivious sampler to matrices $\mathbf{A}$ corresponding to the module version of LWE [BGV12, LS15]. We note that the proof of knowledge security of our selection of SNARKs rely on the the hardness assumption of Linear-Only Vector Encryption/Encoding. We show that the adapted oblivious sampler invalidates this hardness assumption against quantum algorithms. Consequently, this invalidates the security analyses of several standard model lattice-based SNARKs [GMNO18, NYI+20, ISW21, SSEK22, CKKK23, GNSV23]. We stress that this

does not break the constructions themselves. For instance, the authors of [BISW17] mention a different route to analyze their SNARK construction in their Remark 4.9. Their approach is inspired by [BCI+13, Lemma 6.3] and can be applied to several constructions of SNARKs.

Analysis of Fiat-Shamir with aborts

We provide a detailed analysis of the correctness, security, and runtime of the signatures obtained via FSwA. Although there have been analyses of FSwA such as [AFLT16] for the ROM, and [KLS18] for the QROM, all of them contain a subtle common flaw. An alternative proof strategy in the QROM is suggested in [GHHM21], but not detailed.

We briefly explain the main difficulty for obtaining a UF-CMA security proof, and point out to the most common flaw. For aborting Σ-procotols, two different flavors of zero-knowledgeness are conceivable, depending on whether the zero-knowledge simulator is obliged to also fake aborting transcripts or not. The existence of a zero-knowledge simulator for all types of transcripts is a stronger condition. The previous works on FSwA only considered or provided the weaker notion of zero-knowledgeness for their Σ-protocols. Recall that in the security proof of the plain Fiat-Shamir transform, a real transcript $(w, H(w\|\mu), z)$ is replaced by (w', c', z') sampled by the zero-knowledge simulator and then the random oracle is reprogrammed as $H(w'\|\mu) := c'$. With the weak notion of zero-knowledgeness, one may expect that the same technique for answering a sign query μ as in the plain Fiat-Shamir proof applies here. Ideally, one would expect that replacing the last successful transcript (if there exists any) of the repetition by a fake one obtained by the weak simulator is sufficient for the argument. However, this approach has an issue that we explain below.

1. Assume the challenger in the genuine UF-CMA security game answers a sign query μ using a sequence of strings $w_1, w_2, \ldots$. Assume that aborting is a deterministic function of w and c (this is for example the case for Lyubashevsky's signatures with the parameters considered in [AFLT16]). Then, as soon as w_1 fails to produce a valid transcript, since the hash value $H(w_1\|\mu)$ is fixed, the signing algorithm can no longer return a valid signature which uses w_1. This is not the case in the game where the challenger cheat with the simulated transcripts. The reason is that the challenge is sampled fresh in each repetition, and the sign query could return a signature (w_1, c', z') for $c' \neq c$.

One possible approach to circumvent the above issue is to consider the strong notion of zero-knowledgeness. Then one may expect that, in a sign query μ, replacing all transcripts during the repetition by the fake ones generated by the strong simulator solves the issue. Yet, it is not sufficient. We explain below.

2. Recall that the zero-knowledge property of the underlying Σ-protocol is for a single execution of the protocol (as opposed to correlated repetition). Hence, replacing repetitions which rely on challenges computed as hash values by simulated transcripts requires challenges to be statistically independent. This is only possible if the hash function is evaluated on distinct inputs $w\|\mu$, which is not guaranteed: there might be collisions among the strings w's used within a sign query for a message μ.

FSwA has been analyzed and used numerous times, yet the second item above has been neglected in all of them (see, e.g., [Lyu12, Lemma 5.3], [Lyu16, Lemma 4.1], [KLS18, Theorem 3.2], and [Kat21, Lemma 4.6]). It is also neglected in [AFLT16]. Finally, the difficulty with the reprogramming 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.

Our first set of results concerns the correctness and the runtime. Relying on a negative result, we first modify the signing efficiency requirement in the FSwUA paradigm. Generally, it is required that the signing algorithm must be expected polynomial-time, where the expectation is taken over its internal randomness and that of the random oracle. We argue that this requirement is flawed and propose the following one: the runtime must be bounded by a polynomial with overwhelming probability. Then, we provide a runtime and correctness analysis of the FSwUA. We also provide a correctness analysis for FSwBA. As far as we are aware of, there is no detailed correctness analysis of FSwA in the literature.

Our second set of results relates to the security analyses of FSwA. We provide two security analyses for FSwBA in the QROM, the first one by correcting the one from [KLS18], and the second by adapting the approach suggested in [GHHM21]. For the result based on [KLS18], we rely on a detailed history-free analysis of the random oracle in the quantum setting, while For the other approach based on [GHHM21], a less involved analysis suffices thanks to the adaptive reprogramming approach. We further analyze the strong UF-CMA security of the signature. In this model of security, the adversary is allowed to produce a forgery for a message that is queried before, but the forgery must be different.

We provide an overview of our results about FSwBA in Table 1.1. The "reduction loss" is a bound on the difference of success probabilities of the adversary in the UF-CMA and UF-NMA security games, where in the latter the adversary makes no sign query. We assume the circuit model for quantum computations, except when mentioned otherwise. The numbers of hash and sign queries the adversary is allowed to make are respectively denoted by Q_H and Q_S. The losses and runtimes are also parameterized by the maximum number of repetitions B, the min-entropy α of the first message of the prover and zero-knowledge error ε_{zk} of the underlying interactive protocol (see Definitions 28, 29 and 31). The table assumes that $Q_H \geq B \cdot Q_S$ (this assumption is justified by the fact that hash evaluations can be made without restriction whereas sign queries require interaction with the signer). Similarly, the zero-knowledge simulation time is neglected (unless it is very large, its contribution is typically dominated by the terms in the table). We also omit constant factors.

We observe that the QROM analyses are incomparable. In particular, the analysis based on the adaptive reprogramming technique from [GHHM21] is tight only when assuming Quantum Random Access Classical Memory (QRACM), which is not necessarily implied by the quantum circuit model of computation. The analysis based on th history-free technique from [KLS18] is tight only when considering adversaries that may make at most one sign query for any message ($\mathsf{UF\text{-}CMA}_1$ security). This covers the deterministic version of the resulting signature, obtained by deriving the randomness from the message via a pseudo-random function evaluation. For UF-CMA security, the reduction is not tight (even assuming QRACM) and the reduction loss is higher than the one obtained with the adaptive reprogramming technique.

Analysis	Hash function	Reduction loss	Reduction runtime overhead
Adaptive reprogramming (Th. 14)	ROM	$2^{-\alpha}BQ_SQ_H$ $+ \varepsilon_{zk}BQ_S$	$Q_H \log(Q_H)$
Adaptive reprogramming (Th. 14)	QROM	$2^{-\alpha/2}BQ_SQ_H^{1/2}$ $+ \varepsilon_{zk}BQ_S$	$Q_H\log(BQ_S)$ with QRACM BQ_SQ_H without
History-free for CMA_1 security (Th. 11)	QROM	$2^{-\alpha/2}BQ_H$ $+ \varepsilon_{zk}^{1/2}B^{1/2}Q_H^{3/2}$	BQ_H
History-free for CMA security (Th. 13)	QROM	$2^{-\alpha/2}BQ_SQ_H$ $+ \varepsilon_{zk}^{1/2}B^{1/2}Q_H^{3/2}$	BQ_SQ_H

Table 1.1: Comparison of the security analyses of FSwBA.

We then give a security analysis for FSwUA in the QROM (with a tighter reduction in the ROM), by relating it to the security of FSwBA. Simplifying the terms as above, we prove that the adversary cannot distinguish the signing algorithm with B-bounded repetition and the signing algorithm with unbounded repetition unless with the following advantage:

$$Q_S\beta^B + \frac{\beta^B 2^{-\alpha}}{(1-\beta)^3} + \begin{cases} 2^{-\alpha}BQ_SQ_H & \text{in the ROM,} \\ 2^{-\frac{\alpha}{2}}BQ_SQ_H^{\frac{1}{2}} & \text{in the QROM,} \end{cases}$$

where β is the probability of aborting for the underlying Σ-protocol.

Finally, as a side contribution, we generalize our analysis to rely on a Σ-protocol whose simulator's quality is measured in terms of the Rényi divergence (rather than the statistical distance) for non-aborting transcripts. As pointed out in [DFPS22], in the case of Lyubashevsky's signature with Gaussian distributions [Lyu12], when the signature is replaced with the non-aborting simulator in the security proof, the analysis based on the divergence provides security for a larger range of parameters. This allows one to decrease the standard deviation of the distribution in the signature, which in turn reduces the signature size by a small amount.

Related publications

[DFS24] Quantum Oblivious LWE Sampling and Insecurity of Standard Model Lattice-based SNARKs, with Thomas Debris-Alazard and Damien Stehlé. To appear in *Symposium of Theory of Computing (STOC) 2024*.

[DFPS23] A Detailed Analysis of Fiat-Shamir with Aborts, with Julien Devevey, Alain Passelègue, and Damien Stehlé. In *CRYPTO 2023*.

1.7 Organization

The results related to the oblivious LWE sampling is discussed in Chapter 2, and the analysis of Fiat-Shamir with aborts is presented in Chapter 3. The chapters can be read independently. Some general notations are introduced in Chapter 1.7. Further preliminaries are recalled in each chapter.

General Notations

We let λ denote the security parameter in all chapters. We implicitly assume that all variables are parameterized by the security parameter λ.

The functions ln and log refer to the logarithm in base e and 2, respectively.

Definition 1. *Let $f, g : \mathbb{N}^+ \to \mathbb{R}$ be two functions. The Landau notations are defined as follows:*

$$\begin{aligned}
f = O(g) &\iff \exists N, c \in \mathbb{R}\ \forall n \geq N :\ |f(n)| \leq c|g(n)|\ , \\
f = \Omega(g) &\iff \exists N, c \in \mathbb{R}\ \forall n \geq N :\ |f(n)| \geq c|g(n)|\ , \\
f = \Theta(g) &\iff f = O(g) \text{ and } f = \Omega(g), \\
f = \omega(g) &\iff \lim_{n\to\infty} \frac{|f(n)|}{|g(n)|} = 0\ , \\
f = o(g) &\iff \lim_{n\to\infty} \frac{|g(n)|}{|f(n)|} = 0\ .
\end{aligned}$$

When it is not clear from the context, we use subscripts to clarify the input parameter, for instance $\Omega_\lambda(\cdot)$.

Definition 2. *We let $\mathsf{poly}(\lambda)$ denote any function which is of order $O(\lambda^a)$ for some constant a.*

Definition 3. *We let $\mathsf{negl}(\lambda)$ denote a function that is of order $O(1/\lambda^b)$ for every constant $b > 0$.*

Furthermore, a function of λ is called overwhelming if it is equal to $1 - \mathsf{negl}(\lambda)$.

Sometimes, we will use a subscript to stress the random variable specifying the associated probability space over which the probabilities or expectations are taken. For instance the probability $\mathbb{P}_X(E)$ of the event E is taken over the probability space S with respect to the induced measure by X. We let $U(S)$ denote the uniform distribution over S. Given any distribution X, the distribution $X^{\otimes m}$ is defined as $(X_1, \ldots, X_m)$ where X_i's are independently distributed as X.

Definition 4. *For any two discrete probability distributions X and Y over a set S, their statistical distance (also called the total variation distance) is defined as:*

$$\Delta(X, Y) := \frac{1}{2} \sum_{s \in S} |\mathbb{P}_X(s) - \mathbb{P}_Y(s)|\ .$$

We define probabilistic polynomial-time algorithms as follows.

Definition 5. *We say that a sequence of classical circuits $(C_i)_i$ is Probabilistic Polynomial-Time (PPT) if there exists a deterministic polynomial-time Turing machine that takes i in unary as input and outputs the description of C_i with logical gates. The circuit C_i is allowed to use an auxiliary $\mathsf{poly}(i)$-large uniform bit-string.*

We recall quantum polynomial-time algorithms as follows

Definition 6. *We say that a sequence of quantum circuits $(Q_i)_i$ is Quantum Polynomial-Time (QPT) if there exists a deterministic polynomial-time Turing machine that takes i in unary as input and outputs the description of Q_i with quantum gates and measurements. The circuit Q_i is allowed to use an auxiliary $\mathsf{poly}(i)$-large ancilla.*

Chapter 2

Oblivious LWE Sampling

The results of this chapter are based on the collaboration of the author with Thomas Debris-Alazard and Damien Stehlé. The following article is related to this chapter.

[DFS24] Quantum Oblivious LWE Sampling and Insecurity of Standard Model Lattice-based SNARKs, with Thomas Debris-Alazard and Damien Stehlé. To appear in *Symposium of Theory of Computing (STOC) 2024*.

As discussed earlier, the quantum setting causes difficulties for the definition of obliviousness. In Section 2.3, we first discuss this matter and propose a quantum definition of obliviousness which consistently covers the classical one. We then show how obliviousness can be reduced to synthesizing a quantum state that is, roughly speaking, a superposition of LWE samples.

The main result of this chapter is Theorem 1. The proof is presented in Section 2.4 and 2.5. The oblivious LWE sampler is detailed in Algorithm 1. In Section 2.1, we provide a high-level overview of how this sampler is constructed.

In Section 2.6, the application of our main theorem is discussed. We show how obliviously sampling LWE instances breaks the hardness assumption of Linear-Only Vector Encryption/Encoding.

2.1 Overview of the main result

Assume we have a (classically) known matrix $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, and that we manage to build the quantum state from Equation (1.1) using a unitary transformation (with possibly auxiliary registers equal to zero). Creating such a state was studied in [SSTX09] and referred to as the $\mathsf{C}\,|\mathsf{LWE}\rangle$ problem in [CLZ22]. We show that oblivious LWE sampling reduces to $\mathsf{C}\,|\mathsf{LWE}\rangle$. Intuitively, a measurement of the state above provides an LWE sample $\mathbf{As}+\mathbf{e}$ for a uniformly distributed $\mathbf{s}$ and a vector $\mathbf{e}$ with distribution χ proportional to $|f|^2$: there is no reason for a specific $\mathbf{s}$ to be privileged, and this algorithm does not seem to have any additional knowledge about the LWE solution. We formalize this

intuition using the obliviousness sampling definitions discussed above (which coincide here, as we have a unitary-then-measure algorithm). The result also holds if we add non-constant phases for $\mathbf{s}$ (for example to obtain $\mathbf{s}$ that is uniform among those with binary coordinates). It also allows the parameter m from $\mathsf{C}\,|\mathsf{LWE}\rangle$ to be larger than the one we want for oblivious LWE sampling, as we may throw away the superfluous coordinates without compromising the obliviousness.

We now discuss two existing approaches for solving $\mathsf{C}\,|\mathsf{LWE}\rangle$. The first one, derived from the LWE hardness proof from [Reg09], is to generate the following quantum state

$$\sum_{\mathbf{s},\mathbf{e}} \left(\prod_i f(e_i) \right) |\mathbf{s}\rangle \, |\mathbf{A}\mathbf{s} + \mathbf{e}\rangle \ , \tag{2.1}$$

up to normalization, and then to uncompute $|\mathbf{s}\rangle$ from $|\mathbf{A}\mathbf{s} + \mathbf{e}\rangle$. Generating this state can be done efficiently (possibly under some conditions on f) by first creating the superposition over all $\mathbf{s}$ and $\mathbf{e}$ of $|\mathbf{s}\rangle\,|\mathbf{e}\rangle$ with proper amplitudes, and then multiplying the first register by $\mathbf{A}$ to add it to the second one. To remove $\mathbf{s}$, i.e., to replace $\mathbf{s}$ by $\mathbf{0}$ in the first register, the approach from [Reg09] is to recover $\mathbf{s}$ from the second register and subtract it to the first one, by using a quantized LWE solver. This leads to a reduction from $\mathsf{C}\,|\mathsf{LWE}\rangle$ to LWE. Unfortunately, in our context, this is not satisfactory, as LWE must be assumed difficult for oblivious LWE sampling to be feasible.

Another approach for solving $\mathsf{C}\,|\mathsf{LWE}\rangle$ was recently proposed in [CLZ22, Sec. 5]. The proposed algorithm does not require any oracle for a presumably hard problem, but seems restricted to specific parametrizations of $\mathsf{C}\,|\mathsf{LWE}\rangle$, as we discuss below.

- First, it builds the quantum state from Equation (2.1). It can be rewritten as follows:

$$\begin{aligned} \sum_{\mathbf{s},\mathbf{e}} \bigotimes_{i \le m} |\mathbf{s}\rangle \, f(e_i) \, |\langle \mathbf{a}_i, \mathbf{s}\rangle + e_i\rangle &= \sum_{\mathbf{s}} |\mathbf{s}\rangle \left(\bigotimes_{i \le m} \sum_{e_i} f(e_i) \, |\langle \mathbf{a}_i, \mathbf{s}\rangle + e_i\rangle \right) \\ &= \sum_{\mathbf{s}} |\mathbf{s}\rangle \left(\bigotimes_{i \le m} \left|\psi_{\langle \mathbf{a}_i, \mathbf{s}\rangle}\right\rangle \right) \ , \end{aligned}$$

  where $|\psi_k\rangle = \sum_e f(e) \, |k + e\rangle$ for all $k \in \mathbb{Z}/q\mathbb{Z}$ (up to normalization).

- Second, it individually considers all sub-registers $|\psi_{\langle \mathbf{a}_i, \mathbf{s}\rangle}\rangle$ of the $|\mathbf{A}\mathbf{s} + \mathbf{e}\rangle$ register, and performs a measurement for each one of them. For each i, the measurement consists in sampling a uniform $k_i \in \mathbb{Z}/q\mathbb{Z}$ and applying a projective measurement with respect to the (normalized) Gram-Schmidt orthogonalization of $|\psi_{k_i+1}\rangle, |\psi_{k_i+2}\rangle, \ldots, |\psi_{k_i-1}\rangle, |\psi_{k_i}\rangle$ (we assume that the $|\psi_j\rangle$'s are linearly independent, and the indices are taken modulo q). If the measurement is on the last direction, then $|\psi_{\langle \mathbf{a}_i, \mathbf{s}\rangle}\rangle$ cannot have a component on the span of $|\psi_{k_i+1}\rangle, |\psi_{k_i+2}\rangle, \ldots, |\psi_{k_i-1}\rangle$ and must be equal to $|\psi_{k_i}\rangle$. When successful, the measurement indicates that $\langle \mathbf{a}_i, \mathbf{s}\rangle = k_i \bmod q$.

- Third, sufficiently many successful measurements are collected through the different values of i, to obtain many equations of the type $\langle \mathbf{a}_i, \mathbf{s}\rangle = k_i$, where the $\mathbf{a}_i$'s and k_i's are known. This is then fed to a quantized Gaussian elimination algorithm (recall that we are working with a superposition over all $\mathbf{s}$'s). The latter outputs $\mathbf{s}$, which is then subtracted from the first register.

It was proved in [CLZ22] that this algorithm solves $\mathsf{C}\,|\mathsf{LWE}\rangle$ in polynomial time, if m and q are polynomial in the security parameter λ, if a state proportional to $\sum_e f(e)\,|e\rangle$ can be efficiently computed, and if

$$m = \frac{n}{p^{\mathrm{CLZ}}} \cdot \omega(\log \lambda) \quad \text{with} \quad p^{\mathrm{CLZ}} = \frac{\min_x |\widehat{f}(x)|^2}{q} \ .$$

Here, the notation $\widehat{f}$ refers to the Fourier transform over $\mathbb{Z}/q\mathbb{Z}$ of f. The quantity p^{CLZ} corresponds to the probability that an individual measurement of the second step succeeds. This result notably allows to solve $\mathsf{C}\,|\mathsf{LWE}\rangle$ (and hence oblivious LWE sampling) for q polynomial and χ set as the uniform distribution in an interval $[-B, B] \cap \mathbb{Z}$, for any $B \in \mathbb{Z}$ such that $0 < 2B+1 < q$ and $\gcd(2B+1, q) = 1$. Interestingly, by taking $q = 2$, the result also allows to solve the adaptation of $\mathsf{C}\,|\mathsf{LWE}\rangle$ to the decoding problem for uniform binary codes (also known as Learning Parity with Noise), and to obliviously sample points near codewords for the Bernoulli distribution with an arbitrary Bernoulli parameter in $(0, 1/2)$.

This result has two limitations. First, the lower bound on m and the runtime both grow at least polynomially with q (note that $\min |\widehat{f}| \leq 1$), which prevents us from choosing an exponential q. This is in part due to the uniform guess of $\langle \mathbf{a}_i, \mathbf{s} \rangle$ in the measurement, which directly incurs a loss by a factor q in the success probability of each individual measurement. Note that for a fixed standard deviation σ, LWE becomes no harder as q increases, so that one can expect that it is indeed no easier to obliviously sample LWE instances (intuitively, the easier is the considered problem, the harder it is to obliviously sample instances). Most SNARKs that we consider use an exponential q. Second, the quantity $\min |\widehat{f}|$ is extremely small for a wide range of amplitude functions, notably $f = \sqrt{\vartheta_{\sigma,q}}$ up to a normalization factor (recall that $\vartheta_{\sigma,q}$ denotes the folded discrete Gaussian distribution). Indeed, we prove in Lemma 18 that in that case and if $\sigma \geq 1$, we have

$$q \cdot \min |\widehat{f}|^2 \leq 32\sigma \cdot \exp\left(-\min\left(\frac{\pi\sigma^2}{4}, \frac{q^2}{4\sigma^2}\right)\right) \ .$$

This expression is most often extremely small. For example, for $\sigma = \Omega(\sqrt{n})$ and $q = \Omega(\sqrt{n}\sigma)$, the expression is $2^{-\Omega(n)}$. This prevents from meaningfully using the result for the discrete Gaussian distribution, which is the most common choice of error distribution for LWE.

As a remark, we would like to highlight that in the reduction from oblivous LWE sampling to the $\mathsf{C}\,|\mathsf{LWE}\rangle$ problem, a crucial step consists in erasing the memory (up to some negligible error) that has been used during the course of computation. Constructing a superposition of the possible values for the secret $\mathbf{s}$ in a separate register is neccessary for the algorithm to succeed, while the last step to revert it back to $|\mathbf{0}\rangle$ is pivotal for obtaining obliviousness. We note that in designing classical oblivious samplers, forgetting the history of the computation remains a challenging obstacle.

Measuring with increased success probability.

The second step of the algorithm from [CLZ22] consists in taking $|\psi_k\rangle$ for an unknown $k \in \mathbb{Z}/q\mathbb{Z}$ as input and returning k, i.e., distinguishing the quantum states $|\psi_0\rangle, \ldots, |\psi_{q-1}\rangle$. One could proceed as follows if the states were orthogonal. Consider the well-defined

projective measurement $(\mathbf{E}_i)_i$ defined by $\mathbf{E}_i = |\psi_i\rangle\langle\psi_i|$ for $0 \leq i < q$. Then, if the state $|\psi_k\rangle$ is given, the probability to see k as the outcome is $\langle\psi_k| \mathbf{E}_k^\dagger \mathbf{E}_k |\psi_k\rangle = 1$. In other words, this quantum measurement perfectly distinguishes the quantum states. However, when the $|\psi_k\rangle$'s are not orthogonal (which is our case except for particular amplitudes like f being 1 in 0 and 0 elsewhere), it is known that there exists no quantum measurement to perfectly distinguish them (see [NC11, Box 2.3]). The measurement from [CLZ22] may output a special symbol $\perp$ representing the "unknown" answer, but it does not make any mistake, in the sense that it never outputs some $\ell \in \mathbb{Z}/q\mathbb{Z}$ different from k. Such a process is referred to as unambiguous. This property is important for the subsequent Gaussian elimination step, as it requires all linear equations to be correct. We define the error parameter of the unambiguous measurement as the maximal probability that the measurement outputs $\perp$ over all possible input states:

$$p_\perp = \max_k \langle\psi_k| \mathbf{E}_\perp |\psi_k\rangle \ ,$$

where $\mathbf{E}_\perp$ corresponds to the outcome $\perp$. The measurement from [CLZ22] satisfies

$$1 - p_\perp^{\mathrm{CLZ}} = p^{\mathrm{CLZ}} = \frac{\min |\widehat{f}|^2}{q} \ .$$

We propose to change the unambiguous measurement by the positive operator-valued measure (POVM) from [CB98]. It is known to be "optimal" when the $|\psi_i\rangle$'s are symmetric and linearly independent, in the sense that it minimizes the error parameter $p_\perp$ over all possible choice of POVMs. Here, symmetric means that there exists a unitary $\mathbf{U}$ such that $|\psi_i\rangle = \mathbf{U} \cdot |\psi_{i-1 \bmod q}\rangle$ for all $0 \leq i < q$: our states indeed satisfy this property with $\mathbf{U}$ being the mod-q translation operator. The linear independence property may or may not be satisfied, depending on the choice of f (this is a difficulty encountered in other sections of [CLZ22]). The measurement from [CB98] is defined as follows:

$$\forall 0 \leq i < q : \mathbf{E}_i = \alpha \cdot \left|\psi_i^\perp\right\rangle \left\langle\psi_i^\perp\right| \quad \text{and} \quad \mathbf{E}_\perp = \mathbf{I} - \sum_i \mathbf{E}_i \ ,$$

where $\left|\psi_i^\perp\right\rangle$ is a unit vector orthogonal to all $|\psi_j\rangle$'s for $j \neq i$, for all $0 \leq i < q$. The scalar α is chosen maximal such that the POVM is well-defined, i.e., such that $\mathbf{E}_\perp$ is non-negative: it is the inverse of the largest eigenvalue of $\sum_i \mathbf{E}_i$. We compute that this measurement leads to:

$$1 - p_\perp^{\mathrm{CB}} = \frac{q^2\ \alpha}{\sum_{x \in \mathbb{Z}/q\mathbb{Z}} \left|\widehat{f}(x)\right|^{-2}} = q \cdot \min \left|\widehat{f}\right|^2 \ .$$

Note that the success probability of the measurement is a factor q^2 higher than the one from [CLZ22]. By the union bound (and still assuming q prime), with the optimal unambiguous measurement, it suffices to set

$$m = \frac{n}{q \cdot \min |\widehat{f}|^2} \cdot \omega(\log \lambda) \ . \tag{2.2}$$

How to implement the measurement in time polynomial in $\log q$

Compared to the [CLZ22] approach, the quantum distinguishing measurement from [CB98] allows one to choose m smaller by a factor q^2 and also to gain a factor q^2 in the runtime. But beyond these considerations over the parameter m, that we discuss more deeply in Subsection 2.1, recall that we are looking for a sampler whose runtime is polynomial in m and $\log q$. We therefore have to efficiently implement the above POVM. Although the measurement was introduced in [CB98], this work does not specify how to efficiently compute it. The POVM components $(\mathbf{E}_j)_{0\leq j<q}$ turn out to be some projections $(|\psi_j^\perp\rangle\langle\psi_i^\perp|)_{0\leq j<q}$. A first approach would be to compute the quantum states $|\psi_i^\perp\rangle$'s, which are given by (here ω_q refers to a primitive q-th root of unity.)

$$\forall j : \left|\psi_j^\perp\right\rangle = \frac{1}{\sqrt{N}} \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{-jx} \cdot \overline{\widehat{f}(-x)^{-1}} \ |\chi_x\rangle$$

where $N = \sum_{x\in\mathbb{Z}/q\mathbb{Z}} |\widehat{f}(-x)|^{-2}$ and $(|\chi_x\rangle)_{x\in\mathbb{Z}/q\mathbb{Z}}$ denotes the Fourier basis, namely

$$\forall x : |\chi_x\rangle = \frac{1}{\sqrt{q}} \sum_{y\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{xy} |y\rangle \ .$$

However, even if one were able to compute these quantum states in polynomial time, there are q of them, making it difficult to obtain a runtime polynomial in $\log q$. One may therefore try to find a way to efficiently compute a unitary sending $|j\rangle|0\rangle$ to $|j\rangle|\psi_j^\perp\rangle$ for all j. This seems to be a challenging path, as such a unitary would need to implement the POVM. Let us backtrack a little, and try to see how the POVM given by $(\mathbf{E}_j)_{0\leq j<q}$ and $\mathbf{E}_\perp$ acts on the $|\psi_j\rangle$'s. First, let us decompose the $|\psi_j\rangle$'s in the Fourier basis:

$$\forall j : |\psi_j\rangle = \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \widehat{f}(-x) \cdot \omega_q^{-jx} \ |\chi_x\rangle \ .$$

To correctly identify $|\psi_j\rangle$, we project it according to $\mathbf{E}_j = |\psi_j^\perp\rangle\langle\psi_j^\perp|$. This leads to considering the following Hermitian product:

$$\begin{aligned}\left\langle\psi_j^\perp\middle|\psi_j\right\rangle &= \frac{1}{\sqrt{Nq^n}} \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{jx} \cdot \widehat{f}(-x)^{-1} \cdot \widehat{f}(-x) \cdot \omega_q^{-jx} \\ &= \frac{1}{\sqrt{Nq^n}} \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \widehat{f}(-x)^{-1} \cdot \widehat{f}(-x)\end{aligned}$$

In other words, when projecting $|\psi_j\rangle$ on $|\psi_j^\perp\rangle$, we want to "remove" $\widehat{f}(-x)$ in the amplitudes of $|\psi_j\rangle$ in its Fourier basis decomposition. Therefore, simulating the POVM of [CB98] leads to considering the unitary performing this task, i.e., a unitary $\mathbf{V}$ such that

$$\forall x : |\chi_x\rangle|0\rangle \longmapsto \frac{\min|\widehat{f}|}{\widehat{f}(-x)} |\chi_x\rangle|0\rangle + \sqrt{1 - \left|\frac{\min|\widehat{f}|}{\widehat{f}(-x)}\right|^2} |\chi_x\rangle|1\rangle$$

(We note that a similar approach was considered in [CT23].) Such a unitary is efficiently computable under the conditions that both $\min|\widehat{f}|$ and $\widehat{f}(-x)$ can be efficiently

approximated. Let us check that $\mathbf{V}$ indeed "simulates" the measurement from [CB98], by computing how it acts on the $|\psi_j\rangle$'s:

$$\begin{aligned}
\mathbf{V}\left(|\psi_j\rangle\,|0\rangle\right) &= \mathbf{V}\left(\sum_{x\in\mathbb{Z}/q\mathbb{Z}} \widehat{f}(-x)\cdot\omega_q^{-jx}\ \ |\chi_x\rangle\,|0\rangle\right)\\
&= \sum_{x\in\mathbb{Z}/q\mathbb{Z}}\left(\min|\widehat{f}|\cdot\omega_q^{-jx}\ \ |\chi_x\rangle\,|0\rangle + \widehat{f}(-x)\cdot\omega_q^{-jx}\cdot\sqrt{1-\left|\frac{\min|\widehat{f}|}{\widehat{f}(-x)}\right|^2}\ \ |\chi_x\rangle\,|1\rangle\right)\\
&= \sqrt{q}\cdot\min|\widehat{f}|\ \ |j\rangle\,|0\rangle + \sum_{x\in\mathbb{Z}/q\mathbb{Z}}\widehat{f}(-x)\cdot\omega_q^{-jx}\cdot\sqrt{1-\left|\frac{\min|\widehat{f}|}{\widehat{f}(-x)}\right|^2}\ \ |\chi_x\rangle\,|1\rangle
\end{aligned}$$

In other words, we have (for some quantum state $|\eta_j\rangle$):

$$\mathbf{V}\,|\psi_j\rangle\,|0\rangle = \sqrt{p^{\mathrm{CB}}}\,|j\rangle\,|0\rangle + \sqrt{1-p^{\mathrm{CB}}}\,|\eta_j\rangle\,|1\rangle\ \ ,$$

where p^{CB} turns out to be equal to the success probability of the POVM given in [CB98], i.e., $p^{\mathrm{CB}} = 1-p_\perp^{\mathrm{CB}}$. Therefore, by interpreting any quantum state whose last qubit is $|1\rangle$ as $\perp$, applying $\mathbf{V}$ amounts to quantumly recovering j from $|\psi_j\rangle$ with probability $1-p_\perp^{\mathrm{CB}}$.

Increasing the Fourier coefficients

At this stage, we have that the modified $\mathsf{C}\,|\mathsf{LWE}\rangle$ algorithm is polynomial in m and $\log q$. We also have decreased the feasibility threshold on m from $nq/\min|\widehat{f}|^2\cdot\omega(\log\lambda)$ to $n/(q\cdot\min|\widehat{f}|^2)\cdot\omega(\log\lambda)$. However, as observed in [CLZ22], the quantity $\min|\widehat{f}|^2$ can be extremely low for distributions of interest.

Our last technical ingredient stems from the observation that for the purpose of oblivious LWE sampling for a distribution χ, we do not need to set $f=\sqrt{\chi}$ but can set $f=\sqrt{\chi}\cdot u$ for any function $u:\mathbb{Z}/q\mathbb{Z}\to\mathbb{C}$ taking values on the unit circle. Indeed, the new phases disappear when we measure the $\mathsf{C}\,|\mathsf{LWE}\rangle$ state to obtain the LWE sample. Interestingly, the phases can greatly help to increase $\min|\widehat{f}|^2$. The astute reader will note that the circuit described above then needs to be updated to account for the phases, but we show that efficiency can be preserved, notably for the function u that we choose. We propose to set u as the sign function:

$$\forall x\in\mathbb{Z}\cap[0,q/2]: u(x)=1 \quad\text{and}\quad \forall x\in\mathbb{Z}\cap(-q/2,0): u(x)=-1\ .$$

Then the following relations hold, for q odd and for all $x\in\mathbb{Z}/q\mathbb{Z}$ viewed as an integer in $(-q/2,q/2]$:

$$\begin{aligned}
\widehat{f}(x) &= \frac{1}{\sqrt{q}}\sum_{y\in\mathbb{Z}/q\mathbb{Z}} f(y)\cdot\omega_q^{xy}\\
&= \frac{1}{\sqrt{q}}\sum_{\mathbb{Z}\cap[0,q/2]}\sqrt{\chi(y)}\cdot\omega_q^{xy} - \frac{1}{\sqrt{q}}\sum_{\mathbb{Z}\cap(-q/2,0)}\sqrt{\chi(y)}\cdot\omega_q^{xy}\\
&= \frac{\sqrt{\chi(0)}}{\sqrt{q}} + \frac{1}{\sqrt{q}}\sum_{\mathbb{Z}\cap(0,q/2]}\sqrt{\chi(y)}\cdot(\omega_q^{xy}-\omega_q^{-xy})\ .
\end{aligned}$$

Note that the summand is an imaginary number and hence that $\sqrt{\chi(0)/q}$ is the real part of $\widehat{f}(x)$. As a result, we obtain that $\min|\widehat{f}| \geq \sqrt{\chi(0)/q}$. By combining with Equation (2.2), it suffices to set $m = n/\chi(0) \cdot \omega(\log \lambda)$. For the specific case of the folded integer Gaussian distribution, we have that $\chi(0) \approx 1/\sigma$, leading to an efficient algorithm when σ is polynomial in λ.

We stress that we use both the phases and the improved unambiguous measurement to obtain an efficient algorithm. We already saw that the the improved measurement alone is insufficient. Conversely, if we use the phases and the measurement from [CLZ22], then it seems that we need m to grow as $nq^2/\sigma \cdot \omega(\log \lambda)$, which forbids a runtime polynomial in $\log q$.

2.2 Preliminaries

We will consider the additive group $\mathbb{Z}/q\mathbb{Z}$ for $q \geq 2$ and may write its elements as

$$\mathbb{Z}/q\mathbb{Z} = \left\{ j \in \mathbb{Z} \ : \ -\frac{q}{2} < j \leq \frac{q}{2} \right\} \ .$$

We define ω_q as $\exp(2\pi i/q)$. Recall that the discrete Fourier transform of every function $f : \mathbb{Z}/q\mathbb{Z} \to \mathbb{C}$ is defined as follows:

$$\forall x \in \mathbb{Z}/q\mathbb{Z}, \quad \widehat{f}(x) := \frac{1}{\sqrt{q}} \sum_{y \in \mathbb{Z}/q\mathbb{Z}} f(y) \cdot \omega_q^{-xy} \ .$$

For an integer m and a real number $r \geq 0$, we let $\mathrm{B}_m(r)$ denote the ball of $\mathbb{R}^m$ with radius r. Vectors are in column notation and are written with bold letters (such as $\mathbf{x}$). Uppercase bold letters are used to denote matrices (such as $\mathbf{A}$). For vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n$, we let $(\mathbf{a}_1 | \ldots | \mathbf{a}_n)$ denote the matrix whose columns are the $\mathbf{a}_i$'s. For any two vectors $\mathbf{x}, \mathbf{y} \in (\mathbb{Z}/q\mathbb{Z})^d$, we define their inner product as

$$\langle \mathbf{x}, \mathbf{y} \rangle := \sum_{i=1}^{d} x_i y_i \bmod q \ .$$

Let $f : S \to \mathbb{C}$ be a function. We define the function $f^{\otimes d} : S^d \to \mathbb{C}$ as $f^{\otimes d}(x_1, \ldots, x_d) := f(x_1) \cdots f(x_d)$, for all $(x_1, \ldots, x_d) \in S^d$. When the dimension d is clear from the context, we abuse the notation and write f instead of $f^{\otimes d}$. Let S be a finite set and $f : S \to \mathbb{C}$. We say that f is an *amplitude function* if

$$\sum_{x \in S} |f(x)|^2 = 1 \ .$$

We note that $f^{\otimes d}$ is an amplitude function whenever f is an amplitude function.

2.2.1 Quantum computations

We recall some background on quantum computation.

Partial trace. For our purposes, we need to describe sub-systems of a given "composite" quantum system. This description involves the partial trace. Let $\mathcal{A}$ and $\mathcal{B}$ be two Hilbert

spaces with $\{|a\rangle\}_{a\in\mathcal{I}}$ and $\{|b\rangle\}_{b\in\mathcal{J}}$ as their orthonormal bases, respectively. For all $a_1, a_2 \in \mathcal{I}$ and $b_1, b_2 \in \mathcal{J}$, tracing out the register of $\mathcal{B}$ is defined as follows:

$$\mathrm{tr}_{\mathcal{B}}\left(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|\right) := \langle b_1|b_2\rangle\, |a_1\rangle\langle a_2| \ .$$

It is extended by linearity.

Trace distance. We will also use the *trace distance* which is defined over two quantum states ρ, σ as follows:

$$D_{\mathrm{tr}}(\rho,\sigma) := \frac{1}{2}\,\mathrm{tr}\left(\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)}\right) \ .$$

For pure quantum states $|\psi\rangle$ and $|\varphi\rangle$, it can be simplified to $\sqrt{1-|\langle\varphi|\psi\rangle|^2}$. The trace distance has the following properties (see [NC11, Th. 9.2]):

- for any joint states ρ, σ over $\mathcal{A}\otimes\mathcal{B}$, it holds that $D_{\mathrm{tr}}(\mathrm{tr}_{\mathcal{B}}(\rho), \mathrm{tr}_{\mathcal{B}}(\sigma)) \le D_{\mathrm{tr}}(\rho,\sigma)$;
- for any quantum states ρ, σ, τ, it holds that $D_{\mathrm{tr}}(\rho,\sigma) \le D_{\mathrm{tr}}(\rho,\tau) + D_{\mathrm{tr}}(\tau,\sigma)$;
- for any quantum states ρ, σ, τ, it holds that $D_{\mathrm{tr}}(\rho\otimes\tau, \sigma\otimes\tau) = D_{\mathrm{tr}}(\rho,\sigma)$;
- for any quantum algorithm $\mathcal{Q}$ and any quantum states ρ, σ, it holds that $D_{\mathrm{tr}}(\mathcal{Q}(\rho), \mathcal{Q}(\sigma)) \le D_{\mathrm{tr}}(\rho,\sigma)$.

Let M be the set of possible outcomes of a measurement on the above states. Let X and Y be the distributions over M induced by measuring ρ and σ, respectively. We have:

$$\Delta(X,Y) \ \le \ D_{\mathrm{tr}}(\rho,\sigma) \ . \tag{2.3}$$

Quantum Fourier transform (QFT). The **QFT** over the additive group $\mathbb{Z}/q\mathbb{Z}$, whose characters are $\chi_x : y \mapsto \omega_q^{xy}$ for $x \in \mathbb{Z}/q\mathbb{Z}$, is defined as follows:

$$\forall x \in \mathbb{Z}/q\mathbb{Z}, \quad \mathbf{QFT}\,|x\rangle := \frac{1}{\sqrt{q}} \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{xy}\,|y\rangle \ .$$

The quantum states $|\chi_x\rangle := \mathbf{QFT}\,|x\rangle$ for $x \in \mathbb{Z}/q\mathbb{Z}$ are called the *Fourier basis*, whereas the states $|x\rangle$ form the *computational basis*. The following lemma recalls how the computational basis decomposes in the Fourier basis.

Lemma 1. *For any $q \ge 2$, it holds that*

$$\forall y \in \mathbb{Z}/q\mathbb{Z}, \quad |y\rangle = \frac{1}{\sqrt{q}} \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{-yx}\,|\chi_x\rangle \ .$$

Proof. We have the following equalities:

$$\begin{aligned}
\mathbf{QFT} \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{-yx}\,|\chi_x\rangle &= \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{-yx}\,\mathbf{QFT}\,|\chi_x\rangle \\
&= \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{-yx} \frac{1}{\sqrt{q}} \sum_{m\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{xm}\,\mathbf{QFT}\,|m\rangle \\
&= \sum_{m\in\mathbb{Z}/q\mathbb{Z}} \left(\frac{1}{\sqrt{q}} \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{x(m-y)}\right) \mathbf{QFT}\,|m\rangle \\
&= \sqrt{q}\,\mathbf{QFT}\,|y\rangle \ .
\end{aligned}$$

The result is obtained by applying $\mathbf{QFT}^{-1}$. □

2.2.2 Gaussian distributions

The Gaussian function centered around $\mathbf{0}$ with the standard deviation parameter $\sigma > 0$ is defined as:

$$\forall \mathbf{x} \in \mathbb{R}^m : \ \rho_\sigma(\mathbf{x}) := \mathrm{e}^{-\pi \frac{\|\mathbf{x}\|^2}{\sigma^2}} \ .$$

where $\|\cdot\|$ denotes the Euclidean norm of $\mathbf{x}$. The following lemma shows the concentration behaviour of ρ_σ over $\mathbb{Z}^m$.

Lemma 2 (Adapted from [Ban93, Le. 1.5])**.** *For any positive integer m and any real numbers $\sigma > 0$ and $\sigma' \geq \sigma/\sqrt{2\pi}$, it holds that*

$$\rho_\sigma\left(\mathbb{Z}^m \setminus \mathrm{B}_m(\sigma'\sqrt{m})\right) \leq \left(\frac{\sigma'}{\sigma}\ \sqrt{2\pi \mathrm{e}}\ \mathrm{e}^{-\pi \frac{\sigma'^2}{\sigma^2}}\right)^m \rho_\sigma(\mathbb{Z}^m) \ .$$

We have the following inequality.

Lemma 3. *For every $\sigma > 0$, we have $\sigma \leq \rho_\sigma(\mathbb{Z}) \leq 1 + \sigma$.*

Proof. We have $\rho_\sigma(\mathbb{Z}) \leq 1 + 2\int_0^{+\infty} \rho_\sigma(x)\mathrm{d}x = 1 + \sigma$, by comparing the sum and the integral. Moreover, using the Poisson summation formula, one obtains $\rho_\sigma(\mathbb{Z}) = \sigma \cdot \rho_{1/\sigma}(\mathbb{Z}) \geq \sigma$. □

The discrete Gaussian distribution over $\mathbb{Z}^m$ centered around $\mathbf{0}$ with the standard deviation σ is defined as follows:

$$\forall \mathbf{k} \in \mathbb{Z}^m : \ D_{\mathbb{Z}^m,\sigma}(\mathbf{k}) := \frac{\rho_\sigma(\mathbf{k})}{\rho_\sigma(\mathbb{Z}^m)} \ ,$$

where $\rho_\sigma(\mathbb{Z}^m) := \sum_{\mathbf{k} \in \mathbb{Z}^m} \rho_\sigma(\mathbf{k})$. Folding $D_{\mathbb{Z}^m,\sigma}$ modulo an integer q yields the distribution $\vartheta_{\sigma,q}$.

Definition 7 (Folded Discrete Gaussian Distribution)**.** *Let $q \geq 2$ an integer and $\sigma > 0$ a real number. We define the folded discrete Gaussian distribution over $(\mathbb{Z}/q\mathbb{Z})^m$ with standard deviation σ and folding parameter q by its probability mass function $\vartheta_{\sigma,q}$:*

$$\forall \mathbf{x} \in (\mathbb{Z}/q\mathbb{Z})^m : \ \vartheta_{\sigma,q}(\mathbf{x}) := \frac{\sum_{\mathbf{k} \in \mathbb{Z}^m} \rho_\sigma(\mathbf{x} + \mathbf{k}q)}{\rho_s(\mathbb{Z}^m)} \ .$$

We note that in all distributions above, the dimension of the input is implicit and can be derived from the context. The distribution $\vartheta_{\sigma,q}$ behaves very closely to $D_{\mathbb{Z}^m,\sigma}$.

Lemma 4. *Let $m \geq 1$ and $q \geq 2$ integers and $\sigma > 0$ a real number. Assume that $q \geq 2\sigma\sqrt{m}$. Then for every $\mathbf{x} \in \mathbb{Z}^m \cap (-q/2, q/2]^m$, it holds that*

$$D_{\mathbb{Z}^m,\sigma}(\mathbf{x}) \leq \vartheta_{\sigma,q}(\mathbf{x}) \leq D_{\mathbb{Z}^m,\sigma}(\mathbf{x}) + \mathrm{e}^{-\frac{q^2}{(2\sigma)^2}},$$

and

$$\sqrt{D_{\mathbb{Z}^m,\sigma}(\mathbf{x})} \leq \sqrt{\vartheta_{\sigma,q}(\mathbf{x})} \leq \sqrt{D_{\mathbb{Z}^m,\sigma}(\mathbf{x})} + \mathrm{e}^{-\frac{q^2}{8\sigma^2}} \ .$$

Proof. For $\mathbf{x} \in \mathbb{Z}^m \cap (-q/2, q/2]^m$, we have

$$\begin{aligned}
\sum_{\mathbf{k}\in\mathbb{Z}^m} \rho_\sigma(\mathbf{x}+\mathbf{k}q) &= \rho_\sigma(\mathbf{x}) + \sum_{\mathbf{k}\in\mathbb{Z}^m\setminus\{\mathbf{0}\}} \rho_\sigma(\mathbf{x}+\mathbf{k}q) \\
&\le \rho_\sigma(\mathbf{x}) + \sum_{\mathbf{k}\in\mathbb{Z}^m:\|\mathbf{k}\|\ge\frac{q}{2}} \rho_\sigma(\mathbf{k}) \\
&\le \rho_\sigma(\mathbf{x}) + \rho_\sigma\left(\mathbb{Z}^m \setminus \mathrm{B}_m\left(\frac{q}{2}\right)\right) \\
&\le \rho_\sigma(\mathbf{x}) + \left(\frac{q}{2\sigma\sqrt{m}}\,\sqrt{2\pi\mathrm{e}}\,\mathrm{e}^{-\pi\frac{q^2}{m(2\sigma)^2}}\right)^m \rho_\sigma(\mathbb{Z}^m) \quad \text{(Lemma 2 with } \sigma' = \frac{q}{2\sqrt{m}}\text{)} \\
&\le \rho_\sigma(\mathbf{x}) + \left(\mathrm{e}^{-\frac{q^2}{m(2\sigma)^2}}\right)^m \rho_\sigma(\mathbb{Z}^m)\ ,
\end{aligned}$$

where the last inequality follows since for every $x \ge 1$, we have $(1-\pi)x^2 \ge \ln x + \ln\sqrt{2\pi\mathrm{e}}$. This gives the first statement. For the other one, we take the square-root of the last inequality above to obtain:

$$\sqrt{\vartheta_{\sigma,q}(\mathbf{x})} \le \sqrt{\frac{\rho_\sigma(\mathbf{x}) + \mathrm{e}^{-\frac{q^2}{(2\sigma)^2}}\rho_\sigma(\mathbb{Z}^m)}{\rho_\sigma(\mathbb{Z}^m)}} \le \sqrt{\frac{\rho_\sigma(\mathbf{x})}{\rho_s(\mathbb{Z}^m)}} + \mathrm{e}^{-\frac{q^2}{8\sigma^2}}\ .$$

This completes the proof. □

2.2.3 Learning With Errors

We restate the definition of LWE here, with a minor additional notation.

Definition 8 (LWE)**.** *Let $m \ge n \ge 1$ and $q \ge 2$ be integers, and χ be a distribution over $\mathbb{Z}/q\mathbb{Z}$. The parameters m, n, q and χ are functions of some security parameter λ. Let $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, $\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n$ be sampled uniformly and $\mathbf{e} \in (\mathbb{Z}/q\mathbb{Z})^m$ be sampled from $\chi^{\otimes m}$. The search $\mathsf{LWE}_{m,n,q,\chi}$ problem is to find $\mathbf{s}$ and $\mathbf{e}$ given the pair $(\mathbf{A}, \mathbf{A}\mathbf{s}+\mathbf{e})$. The vectors $\mathbf{s}$ and $\mathbf{e}$ are respectively called the secret and the noise.*

Whenever χ is equal to the folded discrete Gaussian distribution $\vartheta_{\sigma,q}$ for some $\sigma > 0$, we overwrite the notation as $\mathsf{LWE}_{m,n,q,\sigma}$.

2.3 Witness Obliviousness

In this section, we are interested in the task of sampling an LWE instance $(\mathbf{A}, \mathbf{b})$, given a matrix $\mathbf{A}$. A direct approach (which follows the definition of the LWE problem) is, using a source of randomness, to produce a secret vector $\mathbf{s}$ and a noise vector $\mathbf{e}$ with appropriate distributions, and then to output $\mathbf{b} = \mathbf{A}\mathbf{s}+\mathbf{e}$. This sampler has a particular property: it itself knows the secret $\mathbf{s}$. In a sense, the LWE problem with the vector $\mathbf{b}$ is not hard for the sampler. In that case, we say that an LWE sampler is *witness-aware*. We are interested in samplers that are not witness-aware, i.e., that are *witness-oblivious*.

Below, we discuss instance samplers and knowledge assumptions with a focus on the LWE problem. We start by splitting our discussion about obliviousness between the

classical and quantum settings in Subsections 2.3.1 and 2.3.2. Furthermore, we show in Subsection 2.3.3 how to deduce from a given oblivious sampler another one via reductions. Finally, we show in Subsection 2.3.4 how to design a quantum oblivious sampler for LWE.

2.3.1 Classical Setting

We begin by the definition of a classical LWE sampler.

Definition 9 (Classical LWE Samplers)**.** *Let $m \geq n \geq 1$ and $q \geq 2$ be integers, and χ be a distribution over $\mathbb{Z}/q\mathbb{Z}$. The parameters m, n, q, χ are functions of some security parameter λ. Let $\mathcal{S}$ be a PPT algorithm that has the following specification:*

> *$\mathcal{S}(1^\lambda, \mathbf{A}; r)$: Given as input the security parameter 1^λ (in unary), the matrix $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$ and an auxiliary bit string r of size $\mathsf{poly}(\lambda)$, it returns a pair $(\mathbf{A}, \mathbf{b})$ with $\mathbf{b} \in (\mathbb{Z}/q\mathbb{Z})^m$.*

We say that $\mathcal{S}$ is a classical $\mathsf{LWE}_{m,n,q,\chi}$ sampler if, for a uniformly distributed input matrix $\mathbf{A}$ and a statistically independent random string r, the distribution of $\mathcal{S}(1^\lambda, \mathbf{A}; r)$ is within statistical distance $\mathsf{negl}(\lambda)$ from the distribution of $\mathsf{LWE}_{m,n,q,\chi}$ as given in Definition 8.

As discussed above, some samplers, during their course of execution, might need to produce the witness in order to be successful, namely they are aware of the witness. Assume that we are given the concrete machine that implements the sampler. If we carefully inspect all steps of the machine, the witness must show up at some point (in an easily recoverable way), which allows us to extract it. We grasp this intuition in the following definition.

Definition 10 (Witness-Oblivious LWE Samplers)**.** *Let m, n, q, χ, λ as above. We say that a classical $\mathsf{LWE}_{m,n,q,\chi}$ sampler $\mathcal{S}$ is witness-oblivious if for every PPT extractor $\mathcal{E}$, we have*

$$\mathbb{P}\left[\mathbf{s}' = \mathbf{s} \text{ and } \mathbf{e}' = \mathbf{e} \;\middle|\; \begin{array}{l} \mathbf{A} \leftarrow U\left((\mathbb{Z}/q\mathbb{Z})^{m\times n}\right) \\ r \leftarrow U\left(\{0,1\}^{\mathsf{poly}(\lambda)}\right) \\ (\mathbf{A}, \mathbf{b} := \mathbf{A}\mathbf{s} + \mathbf{e}) \leftarrow \mathcal{S}(1^\lambda, \mathbf{A}; r) \\ (\mathbf{s}', \mathbf{e}') \leftarrow \mathcal{E}(1^\lambda, \mathbf{A}, \mathbf{b}, r) \end{array}\right] \leq \mathsf{negl}(\lambda) \ ,$$

where the probability is also taken over the randomness of $\mathcal{E}$.

This definition implies that given $(\mathbf{A}, \mathbf{b})$, finding a witness is hard for all PPT algorithms.

Lemma 5. *Let m, n, q, χ, λ as above. Suppose that there exists a classical witness-oblivious $\mathsf{LWE}_{m,n,q,\chi}$ sampler. Then the $\mathsf{LWE}_{m,n,q,\chi}$ problem is hard for every PPT algorithm; for all PPT algorithm $\mathcal{B}$, we have*

$$\mathbb{P}\left[\mathbf{s}' = \mathbf{s} \text{ and } \mathbf{e}' = \mathbf{e} \;\middle|\; \begin{array}{l} (\mathbf{A}, \mathbf{b} := \mathbf{A}\mathbf{s} + \mathbf{e}) \leftarrow \mathsf{LWE}_{m,n,q,\chi} \\ (\mathbf{s}', \mathbf{e}') \leftarrow \mathcal{B}(1^\lambda, \mathbf{A}, \mathbf{b}) \end{array}\right] \leq \frac{1}{\mathsf{negl}(\lambda)} \ ,$$

where the probability is also taken over the randomness of $\mathcal{B}$.

Proof. Let $\mathcal{S}$ denote the witness-oblivious sampler and $\mathcal{B}$ be an arbitrary PPT algorithm. By assumption, if given as input a uniformly distributed matrix $\mathbf{A}$, the output distribution of algorithm $\mathcal{S}$ is within statistical distance $\mathsf{negl}(\lambda)$ from the instance distribution of $\mathsf{LWE}_{m,n,q,\chi}$. Therefore, by properties of the statistical distance, we have:

$$\mathbb{P}\left(\mathbf{s}' = \mathbf{s} \text{ and } \mathbf{e}' = \mathbf{e} \;\middle|\; \begin{array}{l} (\mathbf{A}, \mathbf{b} := \mathbf{A}\mathbf{s} + \mathbf{e}) \leftarrow \mathsf{LWE}_{m,n,q,\chi} \\ (\mathbf{s}', \mathbf{e}') \leftarrow \mathcal{B}(1^\lambda, \mathbf{A}, \mathbf{b}) \end{array}\right) \leq$$
$$\mathbb{P}\left(\mathbf{s}' = \mathbf{s} \text{ and } \mathbf{e}' = \mathbf{e} \;\middle|\; \begin{array}{l} \mathbf{A} \leftarrow U\left((\mathbb{Z}/q\mathbb{Z})^{m\times n}\right) \\ r \leftarrow U\left(\{0,1\}^{\mathsf{poly}(\lambda)}\right) \\ (\mathbf{A}, \mathbf{b} := \mathbf{A}\mathbf{s} + \mathbf{e}) \leftarrow \mathcal{S}(1^\lambda, \mathbf{A}; r) \\ (\mathbf{s}', \mathbf{e}') \leftarrow \mathcal{B}(1^\lambda, \mathbf{A}, \mathbf{b}) \end{array}\right) + \mathsf{negl}(\lambda) \ .$$

Define the following PPT algorithm $\mathcal{E}$:

$$\mathcal{E}(1^\lambda, \mathbf{A}, \mathbf{b}, r) := \mathcal{B}(1^\lambda, \mathbf{A}, \mathbf{b}) \ .$$

Therefore, as $\mathcal{S}$ is a classical witness-oblivious sampler, we have

$$\mathbb{P}\left(\mathbf{s}' = \mathbf{s} \text{ and } \mathbf{e}' = \mathbf{e} \;\middle|\; \begin{array}{l} \mathbf{A} \leftarrow U\left((\mathbb{Z}/q\mathbb{Z})^{m\times n}\right) \\ r \leftarrow U(\{0,1\}^{\mathsf{poly}(\lambda)}) \\ (\mathbf{A}, \mathbf{b} := \mathbf{A}\mathbf{s} + \mathbf{e}) \leftarrow \mathcal{S}(1^\lambda, \mathbf{A}; r) \\ (\mathbf{s}', \mathbf{e}') \leftarrow \mathcal{E}(1^\lambda, \mathbf{A}, \mathbf{b}, r) \end{array}\right) \leq \mathsf{negl}(\lambda) \ ,$$

which completes the proof. □

Lemma 5 states that the existence of a witness-oblivious LWE sampler implies the hardness of the LWE problem. We are interested in the converse, i.e., in obtaining an oblivious sampler, under the assumption that LWE is hard.

2.3.2 Quantum Setting

To discuss the post-quantum security of cryptographic schemes, we must migrate to quantum algorithms with appropriate extension of obliviousness. Before going into the details, we need an appropriate definition of quantum samplers.

Definition 11 (Quantum LWE Samplers)**.** *Let $m \geq n \geq 1$ and $q \geq 2$ be integers, and χ be a distribution over $\mathbb{Z}/q\mathbb{Z}$. The parameters m, n, q, χ are functions of some security parameter λ. Let $\mathcal{S}$ be a QPT algorithm that has the following specification:*

> *$\mathcal{S}\left(1^\lambda, \mathbf{A}, |0\rangle^{\mathsf{poly}(\lambda)}\right)$: Given as input the security parameter 1^λ (in unary), the matrix $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, and a polynomial number of ancillas each initialized to $|0\rangle$ as inputs, it returns a pair $(\mathbf{A}, \mathbf{b})$ with $\mathbf{b} \in (\mathbb{Z}/q\mathbb{Z})^m$.*

We say that $\mathcal{S}$ is a quantum $\mathsf{LWE}_{m,n,q,\chi}$ sampler if, for a uniformly distributed input matrix $\mathbf{A}$, the distribution of $\mathcal{S}(1^\lambda, \mathbf{A}, |0\rangle^{\mathsf{poly}(\lambda)})$ is within statistical distance $\mathsf{negl}(\lambda)$ from the distribution of $\mathsf{LWE}_{m,n,q,\chi}$ as given in Definition 8.

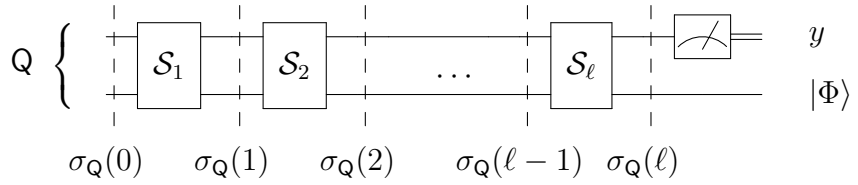


Figure 2.1: The execution of the sampler.

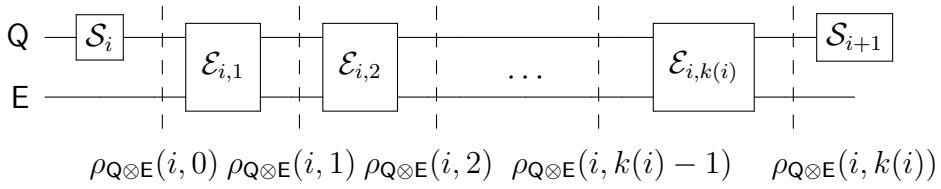


Figure 2.2: The execution of the extractor between Steps i and $i+1$ of the sampler.

The main principle we use to base our obliviousness definition on is that observing or measuring the execution of a machine (be it classical or quantum) must not change the view that the sampler has of itself. Assume that an extractor is observing a sampler. Let $\rho_{\mathsf{Q}\otimes\mathsf{E}}$ represent the joint state of the sampler $\mathcal{S}$ and the extractor $\mathcal{E}$ at some step. The extractor might have carried out particular inspections that ended up in entangling its register with that of the sampler, so the state $\rho_{\mathsf{Q}\otimes\mathsf{E}}$ might not be separable. We intuitively expect from a valid extractor that if we trace out its register, the remaining state must be as if no extractor was inspecting the sampler at all. Namely, if ρ_{Q} was the state of an isolated sampler at the same step, we require that $\mathrm{tr}_{\mathsf{E}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}) = \rho_{\mathsf{Q}}$. We define valid extractors as follows, based on the above discussion, except that we only require that $\mathrm{tr}_{\mathsf{E}}(\rho_{\mathsf{Q}\otimes\mathsf{E}})$ and ρ_{Q} are close for the trace distance.

Definition 12. *Let* Q *and* E *be two quantum registers initialized to* τ_{Q} *and* τ_{E} *where* τ_{Q} *consists of classical information and ancillas while* τ_{E} *consists of only ancillas. Let* $\mathcal{G}$ *be the set of one-qubit and two-qubit unitary gates. Let* $\mathcal{S}$ *be a quantum algorithm operating on register* Q *with gates* $\mathcal{S}_1, \ldots, \mathcal{S}_\ell$ *each of which either belongs to* $\mathcal{G}$ *or is a measurement in the computational basis. Let* $\mathcal{E}$ *be a quantum algorithm operating on the joint register* $\mathsf{Q}\otimes\mathsf{E}$ *with the gates* $(\mathcal{E}_{0,j})_{j\leq k(0)}, (\mathcal{E}_{1,j})_{j\leq k(1)}, \ldots, (\mathcal{E}_{\ell+1,j})_{j\leq k(\ell+1)}$ *each of which either belongs to* $\mathcal{G}$ *or is a measurement in the computational basis.*

In the first scenario, suppose that $\mathcal{S}$ *is operating alone on* Q*. Let* $\sigma_{\mathsf{Q}}(i)$ *be the density matrix representing the state of* Q *just after the* i*-th step of* $\mathcal{S}$ *for* $i \geq 1$ *and just before the first step of* $\mathcal{S}$ *for* $i = 0$*, as depicted in Figure 2.1.*

In the second scenario, suppose that $\mathcal{S}$ *and* $\mathcal{E}$ *are operating jointly on the registers* Q *and* E *as follows. After the* i*-th step of* $\mathcal{S}$ *for* $i \geq 1$ *and before the first step of* $\mathcal{S}$ *for* $i = 0$*, algorithm* $\mathcal{E}$ *is given both registers to perform its operations* $(\mathcal{E}_{i,j})_{j\leq k(i)}$ *and sends register* Q *to* $\mathcal{S}$*. For every* $1 \leq j \leq k(i)$*, let* $\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j)$ *denote the joint state of the registers after applying* $\mathcal{E}_{i,j}$*, and let* $\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,0)$ *denote the state just before applying* $\mathcal{E}_i$*, as depicted in Figure 2.2.*

Let $\varepsilon \geq 0$ *be a real number. We say that* $\mathcal{E}$ *is an* ε*-valid extractor if for every* $0 \leq i \leq \ell$ *and every* $0 \leq j \leq k(i)$*, it holds that:*

$$D_{\mathrm{tr}}\left(\mathrm{tr}_{\mathsf{E}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j)), \sigma_{\mathsf{Q}}(i)\right) \leq \varepsilon \ . \tag{2.4}$$

We say that an extractor is perfect if it is ε-valid for $\varepsilon = 0$. Furthermore, we let $\langle \mathcal{S}, \mathcal{E} \rangle(\tau_{\mathsf{Q}}, \tau_{\mathsf{E}})$ denote the joint output.

We note that this definition does not assume that $\mathcal{S}$ is a sampler, nor that $\mathcal{S}$ and $\mathcal{E}$ are efficient.

This definition covers all valid extractors in the classical setting. A classical sampler only exploits classical registers. Observing and copying the internal states and the randomness encoded in classical registers is perfectly doable. This translates to the extractor having all the information of internal states and randomness of the sampler. This gives exactly the same information to the extractor as in Definition 10.

Definition 13 (Witness-Oblivious Quantum Samplers)**.** *Let m, n, q, χ, λ as in Definition 11. We say that a quantum $\mathsf{LWE}_{m,n,q,\chi}$ sampler $\mathcal{S}$ is witness-oblivious if for every $\mathsf{negl}(\lambda)$-valid QPT extractor $\mathcal{E}$, we have*

$$\mathbb{P}\left(\mathbf{s}' = \mathbf{s} \text{ and } \mathbf{e}' = \mathbf{e} \;\middle|\; \begin{array}{l} \mathbf{A} \leftarrow U\left((\mathbb{Z}/q\mathbb{Z})^{m\times n}\right) \\ ((\mathbf{A}, \mathbf{b} := \mathbf{A}\mathbf{s} + \mathbf{e}), (\mathbf{s}', \mathbf{e}')) \leftarrow \langle \mathcal{S}, \mathcal{E} \rangle \left((1^\lambda, \mathbf{A}, |\mathbf{0}\rangle), |\mathbf{0}\rangle\right) \end{array}\right) \leq \mathsf{negl}(\lambda) \ ,$$

where $|\mathbf{0}\rangle := |0^{\mathsf{poly}(\lambda)}\rangle$, and the probability is also taken over the measurements of $\mathcal{S}$ and $\mathcal{E}$.

We note that a statement similar to Lemma 5 holds for quantum witness-oblivious samplers.

Relation to the definition from [LMZ23]. The authors of [LMZ23] adopted a different approach to define valid extractors. Their definition only deals with unitary algorithms followed by a single final measurement. The sampler is first executed until it performs its final measurement, and then the remaining working register and the measurement outcome are handed over to the extractor. The extractor is not allowed to inspect or observe the sampler during its execution. Using the notations of Figure 2.1, a valid extractor, in their case, is only given the classical output y, and the quantum output $|\Phi\rangle$.

Definition 14 (Adapted from [LMZ23])**.** *Let $\mathcal{S}$ be a unitary algorithm with a pure initial state and some classical string as input. Assume that $\mathcal{S}$ performs a final measurement in the computational basis over part of its register. A quantum algorithm $\mathcal{E}$ is said to be an LMZ extractor for $\mathcal{S}$ if it operates as follows: in the first phase, the sampler runs its circuit and outputs $|y, \Phi\rangle$ where y is classical and $|\Phi\rangle$ is quantum; in the second phase, the circuit of the extractor runs over $|y, \Phi\rangle$, and possibly extra ancillas, and outputs a classical string.*

We stress that in the definition above, the extractor is not allowed to engage in the first phase: it proceeds *after* the sampler. We show in the following lemma that, when restricted to unitary algorithms with a pure initial state, our definition of perfect extractor is equivalent to the one from [LMZ23].

Lemma 6. *Let Q be a quantum register initialized with some classical string z and with the pure state $|0\rangle_{\mathsf{Q}}$. Let E be a quantum register with pure initial states $|0\rangle_{\mathsf{E}}$. Let $\mathcal{S}$ be a quantum algorithm operating on Q by a series of unitary gates followed by a single measurement. Let $\mathcal{E}$ be a QPT ε-valid extractor operating on two registers Q and E as*

per Definition 12. Then, there exists an LMZ extractor $\mathcal{E}'$ (as per Definition 14) that is QPT, and

$$D_{\mathrm{tr}}\Big(\langle \mathcal{S}, \mathcal{E}\rangle\left((z, |0\rangle_{\mathsf{Q}}), |0\rangle_{\mathsf{E}}\right) \ , \ \mathcal{E}'\left(\mathcal{S}\left(z, |0\rangle_{\mathsf{Q}}\right)\right)\Big) \leq 2\sqrt{2\varepsilon} \ .$$

In the case of quantum LWE samplers, the classical string z will be 1^λ and $\mathbf{A}$.

Lemma 7. *Using notations of Lemma 6, for any (i, j), it holds that*

$$D_{\mathrm{tr}}\left(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j), \sigma_{\mathsf{Q}}(i) \otimes \mathcal{E}'_{(i,j)}\left(|0\rangle\right)\right) \leq 2\sqrt{2\varepsilon} \ ,$$

where $\mathcal{E}'_{(i,j)}$ is a QPT algorithm given as input a quantum state $|0\rangle$, and implicitly the description of $\mathcal{S}$ and $\mathcal{E}$ as well as the classical string z.

Proof. Let us first prove the statement for the perfect-case, i.e., $\varepsilon = 0$. We prove it by induction on i. Suppose that the statement holds for $(i-1) \geq 0$ (it clearly holds for $i = 0$ as at this step neither $\mathcal{S}$ nor $\mathcal{E}$ performs any computations). In particular, we have

$$\rho_{\mathsf{Q}\otimes\mathsf{E}}(i-1, k(i-1)) = \sigma_{\mathsf{Q}}(i-1) \otimes \mathcal{E}'_{(i-1,k(i-1))}(|0\rangle) \ .$$

The statement holds for $(i, 0)$ by definition, namely,

$$\begin{aligned}\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,0) &= \mathcal{S}_i \sigma_{\mathsf{Q}}(i-1) \otimes \mathcal{E}'_{(i-1,k(i-1))}(|0\rangle) \\ &= \sigma_{\mathsf{Q}}(i) \otimes \mathcal{E}'_{(i-1,k(i-1))}(|0\rangle) \ .\end{aligned}$$

Note that $\mathcal{E}$ is a perfect extractor, therefore according to Definition 12, we have

$$\mathrm{tr}_{\mathsf{E}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j)) = \sigma_{\mathsf{Q}}(i) \ .$$

The state $\sigma_{\mathsf{Q}}(i)$ is pure since $\mathcal{S}$ only applies unitaries to Q which initially contains the pure quantum state $|0\rangle_{\mathsf{Q}}$. Therefore, according to the above equality, $\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j)$ is a product state. Furthermore, it is necessarily given by

$$\sigma_{\mathsf{Q}}(i) \otimes \rho_{\mathsf{E}} \ ,$$

where ρ_{E} is the quantum state obtained after applying $\mathcal{E}_{i,1}, \ldots, \mathcal{E}_{i,j}$ as in Figure 2.2 on

$$\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,0) = \sigma_{\mathsf{Q}}(i) \otimes \mathcal{E}'_{(i-1,k(i-1))}(|0\rangle) \ ,$$

and then tracing out Q. On one hand, $\sigma_{\mathsf{Q}}(i)$ can be computed via i steps of $\mathcal{S}$ given the classical string z and the quantum state $|0\rangle_{\mathsf{Q}}$ where no measurement are performed. Therefore, given the description of $\mathcal{S}$, we can compute a polynomial time unitary $\mathbf{U}$ (the sampler $\mathcal{S}$ is QPT) such that $\sigma_{\mathsf{Q}}(i) = \mathbf{U}|0\rangle$. On the other hand, the quantum state $\rho_{\mathsf{E}\otimes\mathsf{Q}}(i,j)$ is equal, by definition, to

$$\begin{aligned}\rho_{\mathsf{E}\otimes\mathsf{Q}}(i,j) &= \mathcal{E}_{i,j} \ldots \mathcal{E}_{i,1}\left(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,0)\right) \\ &= \mathcal{E}_{i,j} \ldots \mathcal{E}_{i,1}\left(\sigma_{\mathsf{Q}}(i) \otimes \mathcal{E}'_{(i-1,k(i-1))}(|0\rangle)\right) \\ &= \mathcal{E}_{i,j} \ldots \mathcal{E}_{i,1}\left(\mathbf{U}|0\rangle_{\mathsf{Q}} \otimes \mathcal{E}'_{(i-1,k(i-1))}(|0\rangle)\right) \ .\end{aligned}$$

The algorithm $\mathcal{E}'_{(i,j)}$ computes the above state and then it traces out its first register Q, namely it keeps only the second register E. It shows that the lemma holds when $\varepsilon = 0$ for any i and any $0 \leq j \leq k(i)$. It concludes the proof by induction in this case.

Now suppose that $\varepsilon > 0$. Let us consider the same algorithm $\mathcal{E}'_{(i,j)}$ as above. However, in this case (when keeping the second register) it is not true anymore that, $\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j) = \sigma_{\mathsf{Q}}(i) \otimes \mathcal{E}'_{(i,j)}(|0\rangle)$. Indeed, $\mathrm{tr}_{\mathsf{E}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j))$ is not a pure state, therefore $\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j)$ is not a product state. To handle this case, let us introduce the fidelity $F(\cdot,\cdot)$ between quantum states, i.e., for all quantum states ρ and σ

$$F(\rho,\sigma) := \mathrm{tr}\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\ .$$

By Uhlmann's theorem [NC11, Th. 9.14, Exercise 9.15], there exists some purifications $|\varphi\rangle$ and $|\psi\rangle$ of $\mathrm{tr}_{\mathsf{E}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j))$ and $\sigma_{\mathsf{Q}}(i)$, respectively, such that

$$F(\mathrm{tr}_{\mathsf{E}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j)), \sigma_{\mathsf{Q}}(i)) = |\langle\varphi|\psi\rangle|\ . \tag{2.5}$$

Note that by definition: $\mathrm{tr}_{\mathsf{E}}(|\psi\rangle\langle\psi|) = \sigma_{\mathsf{Q}}(i)$ which is a pure state. Therefore $|\psi\rangle$ is a product state, in particular

$$|\psi\rangle = \sigma_{\mathsf{Q}}(i) \otimes \rho\ .$$

By Fuchs-van de Graaf inequalities, [NC11, Eq. 9.110], it holds that

$$\begin{aligned} F(\mathrm{tr}_{\mathsf{E}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j)), \sigma_{\mathsf{Q}}(i)) &\geq 1 - D_{\mathrm{tr}}(\mathrm{tr}_{\mathsf{E}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j)), \sigma_{\mathsf{Q}}(i)) \\ &\geq 1 - \varepsilon\ . \end{aligned}$$

Therefore, by using Equation (2.5), we have

$$|\langle\varphi|\psi\rangle| \geq 1 - \varepsilon$$

which implies that

$$D_{\mathrm{tr}}(|\varphi\rangle, |\psi\rangle) \leq \sqrt{1-(1-\varepsilon)^2} \leq \sqrt{2\varepsilon}\ .$$

The above inequality holds for any purification $|\varphi\rangle$ of $\mathrm{tr}_{\mathsf{E}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j))$, in particular for $\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j)$ (without loss of generality we can suppose that it is a pure quantum after some purification), namely,

$$D_{\mathrm{tr}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j), |\psi\rangle) \leq \sqrt{1-(1-\varepsilon)^2} \leq \sqrt{2\varepsilon}\ . \tag{2.6}$$

Recall now that $|\psi\rangle = \sigma_{\mathsf{Q}}(i)\otimes\rho$. In its last step, algorithm $\mathcal{E}'_{(i,j)}(|0\rangle)$ keeps only the second register E of $\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j)$ (it traces out its first register Q). Therefore, by properties of the trace distance,

$$D_{\mathrm{tr}}\left(\rho, \mathcal{E}'_{(i,j)}(|0\rangle)\right) \leq \sqrt{2\varepsilon}\ . \tag{2.7}$$

By using the triangular inequality,

$$\begin{aligned} D_{\mathrm{tr}}\left(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j), \sigma_{\mathsf{Q}}(i)\otimes\mathcal{E}'_{(i,j)}(|0\rangle)\right) &\leq D_{\mathrm{tr}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j), |\psi\rangle) + D_{\mathrm{tr}}\left(|\psi\rangle, \sigma_{\mathsf{Q}}(i)\otimes\mathcal{E}'_{(i,j)}(|0\rangle)\right) \\ &= D_{\mathrm{tr}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j), |\psi\rangle) + D_{\mathrm{tr}}\left(\sigma_{\mathsf{Q}}(i)\otimes\rho, \sigma_{\mathsf{Q}}(i)\otimes\mathcal{E}'_{(i,j)}(|0\rangle)\right) \\ &= D_{\mathrm{tr}}(\rho_{\mathsf{Q}\otimes\mathsf{E}}(i,j), |\psi\rangle) + D_{\mathrm{tr}}\left(\rho, \otimes\mathcal{E}'_{(i,j)}(|0\rangle)\right) \\ &\leq 2\sqrt{2\varepsilon} \end{aligned}$$

where we used Equations (2.6) and (2.7). This completes the proof. □

Proof of Lemma 6. Let ℓ be the number of steps of the sampler $\mathcal{S}$. Recall that after the ℓ-th step, the sampler $\mathcal{S}$ measures part of the register Q. According to Lemma 7, we have

$$D_{\mathrm{tr}}\left(\rho_{\mathsf{Q}\otimes\mathsf{E}}(\ell,k(\ell)),\sigma_{\mathsf{Q}}(\ell)\otimes\mathcal{E}'_{\ell,k(\ell)}\left(|0\rangle\right)\right)\leq 2\sqrt{2\varepsilon}\ ,$$

where $\mathcal{E}'_{\ell,k(\ell)}$ is a QPT algorithm that only needs the description of $\mathcal{S}$, $\mathcal{E}$, and the knowledge of the classical string z to run. Therefore, after performing the measurement of $\mathcal{S}$ on $\sigma_{\mathsf{Q}}(\ell)\otimes\mathcal{E}'_{\ell,k(\ell)}\left(|0\rangle\right)$, the post-measurement state is within trace distance $\leq 2\sqrt{2\varepsilon}$ from the post-measurement state after applying the same measurement on $\rho_{\mathsf{Q}\otimes\mathsf{E}}(\ell,k(\ell))$. Let the former be

$$|y,\Phi\rangle\otimes\mathcal{E}'_{\ell,k(\ell)}\left(|0\rangle\right)\ ,$$

where y is classical. To build an LMZ extractor, it suffices to remove the interaction between $\mathcal{E}$ and $\mathcal{S}$. We first run $\mathcal{S}$ once and let it perform its measurement to obtain $|y,\Phi\rangle$. Then, we let $\mathcal{E}$ perform its last steps over $|y,\Phi\rangle\otimes\mathcal{E}'_{\ell,k(\ell)}\left(|0\rangle\right)$. Note that this defines an LMZ extractor since $\mathcal{E}'_{\ell,k(\ell)}\left(|0\rangle\right)$ can be computed in polynomial time and independent of (i) the execution of $\mathcal{S}$ and (ii) its measurement output.

□

2.3.3 Obliviousness and black-box reductions

All definitions of this subsection can be extended to the general class of *distributional problems* as well. Recall that a distributional problem P is a pair (R,D) where R is an NP relation and $\mathrm{D}=\{\mathrm{D}_\lambda\}_\lambda$ is a polynomially sampleable ensemble over the instances of R. The problem P asks for finding a witness for an instance that has been sampled according to D. We note that the search LWE problem belongs to this class.

Definition 15 (Quantum Samplers)**.** *Let λ be the security parameter. Let $\mathrm{P}=(\mathrm{R},\mathrm{D})$ be a distributional problem. Let $\mathcal{S}$ be a QPT algorithm that has the following specification:*

> *$\mathcal{S}\left(1^\lambda,\tilde{x},|0\rangle^{\mathsf{poly}(\lambda)}\right)$: Given the parameter 1^λ, a string $\tilde{x}$, and a polynomial number of ancillas initialized to $|0\rangle$ as inputs, it returns a string x of size $\mathsf{poly}(\lambda)$ that has $\tilde{x}$ as a prefix.*

We say that $\mathcal{S}$ is a quantum P *sampler if there exists a probability distribution $\{\widetilde{D}_\lambda\}_\lambda$ such that for $\tilde{x}$ sampled from $\widetilde{D}_\lambda$, the distribution of $\mathcal{S}\left(1^\lambda,\tilde{x},|0\rangle^{\mathsf{poly}(\lambda)}\right)$ is within statistical distance $\mathsf{negl}(\lambda)$ from D_λ.*

One can define witness-oblivious samplers by adapting Definition 13. We are interested in preservation of witness-obliviousness under reductions. We begin by recalling the definition of reductions with respect to distributional problems.

Definition 16. *A distributional problem $\mathrm{P}_1=(\mathrm{R}_1,\mathrm{D}_1)$ is randomized Karp-reducible to $\mathrm{P}_2=(\mathrm{R}_2,\mathrm{D}_2)$ if there exists:*

- *a PPT algorithm $\mathcal{A}$ that maps instances of P_1 to instances of P_2 such that $\mathcal{A}(\mathrm{D}_1)$ is within negligible statistical distance from D_2 over the randomness of $\mathcal{A}$,*

- *a PPT or QPT algorithm $\mathcal{B}$ for $\mathcal{A}$ such that*

$$\forall x_1, y_2 : (\mathcal{A}(x_1; r), y_2) \in \mathrm{R}_2 \implies (x_1, \mathcal{B}(x_1, y_2, r)) \in \mathrm{R}_1 \ ,$$

with non-negligible probability over the randomness of $\mathcal{B}$. Note that $\mathcal{B}$ has the randomness r of $\mathcal{A}$ as part of its input (and can use extra randomness).

The following theorem states that witness-obliviousness is preserved under randomized Karp reductions.

Lemma 8. *Let P_1 and P_2 be two distributional problems. Assume that P_1 is randomized Karp-reducible to P_2 with the associated algorithms $\mathcal{A}$ and $\mathcal{B}$. If $\mathcal{S}$ is a quantum witness-oblivious P_1 sampler, then $\mathcal{A}(\mathcal{S})$ is a quantum witness-oblivious P_2 sampler.*

Proof. Let $x_1 \leftarrow \mathcal{S}$ and $x_2 \leftarrow \mathcal{A}(x_1; r)$. Suppose that there exists a valid QPT extractor $\mathcal{E}_2$ that finds a witness y_2 for the instance x_2. One can build a new extractor $\mathcal{E}_1$ for $\mathcal{S}$ as follows. To find a witness for x_1, the new extractor (i) collects the randomness r of $\mathcal{A}$, (ii) finds the witness y_2 for x_2 using $\mathcal{E}_2$, and then (iii) applies $\mathcal{B}(x_1, y_2, r)$. The output of $\mathcal{B}$ is a witness for x_1 according to the definition of the randomized Karp reduction. It suffices to note that $\mathcal{B}$ is indeed a valid extractor for $\mathcal{Q}$. □

Note that the P_2 sampler is witness-oblivious under the hardness assumption of P_1. Many classical reductions in the context of lattice problems fall into the above framework.

2.3.4 Reducing oblivious LWE sampling to $\mathsf{C}\,|\mathsf{LWE}\rangle$.

We complete this section by providing a general approach to design a quantum witness oblivious sampler via a single unitary and a final measurement. We show that producing LWE samples in an oblivious manner reduces to synthesizing a quantum state that is a superposition of all LWE samples, as defined in [CLZ22]. This state synthesis problem is called the $\mathsf{C}\,|\mathsf{LWE}\rangle$ problem.

Definition 17 ($\mathsf{C}\,|\mathsf{LWE}\rangle$ State)**.** *Let $m \geq n \geq 1$ and $q \geq 2$ be integers, and f be an amplitude function whose domain is $\mathbb{Z}/q\mathbb{Z}$. The parameters m, n, q, f are functions of some security parameter λ. For $\mathbf{A} = (\mathbf{a}_1 | \dots | \mathbf{a}_m)^\intercal \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$, the $\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}$ state is defined as*

$$\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f} := \frac{1}{\sqrt{Z_f(\mathbf{A})}} \sum_{\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n} \sum_{\mathbf{e} \in (\mathbb{Z}/q\mathbb{Z})^m} \bigotimes_{i=1}^{m} f(e_i)\, |\langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \bmod q\rangle \ ,$$

where $Z_f(\mathbf{A})$ is the normalization scalar such that $\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}$ becomes a unit vector.

To simplify notation, when it is clear from the context, we will drop the dependency on m, n, q, f, and the matrix $\mathbf{A}$.

The normalization term $Z_f(\mathbf{A})$, which guarantees that $\mathsf{C}\,|\mathsf{LWE}\rangle$ is a *valid* quantum state, will play an important role. In particular, we will require that $Z_f(\mathbf{A}) \approx q^n$. We will discuss this matter in detail in Section 2.5, when instantiating our algorithm to the case where $|f|^2$, i.e., the noise distribution of the measured LWE sample, is a Gaussian distribution.

Constructing this state was studied in [CLZ22] in order to solve the *Short Integer Solution* (SIS) problem with some specific parameters. We note that [CLZ22] neglected the normalization factor $Z_f(\mathbf{A})$ by assuming that it is always equal to q^n, see for example [CLZ22, Def. 9 & Cor. 9]. In the general problem of constructing an $\mathsf{C}\,|\mathsf{LWE}\rangle$ state, one should take the normalization into account. For instance, it was shown in [DRT21] that this normalization factor should be handled with care when $|f|^2$ concentrates the error weight close to the minimum distance of the spanned linear code.

We also stress that [CLZ22, Def. 9] only allows non-negative real-valued amplitude functions, while we allow complex-valued ones. Although we only use real-valued (but not positive) instantiations of the amplitude function in this work since they are sufficient for our purposes, more choices of the function might have further applications.

Definition 18 ($\mathsf{C}\,|\mathsf{LWE}\rangle$ Problem)**.** *Let m, n, q, f, λ as in Definition 17. The $\mathsf{C}\,|\mathsf{LWE}\rangle_{m,n,q,f}$ problem is as follows: given as input a matrix $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, the goal is to build the $\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}$ state. More formally, we say that a QPT algorithm $\mathcal{S}$ solves $\mathsf{C}\,|\mathsf{LWE}\rangle_{m,n,q,f}$ if there exists $M \leq \mathsf{poly}(\lambda)$ such that given 1^λ, a uniform $\mathbf{A}$ and $|0\rangle^{m\log q}\,|0\rangle^M$ as inputs, then algorithm $\mathcal{S}$ builds a quantum state within trace distance $\mathsf{negl}(\lambda)$ from $\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}\,|0\rangle^M$, with probability $1-\mathsf{negl}(\lambda)$ over the randomness of $\mathbf{A}$ and its measurements.*

Notice that measuring the $\mathsf{C}\,|\mathsf{LWE}\rangle$ state gives the following m LWE samples:

$$((\mathbf{a}_1, \langle \mathbf{a}_1, \mathbf{s}\rangle + e_1 \bmod q), \ldots, (\mathbf{a}_m, \langle \mathbf{a}_m, \mathbf{s}\rangle + e_m \bmod q)) \ ,$$

where the e_i's are i.i.d. with distribution $|f|^2$, while $\mathbf{s}$ is uniform and independent.

In the following theorem, we show that solving $\mathsf{C}\,|\mathsf{LWE}\rangle$ using a unitary algorithm provides a witness-oblivious LWE sampler by measuring the final superposition. We stress that the result holds even if the $\mathsf{C}\,|\mathsf{LWE}\rangle$ solver only provides a state that is only approximately equal (in trace distance) to $\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}\,|0\rangle^M$.

Theorem 2. *Let m, n, q, f, λ as in Definition 17. Assume that there exists a unitary QPT algorithm $\mathcal{S}$ that solves $\mathsf{C}\,|\mathsf{LWE}\rangle_{m,n,q,f}$ for some $M \leq \mathsf{poly}(\lambda)$ number of auxiliary ancillas as input. Then $\mathcal{S}$ followed by a measurement in the computational basis is a witness-oblivious quantum $\mathsf{LWE}_{m,n,q,|f|^2}$ sampler, assuming the quantum hardness of $\mathsf{LWE}_{m,n,q,|f|^2}$.*

Proof. Let $\mathsf{Q} = \mathsf{C} \otimes \mathsf{W}$ be the register of $\mathcal{S}$ such that the final measurement is performed upon C to obtain the classical output and W is the remaining register. Let $|\psi\rangle$ be the final state of the algorithm $\mathcal{S}$ over Q, right before the measurement. With probability $1-\mathsf{negl}(\lambda)$ over the uniform choice of $\mathbf{A}$, we have:

$$D_{\mathrm{tr}}\left(|\psi\rangle\ ,\ \mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}\,|0\rangle^M\right) \leq \mathsf{negl}(\lambda)\ . \tag{2.8}$$

After applying the measurement, the state $|\psi\rangle$ becomes a mixed state as follows:

$$\sigma_{\mathcal{S}} := \sum_{\mathbf{b}\in(\mathbb{Z}/q\mathbb{Z})^m} p_{\mathbf{b}}\,|\mathbf{b}\rangle\langle\mathbf{b}| \otimes |\phi_{\mathbf{b}}\rangle\langle\phi_{\mathbf{b}}| \ ,$$

where $p_{\mathbf{b}}$ is the probability of observing $\mathbf{b}$ as the outcome, and $|\phi_{\mathbf{b}}\rangle$ is the corresponding state in the working space. After the measurement, the other state becomes:

$$\sigma_{\mathsf{LWE}} := \sum_{\mathbf{b}\in(\mathbb{Z}/q\mathbb{Z})^m} q_{\mathbf{b}}\,|\mathbf{b}\rangle\langle\mathbf{b}| \otimes |\mathbf{0}\rangle\langle\mathbf{0}| \ ,$$

where $\{q_{\mathbf{b}}\}_{\mathbf{b}}$ is the induced distribution of $\mathsf{LWE}_{m,n,q,|f|^2}$ over its support, namely

$$q_{\mathbf{b}} = \mathbb{P}_{\mathbf{s},\mathbf{e}}\left(\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{b}\right) \ , \tag{2.9}$$

where $\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n$ is picked uniformly at random and the e_i's are i.i.d. with distribution $|f|^2$. Using the properties of trace distance, we obtain for a proportion $1 - \mathsf{negl}(\lambda)$ of matrices $\mathbf{A}$:

$$\begin{aligned} D_{\mathrm{tr}}\left(\sigma_{\mathcal{S}}, \sigma_{\mathsf{LWE}}\right) &\leq D_{\mathrm{tr}}\left(|\psi\rangle\,,\ \mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f} \otimes |0\rangle^M\right) \\ &\leq \mathsf{negl}(\lambda) \ , \end{aligned} \tag{2.10}$$

where in the last line we used Equation (2.8). Using now Equation (2.3), we obtain for a proportion $1 - \mathsf{negl}(\lambda)$ of matrices $\mathbf{A}$:

$$\Delta\left(\{p_{\mathbf{b}}\}_{\mathbf{b}}, \{q_{\mathbf{b}}\}_{\mathbf{b}}\right) \leq \mathsf{negl}(\lambda) \ .$$

This proves, by using Equation (2.9), that the sampler $\mathcal{S}$ is a quantum LWE sampler as stated in Definition 11.

Let us now show that $\mathcal{S}$ followed be a single measurement is a *witness-oblivious* quantum sampler as stated in Definition 13. By assumption, the sampler $\mathcal{S}$ is a unitary algorithm. We first consider the case where the extractor $\mathcal{E}$ is perfect, i.e., $\varepsilon = 0$ in Lemma 6. Therefore, we can suppose that the input of $\mathcal{E}$ is $\sigma_{\mathcal{S}}$.

Suppose that $\mathcal{E}$ is instead given σ_{LWE}, namely $|\mathbf{b}\rangle\,|0\rangle^M$ with $\mathbf{b} := \mathbf{A}\mathbf{s} + \mathbf{e} \in (\mathbb{Z}/q\mathbb{Z})^m$ such that it has been picked according to $q_{\mathbf{b}}$ given in Equation (2.9). In that case, for a uniform choice of matrices $\mathbf{A}$, its probability to output $(\mathbf{s}', \mathbf{e}')$ such that $\mathbf{s}' = \mathbf{s}$ and $\mathbf{e}' = \mathbf{e}$ is $\mathsf{negl}(\lambda)$ as we assumed the quantum hardness of $\mathsf{LWE}_{m,n,q,|f|^2}$. However the extractor is given $\sigma_{\mathcal{S}}$. Using the properties of the trace distance, it holds that for a proportion $1 - \mathsf{negl}(\lambda)$ of matrices $\mathbf{A}$:

$$\begin{aligned} D_{\mathrm{tr}}\Big(\mathcal{E}\left(\sigma_{\mathcal{S}}\right), \mathcal{E}\left(\sigma_{\mathsf{LWE}}\right)\Big) &\leq D_{\mathrm{tr}}(\sigma_{\mathcal{S}}, \sigma_{\mathsf{LWE}}) \\ &\leq \mathsf{negl}(\lambda) \ . \end{aligned}$$

where in the last line we used Equation (2.10). This completes the proof in the perfect case, i.e., $\varepsilon = 0$.

Now, suppose that $\mathcal{E}$ is $\mathsf{negl}(\lambda)$-valid extractor. According to Lemma 6, it is given a quantum state a trace distance $2\sqrt{2\mathsf{negl}(\lambda)} = \mathsf{negl}(\lambda)$ from $\sigma_{\mathcal{S}}$. To conclude the proof we proceed as above. □

As a direct application of Theorem 2 and Lemma 8, we obtain that a unitary solver for $\mathsf{C}\,|\mathsf{LWE}\rangle_{m,n,q,f}$ gives an witness-oblivious quantum $\mathsf{LWE}_{m',n,q,|f|^2}$ sampler for any integer $m' \in [n, m]$. Indeed, throwing away the superfluous coordinates is a Karp reduction.

2.4 An algorithm for $\mathsf{C}\,|\mathsf{LWE}\rangle$

In Subsection 2.3.4, we have shown that witness-oblivious sampling reduces to the $\mathsf{C}\,|\mathsf{LWE}\rangle$ problem (Definition 18). Solving this problem consists in building the $\mathsf{C}\,|\mathsf{LWE}\rangle$ state (as

per Definition 17). This state is an m-fold tensor product where each element corresponds to a single LWE sample $\langle \mathbf{a}_i, \mathbf{s}\rangle + e_i \bmod q$. Like in [CLZ22], our approach to solve the $\mathsf{C}\,|\mathsf{LWE}\rangle$ problem singles out each of these elements, analyzes them independently, and finally recombines the results.

Definition 19 (Coordinate States)**.** *Let $q \geq 2$ and $f : \mathbb{Z}/q\mathbb{Z} \to \mathbb{C}$ be an amplitude function. We define the coordinate states as follows:*

$$\forall j \in \mathbb{Z}/q\mathbb{Z}, \quad |\psi_j\rangle := \sum_{e=0}^{q-1} f(e)\,|j + e \bmod q\rangle \ .$$

2.4.1 Description of the algorithm

Before going into the details, we briefly explain how our algorithm solves the $\mathsf{C}\,|\mathsf{LWE}\rangle$ problem for some arbitrary amplitude function f. It proceeds in three general phases that would ideally work as follows.

Phase A. First, it builds the following entangled state

$$\frac{1}{\sqrt{q^n}} \sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n} \sum_{\mathbf{e}\in(\mathbb{Z}/q\mathbb{Z})^m} f^{\otimes m}(\mathbf{e})\,|\mathbf{s}\rangle\,|\mathbf{A}\mathbf{s} + \mathbf{e}\rangle = \frac{1}{\sqrt{q^n}} \sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n} |\mathbf{s}\rangle \bigotimes_{j=1}^{m} \left|\psi_{\langle \mathbf{a}_j,\mathbf{s}\rangle}\right\rangle \ . \quad (2.11)$$

The efficiency of this step depends on the specific choice of f.

Phase B. For each j in parallel, it recovers $\langle \mathbf{a}_j, \mathbf{s}\rangle$ from $\left|\psi_{\langle \mathbf{a}_j,\mathbf{s}\rangle}\right\rangle$ with some probability p (independent from j). If it fails, the outcome could be thought as special symbol $\perp$. This operation is not allowed to "perturb" $\left|\psi_{\langle \mathbf{a}_j,\mathbf{s}\rangle}\right\rangle$: it has to be reversible. We handle this by applying some polynomial-time unitary that maps $\left|\psi_{\langle \mathbf{a}_j,\mathbf{s}\rangle}\right\rangle$ to the state

$$\sqrt{p}\,|\langle \mathbf{a}_j, \mathbf{s}\rangle\rangle\,|0\rangle + \sqrt{1-p}\,|a\rangle\,|1\rangle \ ,$$

for some $|a\rangle$ which does not play any role. We interpret any quantum state whose last qubit is $|1\rangle$ as $\perp$. The quality of that step is quantified by the success probability p.

Phase C. Using the successful coordinates, the algorithm collects some linear equations $\langle \mathbf{a}_j, \mathbf{s}\rangle$ (for known $\mathbf{a}_j$'s). The next step is to recompute $\mathbf{s}$ by Gaussian elimination. This allows to erase it from the content of the first register, i.e., disentangling the state in Equation (2.11) and solving the $\mathsf{C}\,|\mathsf{LWE}\rangle$ problem. However, note that Phase B only enables to recover each $\langle \mathbf{a}_j, \mathbf{s}\rangle$ with some probability p. Therefore our approach will work if the number of non-$\perp$ coordinates is no smaller than n in order to expect to have a non-singular linear system to solve, namely if $m = (n + \log\log q)/p \cdot \omega(\log \lambda)$. Therefore, the success probability p considered at Phase B has to be sufficiently large for the purpose of efficiency.

Combining the steps above, one obtains Algorithm 1. Steps 1 to 4 of Algorithm 1 correspond to Phase A above, Steps 5 and 6 correspond to Phase B above, and Steps 7 and 8 correspond to Phase C above.

Algorithm 1 Quantum $\mathsf{C}\left|\mathsf{LWE}(\mathbf{A})\right\rangle_{m,n,q,f}$ Solver.

Parameters: m, n, q and f.

Input: $\mathbf{A} := (\mathbf{a}_1 | \dots | \mathbf{a}_m)^\mathsf{T} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$.
Output: A quantum state $|\varphi\rangle$.

1: Build the state $\frac{1}{\sqrt{q^n}} \sum_{\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n} |\mathbf{s}\rangle$.

2: Build the state $\sum_{\mathbf{e} \in (\mathbb{Z}/q\mathbb{Z})^m} \bigotimes_{i=1}^{m} f(e_i) |e_i\rangle$.

3: Consider the joint state of Steps 1 and 2 to get

$$\frac{1}{\sqrt{q^n}} \sum_{\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n} |\mathbf{s}\rangle \sum_{\mathbf{e} \in (\mathbb{Z}/q\mathbb{Z})^m} \bigotimes_{i=1}^{m} f(e_i) |e_i\rangle \ .$$

4: Apply the quantum unitary $|\mathbf{s}, \mathbf{e}\rangle \mapsto |\mathbf{s}, \langle \mathbf{a}_1, \mathbf{s}\rangle + e_1, \dots, \langle \mathbf{a}_m, \mathbf{s}\rangle + e_m\rangle$ to get

$$\frac{1}{\sqrt{q^n}} \sum_{\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n} |\mathbf{s}\rangle \sum_{\mathbf{e} \in (\mathbb{Z}/q\mathbb{Z})^m} \bigotimes_{i=1}^{m} f(e_i) |\langle \mathbf{a}_i, \mathbf{s}\rangle + e_i\rangle = \frac{1}{\sqrt{q^n}} \sum_{\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n} |\mathbf{s}\rangle \bigotimes_{i=1}^{m} \left|\psi_{\langle \mathbf{a}_i, \mathbf{s}\rangle}\right\rangle \ .$$

5: Append $|0\rangle^m$.

6: Apply the unitary $\mathbf{I} \otimes \mathbf{V}^{\otimes m}$ with $\mathbf{V}$ as defined in Equation (2.12), to obtain

$$\frac{1}{\sqrt{q^n}} \sum_{\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n} |\mathbf{s}\rangle \bigotimes_{i=1}^{m} \mathbf{V}\left(\left|\psi_{\langle \mathbf{a}_i, \mathbf{s}\rangle}\right\rangle |0\rangle\right) \ .$$

7: Apply the quantum unambiguous Gaussian elimination as given in Equation (2.13) to get

$$\mathbf{U}_{\mathcal{A}_{\mathrm{GE}}} \left(\frac{1}{\sqrt{q^n}} \sum_{\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n} |\mathbf{s}\rangle \bigotimes_{i=1}^{m} \mathbf{V}\left(\left|\psi_{\langle \mathbf{a}_i, \mathbf{s}\rangle}\right\rangle |0\rangle\right) \right) \ .$$

8: Apply $\mathbf{I} \otimes \left(\mathbf{V}^\dagger\right)^{\otimes m}$ and output the resulting quantum state.

More details are required to make Steps 2, 6 and 7 explicit. For Step 2, we assume that we can efficiently implement an approximation of the state (see Condition 1 of Theorem 3). A realization for a specific amplitude function f will be discussed in Lemma 17. Step 6 relies on a unitary $\mathbf{V}$ satisfying:

$$\forall x \in \mathbb{Z}/q\mathbb{Z} : \mathbf{V}\left(|\chi_x\rangle |0\rangle\right) = |\chi_x\rangle \left(u_x |0\rangle + \sqrt{1 - |u_x|^2}\, |1\rangle\right) \ , \tag{2.12}$$

where u_x is an approximation of $(\min |\widehat{f}|)/\widehat{f}(-x)$ (see Condition 2 of Theorem 3). We will explain in Lemma 14 how to implement $\mathbf{V}$. Note that up to the numerical inaccuracy,

the unitary $\mathbf{V}$ can be viewed as an implementation of the unambiguous measurement from [CB98] (see Appendix 3.6). Step 7 uses a version of a Gaussian elimination algorithm $\mathcal{A}_{\mathrm{GE}}$ that works as follows when given as input a matrix $\mathbf{A} := (\mathbf{a}_1|\dots|\mathbf{a}_m)^\intercal \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$ and m equations $(y_i)_{1\leq i\leq m}$ where $y_i = \langle \mathbf{a}_i, \mathbf{s}\rangle$ or $y_i = \perp$ (some equations may be erased): it first tests whether the input matrix $\mathbf{A}$ is invertible modulo q (which is not required to be prime); if it is, it then outputs the unique solution $\mathbf{s}$; if it is not, it outputs $\perp$. Algorithm $\mathcal{A}_{\mathrm{GE}}$ is deterministic polynomial-time and has the following properties that will prove useful in our analysis of Algorithm 1:

- it is unambiguous, in the sense that it never outputs an incorrect solution, i.e., it either outputs the valid $\mathbf{s}$ or it fails and outputs $\perp$;
- if $\mathbf{A}$ is sampled uniformly, and the number of non-$\perp$ input y_i's is $(n+\log\log q)\omega(\log\lambda)$ and the indices of the non-$\perp$ input y_i's are chosen independently from $\mathbf{A}$, then $\mathcal{A}_{\mathrm{GE}}$ returns $\perp$ with probability $\mathsf{negl}(\lambda)$ (this can be obtained, e.g., by adapting [BLP+13, Claim 2.13]);
- for any fixed $\mathbf{A}$, if the indices of the non-$\perp$ y_i's are chosen randomly and independently from the rest, then the success probability is the same for every $(\mathbf{s}\in\mathbb{Z}/q\mathbb{Z})^n$.

In Algorithm 1, we consider a version of the Gaussian elimination $\mathcal{A}_{\mathrm{GE}}$ that is quantized as follows. For any $(\mathbf{s},\mathbf{x},\mathbf{b})\in(\mathbb{Z}/q\mathbb{Z})^n\times(\mathbb{Z}/q\mathbb{Z})^m\times\{0,1\}^m$,

$$\mathbf{U}_{\mathcal{A}_{\mathrm{GE}}} : |\mathbf{s}\rangle\bigotimes_{i=1}^m|x_i,b_i\rangle \longmapsto \Big|\mathbf{s}-\mathcal{A}_{\mathrm{GE}}(\mathbf{A},(y_i)_{1\leq i\leq m})\Big\rangle\bigotimes_{i=1}^m|x_i,b_i\rangle \ . \tag{2.13}$$

where $y_i = x_i$ if $b_i = 0$, and $y_i = \perp$ otherwise. To handle the potential output $\perp$ of $\mathcal{A}_{\mathrm{GE}}$, we embed the first quantum register in Equation (2.13) into $\mathbb{C}^{2q}$ where $(|x\rangle, |\perp\rangle_x)_{x\in\mathbb{Z}/q\mathbb{Z}}$ is the computational basis (for some arbitrary symbols $\perp_x$).

The following theorem gives conditions under which Algorithm 1 solves the $\mathsf{C}\,|\mathsf{LWE}\rangle$ problem in time $\mathsf{poly}(\lambda)$.

Theorem 3. *Let $m\geq n\geq 1$ and $q\geq 2$ be integers, and $f:\mathbb{Z}/q\mathbb{Z}\to\mathbb{C}$ be an amplitude function. The parameters m,n,q,f are functions of some security parameter λ with $m,\log q\leq\mathsf{poly}(\lambda)$. Assume that the following conditions hold:*

1. *there exists a $\mathsf{poly}(\lambda)$-time algorithm that builds a state within $\mathsf{negl}(\lambda)$ trace distance of the state $\sum_{e\in\mathbb{Z}/q\mathbb{Z}} f(e)\,|e\rangle$;*
2. *there exists a $\mathsf{poly}(\lambda)$-time algorithm that, given x as input, outputs*

$$u_x := (\min|\widehat{f}|)/\widehat{f}(-x) + e_{\mathrm{apx}}(x)$$

on $\mathsf{poly}(\lambda)$ bits with $\max_x|e_{\mathrm{apx}}(x)| = \mathsf{negl}(\lambda)/\sqrt{q^n}$;

3. *we have that $m = (n+\log\log q)/p\cdot\omega(\log\lambda)$, where $p := q\cdot\min|\widehat{f}|^2$;*
4. *assuming that $\mathbf{A}\in(\mathbb{Z}/q\mathbb{Z})^{m\times n}$ is uniformly distributed, we have*

$$\mathbb{P}_{\mathbf{A}}\left(\left|\frac{Z_f(\mathbf{A})}{q^n}-1\right|\geq\mathsf{negl}(\lambda)\right) = \mathsf{negl}(\lambda)\ ,$$

where $Z_f(\mathbf{A})$ is the normalization scalar such that $\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}$ becomes a unit vector, as per Definition 17.

Then Algorithm 1 runs in time $\mathsf{poly}(\lambda)$ and, for a proportion $1-\mathsf{negl}(\lambda)$ of matrices $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, it outputs a quantum state $|\varphi\rangle$ such that

$$D_{\mathrm{tr}}\left(|\varphi\rangle\,, |0\rangle^{n\log q}\,\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}\,|0\rangle^{m}\right) = \mathsf{negl}(\lambda)\ . \tag{2.14}$$

The first two conditions enable an efficient implementation of Algorithm 1. In Condition 3, the value p refers to the success probability in recovering j from $|\psi_j\rangle$. This condition ensures that m is sufficiently large for the unambiguous Gaussian elimination algorithm to succeed with probability $1-\mathsf{negl}(\lambda)$. Note that the condition on m and the fact that $m \leq \mathsf{poly}(\lambda)$ imply that we must have $p \geq 1/\mathsf{poly}(\lambda)$. The latter implies that $\widehat{f}(x)$ is non-zero for all $x \in \mathbb{Z}/q\mathbb{Z}$, a condition that is necessary to rely on the measurement from [CB98] and, more concretely, for the unitary $\mathbf{V}$ used in Step 6 of Algorithm 1 to be well-defined (see Condition 3). Still concerning Condition 3, the lower bound on m is to ensure that a uniform $m \times n$ matrix modulo q has an image of size $(\mathbb{Z}/q\mathbb{Z})^n$ with overwhelming probability. If q is prime, this condition can be simplified to $m = n/p \cdot \omega(\log\lambda)$. Finally, Condition 4 intuitively states that the parametrization of LWE provides a unique solution with overwhelming probability. The last two conditions can be simplified if q is assumed to be prime.

We will first consider the correctness of Algorithm 1 (Lemma 13), and then analyze its runtime (Lemma 15).

2.4.2 Correctness

The purpose of the unitary $\mathbf{V}$ (introduced in Equation (2.12)) is to recover the quantity of $\langle \mathbf{a}_i, \mathbf{s}\rangle$ from $|\psi_{\langle \mathbf{a}_i,\mathbf{s}\rangle}\rangle$. More formally, we have the following lemma.

Lemma 9. *Using notations of Theorem 3 and with $\mathbf{V}$ as defined in Equation (2.12), we have*

$$\forall j \in \mathbb{Z}/q\mathbb{Z},\ \mathbf{V}\left(|\psi_j\rangle\,|0\rangle\right) = \sqrt{p}\,|j\rangle\,|0\rangle + \sqrt{1-p}\,|\eta_j\rangle\,|1\rangle + |\mathrm{error}_j\rangle\ ,$$

for some quantum states $|\eta_j\rangle$ and $|\mathrm{error}_j\rangle$ with $\max_j \| \,|\mathrm{error}_j\rangle\,\| = \mathsf{negl}(\lambda)/\sqrt{q^n}$.

Proof. Let us write the $|\psi_j\rangle$'s (Definition 19) in the Fourier basis $(|\chi_x\rangle)_{x\in\mathbb{Z}/q\mathbb{Z}}$. We have, for all $j \in \mathbb{Z}/q\mathbb{Z}$:

$$\begin{aligned}
|\psi_j\rangle &= \sum_{e\in\mathbb{Z}/q\mathbb{Z}} f(e)\,|j+e \bmod q\rangle \\
&= \frac{1}{\sqrt{q}} \sum_{e\in\mathbb{Z}/q\mathbb{Z}} f(e) \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{-(j+e)x}\,|\chi_x\rangle \qquad \text{(by Lemma 1)} \\
&= \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \left(\frac{1}{\sqrt{q}} \sum_{e\in\mathbb{Z}/q\mathbb{Z}} f(e)\,\omega_q^{-xe}\right) \omega_q^{-jx}\,|\chi_x\rangle \\
&= \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \widehat{f}(-x)\,\omega_q^{-jx}\,|\chi_x\rangle\ .
\end{aligned}$$

Therefore, by linearity and definition of $\mathbf{V}$, we have:

$$\begin{aligned}
&\mathbf{V}\left(|\psi_j\rangle\,|0\rangle\right)\\
&= \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \widehat{f}(-x)\omega_q^{-jx}\ \mathbf{V}\left(|\chi_x\rangle\,|0\rangle\right)\\
&= \underbrace{\left(\sum_{x\in\mathbb{Z}/q\mathbb{Z}} u_x\ \widehat{f}(-x)\ \omega_q^{-jx}\,|\chi_x\rangle\right)}_{:=|\psi_{j,0}\rangle}|0\rangle + \underbrace{\left(\sum_{x\in\mathbb{Z}/q\mathbb{Z}} \sqrt{1-|u_x|^2}\ \widehat{f}(-x)\ \omega_q^{-jx}\,|\chi_x\rangle\right)}_{:=|\psi_{j,1}\rangle}|1\rangle\ .
\end{aligned}$$

Let us consider $|\psi_{j,0}\rangle$. By definition of u_x and p, we have:

$$|\psi_{j,0}\rangle = \underbrace{\sqrt{q}\cdot\min|\widehat{f}|\ \left(\frac{1}{\sqrt{q}}\sum_{x\in\mathbb{Z}/q\mathbb{Z}}\omega_q^{-jx}\,|\chi_x\rangle\right)}_{=\sqrt{p}|j\rangle} + \underbrace{\sum_{x\in\mathbb{Z}/q\mathbb{Z}} e_{\mathrm{apx}}(x)\widehat{f}(-x)\omega_q^{-jx}\,|\chi_x\rangle}_{:=|\mathrm{error}_{j,0}\rangle}\ .$$

Notice that:

$$\|\,|\mathrm{error}\rangle_{j,0}\,\|^2 = \sum_{x\in\mathbb{Z}/q\mathbb{Z}} e_{\mathrm{apx}}(x)^2\ |\widehat{f}(-x)|^2 \leq \left(\max_{x\in\mathbb{Z}/q\mathbb{Z}}|e_{\mathrm{apx}}(x)|\right)^2\ .$$

Hence, so far, we have:

$$\mathbf{V}\left(|\psi_j\rangle\,|0\rangle\right) = \sqrt{p}\,|j\rangle\,|0\rangle + |\mathrm{error}_{j,0}\rangle\,|0\rangle + |\psi_{j,1}\rangle\,|1\rangle\ , \tag{2.15}$$

where $\|\,|\mathrm{error}_{j,0}\rangle\,\| = \mathsf{negl}(\lambda)/\sqrt{q^n}$, by assumption on $\max_x|e_{\mathrm{apx}}(x)|$. Notice that $\mathbf{V}(|\psi_j\rangle\,|0\rangle)$ is a quantum state as $\mathbf{V}$ is unitary. Therefore, we can write

$$|\psi_{j,1}\rangle = \sqrt{1-p}\,|\eta_j\rangle + |\mathrm{error}_{j,1}\rangle\ ,$$

for some quantum states $|\eta_j\rangle$ and $|\mathrm{error}_{j,1}\rangle$ such that $\|\,|\mathrm{error}_{j,1}\rangle\,\| = \mathsf{negl}(\lambda)/\sqrt{q^n}$. Plugging this into Equation (2.15) gives the result. □

As can be seen from Lemma 9, the transformation $\mathbf{V}$ introduces an error term. It basically comes from the fact that we only assume that we can approximate $(\min|\widehat{f}|)/\widehat{f}(-x)$ (as opposed to exactly computing it). This seems necessary for our subsequent choice of f. Ideally, we would analyze the correctness of Algorithm 1 as if we were applying a unitary $\mathbf{W}$ (that we do not know how to implement efficiently) such that

$$\forall j\in\mathbb{Z}/q\mathbb{Z},\quad \mathbf{W}\left(|\psi_j\rangle\,|0\rangle\right) = \sqrt{p}\,|j\rangle\,|0\rangle + \sqrt{1-p}\,|\eta_j\rangle\,|1\rangle\ ,$$

The value $\langle\mathbf{a}_i,\mathbf{s}\rangle$ appears (in superposition) in the first register of $\mathbf{W}(\left|\psi_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle|0\rangle)$, for all $i\leq m$. Therefore, applying the unitary $\mathbf{U}_{\mathcal{A}_{\mathrm{GE}}}$ as in Step 7 to the quantum state

$$\frac{1}{\sqrt{q^n}}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}|\mathbf{s}\rangle\bigotimes_{i=1}^{m}\mathbf{W}\left(\left|\psi_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle|0\rangle\right)$$

will allow us to erase $\mathbf{s}$ from the first register. More precisely, we hope that after Step 7, the quantum state

$$\mathbf{U}_{\mathcal{A}_{\mathrm{GE}}}\left(\frac{1}{\sqrt{q^n}}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}|\mathbf{s}\rangle\bigotimes_{i=1}^{m}\mathbf{W}\left|\psi_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle|0\rangle\right)$$

will be "close" to the disentangled state

$$\frac{1}{\sqrt{Z_f(\mathbf{A})}}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}|\mathbf{0}\rangle\bigotimes_{i=1}^{m}\mathbf{W}\left|\psi_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle|0\rangle\,.$$

Notice now that applying $\mathbf{I}\otimes\left(\mathbf{V}^\dagger\right)^{\otimes m}$, as in Step 8, to the state above does not yield the quantum state $|\mathbf{0}\rangle\,\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}\,|0\rangle$. Instead, this would hold if we were rather applying $\mathbf{I}\otimes\left(\mathbf{W}^\dagger\right)^{\otimes m}$. But we do not know how to implement $\mathbf{W}$ efficiently. In the following two lemmas we show that applying $\mathbf{V}$ and $\mathbf{V}^\dagger$ lead to a quantum state that is close with respect to the trace distance to the case where we would instead apply $\mathbf{W}$ and $\mathbf{W}^\dagger$.

Lemma 10. *Using notations of Theorem 3 and letting*

$$|\varphi'\rangle := \left(\mathbf{I}\otimes\left(\mathbf{V}^\dagger\right)^{\otimes m}\right)\mathbf{U}_{\mathcal{A}_{\mathrm{GE}}}\left(\frac{1}{\sqrt{q^n}}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}|\mathbf{s}\rangle\bigotimes_{i=1}^{m}\left(\sqrt{p}\,|\langle\mathbf{a}_i,\mathbf{s}\rangle\rangle\,|0\rangle+\sqrt{1-p}\left|\eta_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle|1\rangle\right)\right)\,, \tag{2.16}$$

we have

$$D_{\mathrm{tr}}\left(|\varphi\rangle\,,|\varphi'\rangle\right)=\frac{\mathsf{negl}(\lambda)}{q^{n/4}}\,.$$

Proof. Recall that $|\varphi\rangle$ is obtained at the end of Step 8 of Algorithm 1. In particular, thanks to Lemma 9, we have

$$\begin{aligned}|\varphi\rangle &= \left(\mathbf{I}\otimes\left(\mathbf{V}^\dagger\right)^{\otimes m}\right)\mathbf{U}_{\mathcal{A}_{\mathrm{GE}}}\left(\frac{1}{\sqrt{q^n}}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}|\mathbf{s}\rangle\bigotimes_{i=1}^{m}\mathbf{V}\left(\left|\psi_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle|0\rangle\right)\right)\\ &= \left(\mathbf{I}\otimes\left(\mathbf{V}^\dagger\right)^{\otimes m}\right)\mathbf{U}_{\mathcal{A}_{\mathrm{GE}}}\left(\frac{1}{\sqrt{q^n}}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}|\mathbf{s}\rangle\bigotimes_{i=1}^{m}\left(\sqrt{p}\,|\langle\mathbf{a}_i,\mathbf{s}\rangle\rangle\,|0\rangle+\sqrt{1-p}\left|\eta_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle|1\rangle\right.\right.\\ &\qquad\qquad\left.\left.+\left|\mathrm{error}_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle\right)\right)\,.\end{aligned}$$

Taking the Hermitian product, we obtain:

$$\langle\varphi'|\varphi\rangle=\frac{1}{q^n}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}\prod_{i=1}^{m}\left(1+\sqrt{p}\left\langle\langle\mathbf{a}_i,\mathbf{s}\rangle,0\middle|\mathrm{error}_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle+\sqrt{1-p}\left\langle\eta_{\langle\mathbf{a}_i,\mathbf{s}\rangle},1\middle|\mathrm{error}_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle\right)\,.$$

As $\max_j\||\mathrm{error}_j\rangle\|=\mathsf{negl}(\lambda)/\sqrt{q^n}$, we have that

$$\langle\varphi'|\varphi\rangle=q^{-n}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}\prod_{i=1}^{m}(1+z_{\mathbf{s},i})\,,$$

for some $z_{\mathbf{s},i}\in\mathbb{C}$ satisfying $\max_{\mathbf{s},i}|z_{\mathbf{s},i}|\leq\mathsf{negl}(\lambda)/\sqrt{q^n}$. Using the fact that $m\leq\mathsf{poly}(\lambda)$, we obtain that $\langle\varphi'|\varphi\rangle=1-\mathsf{negl}(\lambda)/\sqrt{q^n}$. □

Lemma 11. *Using notations of Theorem 3 and letting*

$$|\psi'\rangle := \left(\mathbf{I}\otimes\left(\mathbf{V}^\dagger\right)^{\otimes m}\right)\left(\frac{1}{\sqrt{Z_f(\mathbf{A})}}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}|\mathbf{0}\rangle\bigotimes_{i=1}^{m}\left(\sqrt{p}\,|\langle\mathbf{a}_i,\mathbf{s}\rangle\rangle\,|0\rangle+\sqrt{1-p}\left|\eta_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle|1\rangle\right)\right) \tag{2.17}$$

we have

$$D_{\mathrm{tr}}\left(|\psi'\rangle\,,|\mathbf{0}\rangle\,\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}\,|0\rangle\right)\leq\sqrt{1-\left(1-\sqrt{\frac{q^n}{Z_f(\mathbf{A})}}\,\mathsf{negl}(\lambda)\right)^2}\,.$$

Proof. By Definition 17, we have

$$\begin{aligned}
|\mathbf{0}\rangle\,\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}\,|0\rangle &= \frac{1}{\sqrt{Z_f(\mathbf{A})}}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}|\mathbf{0}\rangle\bigotimes_{i=1}^{m}\left|\psi_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle|0\rangle\\
&=\left(\mathbf{I}\otimes\left(\mathbf{V}^\dagger\right)^{\otimes m}\right)\left(\frac{1}{\sqrt{Z_f(\mathbf{A})}}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}|\mathbf{0}\rangle\bigotimes_{i=1}^{m}\mathbf{V}\left(\left|\psi_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle|0\rangle\right)\right)\\
&=\left(\mathbf{I}\otimes\left(\mathbf{V}^\dagger\right)^{\otimes m}\right)\left(\frac{1}{\sqrt{Z_f(\mathbf{A})}}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}|\mathbf{0}\rangle\bigotimes_{i=1}^{m}\left(\sqrt{p}\,|\langle\mathbf{a}_i,\mathbf{s}\rangle\rangle\,|0\rangle\right.\right.\\
&\qquad\left.\left.+\sqrt{1-p}\left|\eta_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle|1\rangle+\left|\mathrm{error}_{\langle\mathbf{a}_i,\mathbf{s}\rangle}\right\rangle\right)\right)\,.
\end{aligned}$$

Recall that $\max_j\|\,|\mathrm{error}_j\rangle\,\|=\mathsf{negl}(\lambda)/\sqrt{q^n}$ (see Lemma 9). Therefore, we have

$$|\mathbf{0}\rangle\,\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}\,|0\rangle^m=|\psi'\rangle+\left(\mathbf{I}\otimes\left(\mathbf{V}^\dagger\right)^{\otimes m}\right)\left(\frac{1}{\sqrt{Z_f(\mathbf{A})}}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}|\mathbf{0}\rangle\,|\mathrm{error}_{\mathbf{s}}\rangle\right)\,,$$

for some $\mathrm{error}_{\mathbf{s}}$ satisfying $\max_{\mathbf{s}}\|\,|\mathrm{error}_{\mathbf{s}}\rangle\,\|\leq m\;\mathsf{negl}(\lambda)/\sqrt{q^n}\leq\mathsf{negl}(\lambda)/\sqrt{q^n}$, since $m\leq\mathsf{poly}(\lambda)$. We hence obtain that

$$\left\|\left(\mathbf{I}\otimes\left(\mathbf{V}^\dagger\right)^{\otimes m}\right)\left(\frac{1}{\sqrt{Z_f(\mathbf{A})}}\sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n}|\mathbf{0}\rangle\,|\mathrm{error}_{\mathbf{s}}\rangle\right)\right\|\leq\frac{q^n}{\sqrt{Z_f(\mathbf{A})}}\,\frac{\mathsf{negl}(\lambda)}{\sqrt{q^n}}=\sqrt{\frac{q^n}{Z_f(\mathbf{A})}}\mathsf{negl}(\lambda)\,,$$

which completes the proof. □

The following lemma will help us in analyzing the effect of the unitary $\mathbf{U}_{\mathcal{A}_{\mathrm{GE}}}$. It considers its application on a state whose second and third registers contain a superposition of solved and undetermined linear equations. It is obtained from (2.13) by linearity and the fact that y_i in Equation (2.13) depends only in the last qubit.

Lemma 12. *Let $\mathbf{U}_{\mathcal{A}_{\mathrm{GE}}}$ be defined as in Equation (2.13). Let $x_1,\ldots,x_m\in\mathbb{Z}/q\mathbb{Z}$ and $|\eta_1\rangle,\ldots,|\eta_m\rangle$ be some quantum states. We have*

$$\mathbf{U}_{\mathcal{A}_{\mathrm{GE}}}\left(|\mathbf{s}\rangle\bigotimes_{i=1}^{m}\left(\sqrt{p}\,|x_i\rangle\,|0\rangle+\sqrt{1-p}\,|\eta_i\rangle\,|1\rangle\right)\right)=\sum_{\mathbf{y}\in\{x_i,\perp\}^m}|\mathbf{s}-\mathcal{A}_{\mathrm{GE}}(\mathbf{A},\mathbf{y})\rangle\bigotimes_{i=1}^{m}\lambda(y_i)\left|\alpha_{y_i}^{\mathbf{s}}\right\rangle\,,$$

where

$$\left|\alpha_{y_i}^{\mathbf{s}}\right\rangle:=\begin{cases}|x_i\rangle\,|0\rangle & \text{if } y_i=x_i\\ |\eta_i\rangle\,|1\rangle & \text{otherwise}\end{cases}\qquad\text{and}\qquad\lambda(y_i):=\begin{cases}\sqrt{p} & \text{if } y_i=x_i\\ \sqrt{1-p} & \text{otherwise}\end{cases}\,.$$

We can now show the correctness of Algorithm 1, i.e., that Equation (2.14) holds.

Lemma 13. *Using the notations of Theorem 3, we have, for a proportion* $1 - \mathsf{negl}(\lambda)$ *of matrices* $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$*:*

$$D_{\mathrm{tr}}\left(|\varphi\rangle, |0\rangle^{n \log q} \mathsf{C}\, |\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f} |0\rangle^{m}\right) = \mathsf{negl}(\lambda) \ .$$

Proof. First, by Condition 1 of Theorem 3, we can build the quantum state $\sum_{e \in \mathbb{Z}/q\mathbb{Z}} f(e) |e\rangle$ up to a trace distance $\mathsf{negl}(\lambda)$. Therefore, when analyzing the trace distance between the output $|\varphi\rangle$ of Algorithm 1 and $|\mathbf{0}\rangle \mathsf{C}\, |\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f} |\mathbf{0}\rangle$, we can assume that it is exactly $\sum_{\mathbf{e} \in (\mathbb{Z}/q\mathbb{Z})^m} \bigotimes_{j=1}^{m} f(e) |e_i\rangle$ that is built at Step 2. Indeed, this only affects the trace distance by an additive $m\mathsf{negl}(\lambda) = \mathsf{negl}(\lambda)$ term (recall that we have $m \leq \mathsf{poly}(\lambda)$).

By Lemmas 10 and 11, and the triangular inequality over the trace distance, we have

$$\begin{aligned}
&D_{\mathrm{tr}}\left(|\varphi\rangle, |0\rangle^{n \log q} \mathsf{C}\, |\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f} |0\rangle^{m}\right) \\
&\qquad \leq D_{\mathrm{tr}}\left(|\varphi\rangle, |\varphi'\rangle\right) + D_{\mathrm{tr}}\left(|\varphi'\rangle, |\psi'\rangle\right) + D_{\mathrm{tr}}\left(|\psi'\rangle, |0\rangle^{n \log q} \mathsf{C}\, |\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f} |0\rangle^{m}\right) \\
&\qquad \leq D_{\mathrm{tr}}\left(|\varphi'\rangle, |\psi'\rangle\right) + \frac{\mathsf{negl}(\lambda)}{q^{n/4}} + \sqrt{1 - \left(1 - \frac{q^n}{Z_f(\mathbf{A})} \mathsf{negl}(\lambda)\right)^2} \ , \qquad (2.18)
\end{aligned}$$

where $|\varphi'\rangle$ and $|\psi'\rangle$ are respectively defined in Equations (2.16) and (2.17). Applying the unitary $\mathbf{I} \otimes \left(\mathbf{V}^{\dagger}\right)^{\otimes m}$ does not change the trace distance. Therefore, by using the definitions of $|\varphi'\rangle$ and $|\psi'\rangle$, we have

$$D_{\mathrm{tr}}\left(|\varphi'\rangle, |\psi'\rangle\right) = D_{\mathrm{tr}}\left(|\psi\rangle, |\psi_{\mathrm{ideal}}\rangle\right) \qquad (2.19)$$

where

$$|\psi\rangle := \mathbf{U}_{\mathcal{A}_{\mathrm{GE}}} \left(\frac{1}{\sqrt{q^n}} \sum_{\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n} |\mathbf{s}\rangle \bigotimes_{i=1}^{m} \left(\sqrt{p}\, |\langle \mathbf{a}_i, \mathbf{s} \rangle\rangle\, |0\rangle + \sqrt{1-p} \left| \eta_{\langle \mathbf{a}_i, \mathbf{s} \rangle} \right\rangle |1\rangle \right) \right)$$

and

$$|\psi_{\mathrm{ideal}}\rangle := \frac{1}{\sqrt{Z_f(\mathbf{A})}} \sum_{\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n} |\mathbf{0}\rangle \bigotimes_{i=1}^{m} \left(\sqrt{p}\, |\langle \mathbf{a}_i, \mathbf{s} \rangle\rangle\, |0\rangle + \sqrt{1-p} \left| \eta_{\langle \mathbf{a}_i, \mathbf{s} \rangle} \right\rangle |1\rangle \right) \ .$$

By Lemma 12, we have

$$|\psi\rangle = \frac{1}{\sqrt{q^n}} \sum_{\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n} \sum_{\mathbf{y} \in \{\langle \mathbf{a}_i, \mathbf{s} \rangle, \perp\}^m} |\mathbf{s} - \mathcal{A}_{\mathrm{GE}}(\mathbf{A}, \mathbf{y})\rangle \bigotimes_{i=1}^{m} \lambda(y_i) \left| \alpha^{\mathbf{s}}_{y_i} \right\rangle$$

where,

$$\left| \alpha^{\mathbf{s}}_{y_i} \right\rangle := \begin{cases} |\langle \mathbf{a}_i, \mathbf{s} \rangle\rangle\, |0\rangle & \text{if } y_i = \langle \mathbf{a}_i, \mathbf{s} \rangle \\ \left| \eta_{\langle \mathbf{a}_i, \mathbf{s} \rangle} \right\rangle |1\rangle & \text{otherwise} \end{cases} \quad \text{and} \quad \lambda(y_i) := \begin{cases} \sqrt{p} & \text{if } y_i = \langle \mathbf{a}_i, \mathbf{s} \rangle \\ \sqrt{1-p} & \text{otherwise} \end{cases} \ . \qquad (2.20)$$

Similarly, we have

$$|\psi_{\text{ideal}}\rangle = \frac{1}{\sqrt{Z_f(\mathbf{A})}} \sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n} \sum_{\mathbf{y}\in\{\langle\mathbf{a}_i,\mathbf{s}\rangle,\perp\}^m} |\mathbf{0}\rangle \bigotimes_{i=1}^m \lambda(y_i) \left|\alpha_{y_i}^{\mathbf{s}}\right\rangle \ .$$

We deduce that

$$\langle\psi_{\text{ideal}}|\psi\rangle = \frac{1}{\sqrt{q^n\ Z_f(\mathbf{A})}} \sum_{\mathbf{s},\mathbf{s}'\in(\mathbb{Z}/q\mathbb{Z})^n} \sum_{\mathbf{y}\in\{\langle\mathbf{a}_i,\mathbf{s}\rangle,\perp\}^m} \sum_{\mathbf{y}'\in\{\langle\mathbf{a}_i,\mathbf{s}'\rangle,\perp\}^m} \underbrace{\langle\mathbf{0}|\mathbf{s}'-\mathcal{A}_{\text{GE}}(\mathbf{A},\mathbf{y}')\rangle \prod_{i=1}^m \lambda(y_i)\lambda(y_i') \left\langle\alpha_{y_i}^{\mathbf{s}}\middle|\alpha_{y_i'}^{\mathbf{s}'}\right\rangle}_{:=P_{\mathbf{s},\mathbf{s}',\mathbf{y},\mathbf{y}'}} \ .$$

Our aim is to show that $P_{\mathbf{s},\mathbf{s}',\mathbf{y},\mathbf{y}'}$ is always equal to 0 except when $\mathbf{s}=\mathbf{s}'$ and $\mathbf{y}=\mathbf{y}'$. First, notice that $P_{\mathbf{s},\mathbf{s}',\mathbf{y},\mathbf{y}'}$ can be non-zero only if the following holds

$$\mathbf{s}' = \mathcal{A}_{\text{GE}}(\mathbf{A},\mathbf{y}') \ .$$

At this stage, recall that our Gaussian elimination algorithm $\mathcal{A}_{\text{GE}}$ is unambiguous: with the knowledge of $\mathbf{y}'$, it can only output $\mathbf{s}'$ of $\perp$ (but not output another vector). Further, to have $P_{\mathbf{s},\mathbf{s}',\mathbf{y},\mathbf{y}'}\neq 0$, we also need

$$\forall i : \left\langle\alpha_{y_i}^{\mathbf{s}}\middle|\alpha_{y_i'}^{\mathbf{s}'}\right\rangle \neq 0 \ .$$

Therefore, by definition of the $\left|\alpha_{y_i}^{\mathbf{s}}\right\rangle$'s in Equation (2.20), it is necessary that for all i, we have $y_i = y_i'$. However, the y_i''s uniquely determine $\mathbf{s}'$, therefore $\mathbf{s}=\mathbf{s}'$ in that case. Overall, we obtain that $P_{\mathbf{s},\mathbf{s}',\mathbf{y},\mathbf{y}'}\neq 0$ implies that $\mathbf{s}=\mathbf{s}'$ and $\mathbf{y}=\mathbf{y}'$. Therefore, we obtain

$$\begin{aligned}\langle\psi_{\text{ideal}}|\psi\rangle &= \frac{1}{\sqrt{Z_f(\mathbf{A})\ q^n}} \sum_{\mathbf{s}\in(\mathbb{Z}/q\mathbb{Z})^n} \sum_{\substack{\mathbf{y}\in(\{\langle\mathbf{a}_j,\mathbf{s}\rangle,\ \perp\})_{j=1}^m:\\ \mathbf{s}=\mathcal{A}_{\text{GE}}(\mathbf{A},\mathbf{y})}} \prod_{i=1}^m \lambda(y_i)^2 \\ &= \sqrt{\frac{q^n}{Z_f(\mathbf{A})}}\ p_{\mathcal{A}_{\text{GE}}}(\mathbf{A}) \ , \end{aligned} \tag{2.21}$$

where $p_{\mathcal{A}_{\text{GE}}}(\mathbf{A})$ is the success probability of $\mathcal{A}_{\text{GE}}$ when each of its m equations as input is $\perp$ with probability $1-p$ and $\langle\mathbf{a}_i,\mathbf{s}\rangle$, with probability p (recall that $p_{\mathcal{A}_{\text{GE}}}(\mathbf{A})$ is independent from $\mathbf{s}$). Now, by Condition 3 of Theorem 3, we have $m=(n+\log\log q)/p\cdot\omega(\log\lambda)$. Therefore, by assumption on algorithm $\mathcal{A}_{\text{GE}}$, except for a $\mathsf{negl}(\lambda)$-proportion of matrices $\mathbf{A}$, we have

$$p_{\mathcal{A}_{\text{GE}}}(\mathbf{A}) = 1-\mathsf{negl}(\lambda) \ .$$

By using Equations (2.18), (2.19) and (2.21), we deduce that for a proportion $1-\mathsf{negl}(\lambda)$ of matrices $\mathbf{A}$, we have

$$\begin{aligned} D_{\text{tr}}\left(|\varphi\rangle\,,|0\rangle^{n\log q}\,\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}\,|0\rangle^m\right) &\leq \sqrt{1-\frac{q^n}{Z_f(\mathbf{A})}(1-\mathsf{negl}(\lambda))^2} \\ &\quad + \frac{\mathsf{negl}(\lambda)}{q^{n/4}} + \sqrt{1-\left(1-\sqrt{\frac{q^n}{Z_f(\mathbf{A})}}\ \mathsf{negl}(\lambda)\right)^2} \ . \end{aligned}$$

To complete the proof, it suffices to use Condition 4 of Theorem 3. □

2.4.3 Runtime

We now focus on the runtime of Algorithm 1. So far, we did not specify how to compute the unitary $\mathbf{V}$. This is the focus of the following lemma.

Lemma 14. *Using notations of Theorem 3, we can evaluate a unitary* $\mathbf{V}$ *satisfying Equation* (2.12) *in time* $\mathsf{poly}(\lambda)$.

Proof. Our objective is to implement $\mathbf{V}$ such that

$$\forall x \in \mathbb{Z}/q\mathbb{Z} : \mathbf{V}\left(|\chi_x\rangle\,|0\rangle\right) = |\chi_x\rangle\left(u_x\,|0\rangle + \sqrt{1-|u_x|^2}\,|1\rangle\right) .$$

By Condition 2 of Theorem 3, we can efficiently compute $u_x = (\min|\widehat{f}|)/\widehat{f}(-x)+e_{\mathrm{apx}}(x) \in \mathbb{C}$ on $\mathsf{poly}(\lambda)$ bits with $\max_x |e_{\mathrm{apx}}(x)| = \mathsf{negl}(\lambda)/\sqrt{q^n}$. Without loss of generality, we assume that u_x is written as its magnitude and phase (m_x, θ_x) where m_x and θ_x have $b = \mathsf{poly}(\lambda)$ bits. As $x \mapsto u_x$ is computable in time $\mathsf{poly}(\lambda)$, we can evaluate a unitary $\mathbf{O}_u$ satisfying the following, in quantum-time $\mathsf{poly}(\lambda)$:

$$\forall x : \mathbf{O}_u\left(|x\rangle\left|0^{2b}\right\rangle\right) = |x\rangle\,|m_x\rangle\,|\theta_x\rangle .$$

Now, consider the following two unitaries:

$$\mathbf{M} := \sum_{y\in\{0,1\}^b} |y\rangle\langle y| \otimes \mathbf{I}_p \otimes \left(\widetilde{y}\,|0\rangle + \sqrt{1-\widetilde{y}^2}\,|1\rangle\right)\langle 0| ,$$
$$\mathbf{\Theta} := \sum_{z\in\{0,1\}^b} \mathbf{I}_b \otimes |z\rangle\langle z| \otimes \left(\mathrm{e}^{2\pi i\widetilde{z}}\,|0\rangle\langle 0| + |1\rangle\langle 1|\right) ,$$

where $\widetilde{y} = \sum_{i=1}^b y_i/2^i$ and $\widetilde{z} = \sum_{i=1}^b z_i/2^i$. It can be checked that

$$\mathbf{O}_u^\dagger \mathbf{\Theta} \mathbf{M} \mathbf{O}_u\left(|x\rangle\left|0^{2b}\right\rangle|0\rangle\right) = |x\rangle\left|0^{2b}\right\rangle\left(u_x\,|0\rangle + \sqrt{1-|u_x|^2}\,|1\rangle\right) .$$

The unitary $\mathbf{M}$ can be implemented with $O(b) = \mathsf{poly}(\lambda)$ unary and binary gates [dW23, Ch. 9, Exercise 7.a]. Furthermore, we have

$$\mathrm{e}^{2\pi i\widetilde{z}} = \prod_{k=1}^{b} \mathrm{e}^{2\pi i 2^{-k} z_k} .$$

It shows that one only requires $b = \mathsf{poly}(\lambda)$ controlled gates to implement $\mathbf{\Theta}$. This completes the proof. □

We are now ready to prove that we can run Algorithm 1 in polynomial time.

Lemma 15. *Using notations of Theorem 3, Algorithm 1 can be executed in time* $\mathsf{poly}(\lambda)$.

Proof. Step 2 of Algorithm 1 can be executed in time $\mathsf{poly}(\lambda)$ by Condition 1 of Theorem 3. All steps except Steps 6, 7 and 8 are readily seen to be computable in time $\mathsf{poly}(\lambda)$ as $m, \log q \leq \mathsf{poly}(\lambda)$. By Lemma 14, Steps 6 and 8 can be executed in time $m\,\mathsf{poly}(\lambda) = \mathsf{poly}(\lambda)$. Finally, Step 7 applies $\mathbf{U}_{\mathrm{GE}}$. This unitary quantizes a $\mathsf{poly}(\lambda)$-time Gaussian elimination algorithm. □

2.5 $\mathsf{C}\,|\mathsf{LWE}\rangle$ for the Gaussian distribution and witness-oblivious LWE sampling

Our aim in this section is to construct a witness-oblivious quantum $\mathsf{LWE}_{m,n,q,|f|^2}$ sampler. For this purpose, we use Algorithm 1 with a specific choice of parameter f, to obtain the following theorem. The second part of the statement below is obtained by combining the first part and Theorem 2. This proves Theorem 1.

Theorem 4. *Let $m \geq n \geq 1$ and $q \geq 3$ be integers and $\sigma \geq 2$ be a real number. The parameters m, n, q, σ are functions of the security parameter λ with $m, \log q \leq \mathsf{poly}(\lambda)$ and q prime. Assume that the parameters satisfy the following conditions:*

$$m \geq n\sigma \cdot \omega(\log \lambda) \quad \text{and} \quad 2 \leq \sigma \leq \frac{q}{\sqrt{8m \ln q}} \,.$$

Furthermore, let $f : \mathbb{Z}/q\mathbb{Z} \to \mathbb{C}$ be such that

$$f(x) := \begin{cases} \sqrt{\vartheta_{\sigma,q}(x)} & \text{if } 0 \leq x \leq \frac{q}{2} \\ -\sqrt{\vartheta_{\sigma,q}(x)} & \text{otherwise} \end{cases} \,.$$

Then Algorithm 1 runs in time $\mathsf{poly}(\lambda)$ and, for a proportion $1 - \mathsf{negl}(\lambda)$ of matrices $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, it outputs a quantum state $|\varphi\rangle$ such that $D_{\mathrm{tr}}(|\varphi\rangle, |\mathbf{0}\rangle\,\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}\,|0\rangle) = \mathsf{negl}(\lambda)$.

In particular, if $\mathsf{LWE}_{m,n,q,\sigma}$ is quantumly hard, then there exists a $\mathsf{poly}(\lambda)$-time quantum witness-oblivious $\mathsf{LWE}_{m,n,q,\sigma}$ sampler.

Note that Theorem 1 puts some constraints on the arithmetic shape of the modulus q, on the number of samples m, and on the standard deviation parameter σ. It would be convenient to allow smaller values of m, arbitrary arithmetic shapes for q and super-polynomial values of σ. Indeed, these are frequent parametrizations of LWE. To reach such values, we can use randomized Karp reductions from LWE for some parameters to LWE for other parameters. For instance, we can use Theorem 4 with many samples, and just throw away the superfluous ones. We may also use Theorem 4 with some permitted parameters n, σ, q for which LWE is hard, and then perform modulus-switching or modulus-dimension switching [BLP+13]. As an example, using modulus switching and throwing away superfluous samples, we obtain the following corollary.

Corollary 1. *Let $m \geq n \geq 1$ and $q \geq 2$ be integers and $\sigma \geq 2$ be a real number. The parameters m, n, q, σ are functions of the security parameter λ with $m, \sigma, \log q \leq \mathsf{poly}(\lambda)$. Assume that $\mathsf{LWE}_{m',n,q',\sigma'}$ is hard, where $q' \leq 2q$ is the smallest prime larger than q, $\sigma' = \sigma/(n+\lambda) \cdot \Omega_\lambda(1)$ and $m' = \max(m, n\sigma' \cdot \omega(\log \lambda))$. If $2 \leq \sigma' \leq q'/\sqrt{8m' \ln q'}$, then there exists a witness-oblivious $\mathsf{LWE}_{m,n,q,\sigma}$ sampler.*

To prove Theorem 4, we show that the conditions of Theorem 3 are fulfilled for the amplitude function of Theorem 4. This is the purpose of the rest of this section.

2.5.1 On Conditions 1 and 2 of Theorem 3

In the lemmas below, we show that f and $\widehat{f}$ can be approximated with sufficient precision for Conditions 1 and 2 to apply.

Lemma 16. *Let $n \geq 1$, $q \geq 3$ integers, $\sigma > 0$ a real number and $f : \mathbb{Z}/q\mathbb{Z} \to \mathbb{C}$ as in Theorem 4. Assume that $n, \sigma = \mathsf{poly}(\lambda)$ and $q = 2^{\mathsf{poly}(\lambda)}$ is odd, where λ is security parameter. Then we can compute $u_x = (\min |\widehat{f}|)/\widehat{f}(-x) + e_{\mathrm{apx}}(x)$ on $\mathsf{poly}(\lambda)$ bits with $\max_x |e_{\mathrm{apx}}(x)| = \mathsf{negl}(\lambda)/\sqrt{q^n}$, in classical time $\mathsf{poly}(\lambda)$.*

Proof. We show how to approximate $\widehat{f}(x)$ for every x within appropriate accuracy. This also suffices to approximate $\min |\widehat{f}|$ because, by Lemma 19, we have $\min |\widehat{f}| = |\widehat{f}(0)|$. As seen in the proof of Lemma 19, we have, for all $y \in \mathbb{Z}/q\mathbb{Z}$:

$$\widehat{f}(y) = \frac{f(0)}{\sqrt{q}} + i\,\frac{2}{\sqrt{q}} \sum_{x \in \mathbb{Z} \cap (0, q/2)} f(x)\ \sin \frac{2\pi x y}{q}\ .$$

First, note that one can efficiently approximate $f(x)$ on $\mathsf{poly}(\lambda)$ bits and within an absolute error $\mathsf{negl}(\lambda)/\sqrt{q^n}$, by relying on the Gaussian tail bound and summing $\mathsf{poly}(\lambda)$ terms (as $\sigma = \mathsf{poly}(\lambda)$). The quantities $\sin(2\pi xy/q)$ can be similarly approximated, using the Taylor approximation of sin up to degree $\mathsf{poly}(\lambda)$. To approximate $\widehat{f}(y)$, we claim that it suffices to compute the summation above for the summands $x \in \{1, 2, \ldots, \mathsf{poly}(\lambda)\}$. We use the tail bound for the Gaussian distribution. Let $C := \mathsf{poly}(\lambda)\frac{n}{2}\log q \leq \mathsf{poly}(\lambda)$. We have, for all $y \in \mathbb{Z}/q\mathbb{Z}$:

$$\begin{aligned}
\left| \sum_{x \in \mathbb{Z} \cap (C, q/2)} \sqrt{\vartheta_{\sigma,q}(x)}\ \sin \frac{2\pi x y}{q} \right| &\leq \sum_{x \in \mathbb{Z} \cap (C, q/2)} \sqrt{\vartheta_{\sigma,q}(x)} \\
&= \frac{1}{\sqrt{\rho_\sigma(\mathbb{Z})}} \sum_{x \in \mathbb{Z} \cap (C, q/2)} \sqrt{\sum_{k \in \mathbb{Z}} \rho_\sigma(x + kq)} \\
&\leq \frac{1}{\sqrt{\rho_\sigma(\mathbb{Z})}} \sum_{x \in \mathbb{Z} \cap (C, q/2)} \sum_{k \in \mathbb{Z}} \rho_{\sqrt{2}\sigma}(x + kq) \\
&\leq \frac{1}{\sqrt{\rho_\sigma(\mathbb{Z})}} \sum_{x \in \mathbb{Z} \setminus [-C, C]} \rho_{\sqrt{2}\sigma}(x) \\
&\leq \frac{\rho_{\sqrt{2}\sigma}(\mathbb{Z})}{\sqrt{\rho_\sigma(\mathbb{Z})}}\ \frac{C}{\sqrt{2}\sigma}\ \sqrt{2\pi \mathrm{e}}\ \mathrm{e}^{-\pi \frac{C^2}{2\sigma^2}} \quad \text{(by Lemma 2)} \\
&\leq \frac{1 + \sqrt{2}\sigma}{\sqrt{\sigma}}\ \frac{C}{\sqrt{2}\sigma}\ \sqrt{2\pi \mathrm{e}}\ \mathrm{e}^{-\pi \frac{C^2}{2\sigma^2}} \quad \text{(by Lemma 3)} \\
&\leq \mathsf{negl}(\lambda)/\sqrt{q^n}\ .
\end{aligned}$$

Finally, we observe that the truncated summation can be computed in time $\mathsf{poly}(\lambda)$. □

Lemma 17. *Let $n \geq 1$, $q \geq 3$ integers, $\sigma > 0$ a real number and $f : \mathbb{Z}/q\mathbb{Z} \to \mathbb{C}$ as in Theorem 4. Assume that $n, \sigma = \mathsf{poly}(\lambda)$ and $q = 2^{\mathsf{poly}(\lambda)}$, where λ is security parameter. Then we can build the following state in runtime $\mathsf{poly}(\lambda)$ and within error $\mathsf{negl}(\lambda)/\sqrt{q^n}$ in trace distance:*

$$\sum_{x \in \mathbb{Z}/q\mathbb{Z}} f(x)\,|x\rangle\ .$$

Proof. Let $C = \mathsf{poly}(\lambda)\frac{n}{2}\log q \leq \mathsf{poly}(\lambda)$. First, we build a state proportional to:

$$\sum_{x\in\mathbb{Z}\cap[-C,C]} \sqrt{\rho_\sigma(x)}\,|x\rangle \ .$$

Thanks to [GR02], such a state can be built in time $\mathsf{poly}(\lambda)$. This state is within trace distance $\mathsf{negl}(\lambda)/\sqrt{q^n}$ (by using the same reasoning as in the proof of Lemma 16) from

$$\sum_{x\in\mathbb{Z}/q\mathbb{Z}} \sqrt{\vartheta_{\sigma,q}}\,|x\rangle \ .$$

To complete the proof, it remains to add a -1 phase to the states $|x\rangle$ with $x < 0$. This can be implemented by using a control gate on the appropriate register of $|x\rangle$. □

2.5.2 On Condition 3 of Theorem 3

We now want to show that $q\cdot\min|\widehat{f}|^2$ is $1/\mathsf{poly}(\lambda)$. We first observe that, in most cases, the direct choice of $f_0 = \sqrt{\vartheta_{\sigma,q}}$ does not satisfy this condition. This motivates the introduction of ±1 phases.

Lemma 18. *Let $q \geq 2$ and integer and $\sigma \geq 1$ a real number. Let $f_0 = \sqrt{\vartheta_{\sigma,q}}$. We have:*

$$q\cdot\min|\widehat{f_0}|^2 \leq 32\sigma\cdot\max\left(\mathrm{e}^{-\frac{\pi\sigma^2}{4}}, \mathrm{e}^{-\frac{q^2}{4\sigma^2}}\right) \ .$$

Proof. Recall that

$$\forall e\in\mathbb{Z} : \vartheta_{\sigma,q}(e) = \frac{1}{\rho_\sigma(\mathbb{Z})}\sum_{k\in\mathbb{Z}}\exp\left(-\frac{|e+qk|^2}{\sigma^2}\right) \ .$$

Let $A, B : \mathbb{Z}/q\mathbb{Z}\to\mathbb{C}$ be defined as follows:

$$\forall y\in\mathbb{Z}/q\mathbb{Z} : A(y) := \sum_{x\in\mathbb{Z}\cap(-q/2,q/2]} \omega_q^{xy}\left(\sqrt{\vartheta_{\sigma,q}(x)} - \sqrt{D_{\mathbb{Z},\sigma}(x)}\right),$$

$$\forall y\in\mathbb{Z}/q\mathbb{Z} : B(y) := \sum_{x\in\mathbb{Z}\cap(-q/2,q/2]} \omega_q^{xy}\left(\frac{\sum_{k\in\mathbb{Z}}\rho_{\sqrt{2}\sigma}(x+kq)}{\sqrt{\rho_\sigma(\mathbb{Z})}} - \sqrt{D_{\mathbb{Z},\sigma}(x)}\right).$$

Then, for all $y\in\mathbb{Z}/q\mathbb{Z}$, it holds that

$$\begin{aligned}\widehat{f_0}(y) &= \frac{1}{\sqrt{q}}\sum_{x\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{xy}\sqrt{\vartheta_{\sigma,q}(x)} \\ &= \frac{1}{\sqrt{q}}\left(A(y) - B(y) + \sum_{x\in\mathbb{Z}/q\mathbb{Z}} \omega_q^{xy}\frac{\sum_{k\in\mathbb{Z}}\rho_{\sqrt{2}\sigma}(x+kq)}{\sqrt{\rho_\sigma(\mathbb{Z})}}\right).\end{aligned}$$

By the Poisson summation formula, the above term is equal to:

$$\frac{1}{\sqrt{q}}\left(A(y) - B(y) + \sum_{\ell\in\mathbb{Z}}\omega_q^{\ell y}\frac{\rho_{\sqrt{2}\sigma}(\ell)}{\sqrt{\rho_\sigma(\mathbb{Z})}}\right) = \frac{1}{\sqrt{q}}\left(A(y) - B(y) + \frac{\sqrt{2}\sigma}{\sqrt{\rho_\sigma(\mathbb{Z})}}\sum_{\ell\in\mathbb{Z}}\rho_{\frac{1}{\sqrt{2}\sigma}}\left(\ell+\frac{y}{q}\right)\right). \tag{2.22}$$

We now find upper bounds for the terms $A(y)$ and $B(y)$ and a lower bound for the remaining term of Equation (2.22). Using the fact that $\sqrt{\rho_\sigma} = \rho_{\sqrt{2}\sigma}$, we have, for all $y \in \mathbb{Z}/q\mathbb{Z}$:

$$\begin{aligned} B(y) &= \sum_{x \in \mathbb{Z} \cap (-q/2, q/2]} \omega_q^{xy} \left(\frac{\sum_{k \in \mathbb{Z}} \rho_{\sqrt{2}\sigma}(x + kq)}{\sqrt{\rho_\sigma(\mathbb{Z})}} - \sqrt{D_{\mathbb{Z},\sigma}(x)} \right) \\ &= \frac{\rho_{\sqrt{2}\sigma}(\mathbb{Z})}{\sqrt{\rho_\sigma(\mathbb{Z})}} \sum_{x \in \mathbb{Z} \cap (-q/2, q/2]} \omega_q^{xy} \left(\vartheta_{\sqrt{2}\sigma, q}(x) - D_{\mathbb{Z}, \sqrt{2}\sigma}(x) \right) . \end{aligned}$$

By the triangular inequality, it follows that

$$\begin{aligned} |B(y)| &\le \frac{\rho_{\sqrt{2}\sigma}(\mathbb{Z})}{\sqrt{\rho_\sigma(\mathbb{Z})}} \sum_{x \in \mathbb{Z} \cap (-q/2, q/2]} \left(\vartheta_{\sqrt{2}\sigma, q}(x) - D_{\mathbb{Z}, \sqrt{2}\sigma}(x) \right) \\ &\le \frac{\rho_{\sqrt{2}\sigma}(\mathbb{Z})}{\sqrt{\rho_\sigma(\mathbb{Z})}} \, q \, \mathrm{e}^{-\frac{q^2}{8\sigma^2}} \quad \text{(by Lemma 4)} . \end{aligned}$$

The use of Lemma 4 requires that $\sigma \le q/2$, which is implied by our assumptions.

We also have, for all $y \in \mathbb{Z}/q\mathbb{Z}$:

$$\begin{aligned} |A(y)| &\le \sum_{x \in \mathbb{Z} \cap (-q/2, q/2]} \left(\sqrt{\vartheta_{\sigma, q}(x)} - \sqrt{D_{\mathbb{Z},\sigma}(x)} \right) \quad \text{(by the triangular inequality)} \\ &\le \sum_{x \in \mathbb{Z} \cap (-q/2, q/2]} \mathrm{e}^{-\frac{q^2}{8\sigma^2}} \quad \text{(by Lemma 4)} \\ &= q \, \mathrm{e}^{-\frac{q^2}{8\sigma^2}} . \end{aligned}$$

Further, for every $y \in \mathbb{Z}$, it holds that

$$\sum_{\ell \in \mathbb{Z}} \rho_{\frac{1}{\sqrt{2}\sigma}} \left(\ell + \frac{y}{q} \right) \ge \mathrm{e}^{-\pi \frac{\sigma^2}{8}} .$$

To see this, note that the sum contains at least one term $\ell + y/q$ that has absolute value $\le 1/2$.

Going back to Equation (2.22) and using the triangular inequality, we see that, for all $y \in \mathbb{Z}/q\mathbb{Z}$:

$$\begin{aligned} |\widehat{f_0}(y)| &\le \frac{1}{\sqrt{q}} \left(\frac{\sqrt{2}\sigma}{\sqrt{\rho_\sigma(\mathbb{Z})}} \mathrm{e}^{-\pi \frac{\sigma^2}{8}} + q \, \mathrm{e}^{-\frac{q^2}{8\sigma^2}} + \frac{\rho_{\sqrt{2}\sigma}(\mathbb{Z})}{\sqrt{\rho_\sigma(\mathbb{Z})}} \, q \, \mathrm{e}^{-\frac{q^2}{8\sigma^2}} \right) \\ &\le \frac{1}{\sqrt{q}} \left(\sqrt{2\sigma} \mathrm{e}^{-\pi \frac{\sigma^2}{8}} + q \, \mathrm{e}^{-\frac{q^2}{8\sigma^2}} + \frac{\sqrt{2}\sigma + 1}{\sqrt{\sigma}} \, q \, \mathrm{e}^{-\frac{q^2}{8\sigma^2}} \right) \\ &\le \frac{4\sqrt{\sigma}}{\sqrt{q}} \left(\mathrm{e}^{-\pi \frac{\sigma^2}{8}} + q \, \mathrm{e}^{-\frac{q^2}{8\sigma^2}} \right) , \end{aligned}$$

where the second inequality follows from Lemma 3 and the third one from $\sigma \ge 1$. □

The result shows that, for Condition 3 of Theorem 3 to have a chance to hold, one is required to set the standard deviation parameter σ as $O(\sqrt{\log \lambda})$ or such that $q/\sigma =$

$O(\sqrt{\log \lambda})$. Unfortunately, in the first case, the $\mathsf{LWE}_{m,n,q,\sigma}$ problem can be solved efficiently [AG11], whereas the second one is too restrictive to enable cryptographic constructions.

To circumvent the above difficulty, we consider phases. Note that adding phases to f does not have any impact on the measurements and, therefore, after measuring the state, one still obtains an LWE sample with the same distribution. In the following lemmas, we show that the phases considered in Theorem 4 can sufficiently increase the quantity $q \cdot \min |\widehat{f}|^2$.

Lemma 19. *Let $q \geq 2$ an odd integer and $f : \mathbb{Z}/q\mathbb{Z} \to \mathbb{R}$ such that $f(-x) = -f(x)$ for all $x \in \mathbb{Z}/q\mathbb{Z} \setminus \{0\}$. Then we have*

$$q \cdot \min |\widehat{f}|^2 = q \cdot |\widehat{f}(0)| = |f(0)|^2 \ .$$

Proof. The discrete Fourier transform of f is given by

$$\begin{aligned}
\widehat{f}(y) &= \frac{f(0)}{\sqrt{q}} + \frac{1}{\sqrt{q}} \sum_{x \in \mathbb{Z} \cap (0,q/2)} f(x)\ \omega_q^{xy} + \frac{1}{\sqrt{q}} \sum_{x \in \mathbb{Z} \cap (-q/2,0)} f(x)\ \omega_q^{xy} \\
&= \frac{f(0)}{\sqrt{q}} + \frac{1}{\sqrt{q}} \sum_{x \in \mathbb{Z} \cap (0,q/2)} f(x) \left(\omega_q^{xy} - \omega_q^{-xy}\right) \quad (\text{as } \forall x \neq 0 : f(-x) = -f(x)) \\
&= \frac{f(0)}{\sqrt{q}} + i\ \frac{2}{\sqrt{q}} \sum_{x \in \mathbb{Z} \cap (0,q/2)} f(x)\ \sin \frac{2\pi xy}{q} \ ,
\end{aligned}$$

for all $y \in \mathbb{Z}/q\mathbb{Z}$. Since f is a real-valued function, the quantity $|\widehat{f}(y)|$ is no smaller than $|f(0)/\sqrt{q}|$ and the lower bound is reached at $y = 0$. □

We have the following lemma as a special case for the distribution $\vartheta_{\sigma,q}$.

Lemma 20. *Let $q \geq 2$ an odd integer, $\sigma > 0$ a real number and $f : \mathbb{Z}/q\mathbb{Z} \to \mathbb{C}$ as in Theorem 4. Then we have*

$$q \cdot \min |\widehat{f}|^2 \geq \frac{1}{1+\sigma} \ .$$

Proof. The statement $f(-x) = -f(x)$ holds for all $x \neq 0$. Therefore, using the positivity of $\vartheta_{\sigma,q}$, we obtain

$$q \cdot \min |\widehat{f}|^2 \geq \vartheta_{\sigma,q}(0) \geq \frac{1}{\rho_\sigma(\mathbb{Z})} \ .$$

Lemma 3 then gives the result. □

Adding ± 1 phases "exponentially" increases the success probability $p = q \cdot \min |\widehat{f}|^2$, when choosing $|f|^2 = \vartheta_{\sigma,q}$, which allows to fulfill Condition 3 of Theorem 2 under the constraint that $m, \sigma \leq \mathsf{poly}(\lambda)$. This improvement is crucial as otherwise we could not set m (which plays a significant role in the runtime of the algorithm) as some $\mathsf{poly}(\lambda)$.

2.5.3 On Condition 4 of Theorem 3

To instantiate Theorem 3, it now suffices to show that Condition 4 holds. Recall that it involves $Z_f(\mathbf{A})$, which is the normalization scalar ensuring that $\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}$ is unit vector.

Lemma 21. *Let $m, n \geq 1, q \geq 2$ integers, $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, f an amplitude function over $\mathbb{Z}/q\mathbb{Z}$, and $Z_f(\mathbf{A})$ as per Definition 17. Then we have:*

$$\left|\frac{Z_f(\mathbf{A})}{q^n} - 1\right| \leq \sum_{\substack{\mathbf{e}\neq\mathbf{e}' \\ \mathbf{e}-\mathbf{e}'\in\mathrm{Im}(\mathbf{A})}} |f|(\mathbf{e}) \cdot |f|(\mathbf{e}') \ .$$

Proof. For every vector $\mathbf{e} \in (\mathbb{Z}/q\mathbb{Z})^m$, let $|\mathrm{Im}(\mathbf{A}) + \mathbf{e}\rangle$ denotes the following state:

$$|\mathrm{Im}(\mathbf{A}) + \mathbf{e}\rangle := \sum_{\mathbf{x}\in(\mathbb{Z}/q\mathbb{Z})^n} |\mathbf{A}\mathbf{x} + \mathbf{e}\rangle \ .$$

(The state is purposefully not normalized.) For two vectors $\mathbf{e}, \mathbf{e}'$, we have

$$\langle \mathrm{Im}(\mathbf{A}) + \mathbf{e}' | \mathrm{Im}(\mathbf{A}) + \mathbf{e}\rangle = \begin{cases} q^n & \text{if } \mathbf{e}-\mathbf{e}' \in \mathrm{Im}(\mathbf{A}) \\ 0 & \text{otherwise} \end{cases} \ . \tag{2.23}$$

Then the $\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f}$ state can be expressed as follows:

$$\mathsf{C}\,|\mathsf{LWE}(\mathbf{A})\rangle_{m,n,q,f} = \frac{1}{\sqrt{Z_f(\mathbf{A})}} \sum_{\mathbf{e}\in(\mathbb{Z}/q\mathbb{Z})^m} f(\mathbf{e})\,|\mathrm{Im}(\mathbf{A}) + \mathbf{e}\rangle \ .$$

Therefore, we have

$$Z_f(\mathbf{A}) = \left\| \sum_{\mathbf{e}\in(\mathbb{Z}/q\mathbb{Z})^m} f(\mathbf{e})\,|\mathrm{Im}(\mathbf{A}) + \mathbf{e}\rangle \right\|^2 .$$

The above term is equal to:

$$\sum_{\mathbf{e},\mathbf{e}'} f(\mathbf{e})\overline{f(\mathbf{e}')}\,\langle \mathrm{Im}(\mathbf{A}) + \mathbf{e}' | \mathrm{Im}(\mathbf{A}) + \mathbf{e}\rangle = q^n \sum_{\substack{\mathbf{e},\mathbf{e}' \\ \mathbf{e}-\mathbf{e}'\in\mathrm{Im}(\mathbf{A})}} f(\mathbf{e})\overline{f(\mathbf{e}')} = q^n + q^n \sum_{\substack{\mathbf{e}\neq\mathbf{e}' \\ \mathbf{e}-\mathbf{e}'\in\mathrm{Im}(\mathbf{A})}} f(\mathbf{e})\overline{f(\mathbf{e}')} \ ,$$

where we used Equation (2.23). We obtain:

$$\left|\frac{Z_f(\mathbf{A})}{q^n} - 1\right| = \left| \sum_{\substack{\mathbf{e}\neq\mathbf{e}' \\ \mathbf{e}-\mathbf{e}'\in\mathrm{Im}(\mathbf{A})}} f(\mathbf{e})\overline{f(\mathbf{e}')} \right| .$$

The result follows from the triangular inequality. □

We now prove the following lemma.

Lemma 22. *Let $m, n \geq 1$ and $q \geq 2$ integers, and f an amplitude function over $\mathbb{Z}/q\mathbb{Z}$. Assume that q is prime. Let $\mathbf{A}$ be sampled uniformly in $(\mathbb{Z}/q\mathbb{Z})^{m\times n}$, and let $Z_f(\mathbf{A})$ be as per Definition 17. Then we have, for any $\delta > 0$:*

$$\mathbb{P}_{\mathbf{A}}\left(\left|\frac{Z_f(\mathbf{A})}{q^n} - 1\right| \geq \delta\right) \leq \frac{\sum_{\mathbf{e}\neq\mathbf{e}'} |f|(\mathbf{e}) \cdot |f|(\mathbf{e}')}{\delta \cdot q^{m-n}} \ .$$

Proof. We define:

$$S := \sum_{\substack{\mathbf{e}\neq\mathbf{e}' \\ \mathbf{e}-\mathbf{e}'\in \mathrm{Im}(\mathbf{A})}} |f|(\mathbf{e}) \cdot |f|(\mathbf{e}') \ ,$$

and view it as a random variable over the random choice of $\mathbf{A}$. By Lemma 21, we have that $|Z_f(\mathbf{A})/q^n - 1| \leq S$ holds for all $\mathbf{A}$. Further, by Markov's inequality, one obtains that $\mathbb{P}_{\mathbf{A}}(S \geq \delta) \leq \mathbb{E}_{\mathbf{A}}(S)/\delta$ holds for every $\delta > 0$. Using the linearity of the expectation, one obtains:

$$\begin{aligned}
\mathbb{E}_{\mathbf{A}}(S) &= \mathbb{E}_{\mathbf{A}}\left(\sum_{\mathbf{e}\neq\mathbf{e}'} \mathbb{1}_{\mathrm{Im}(\mathbf{A})}(\mathbf{e}-\mathbf{e}') \ |f|(\mathbf{e}) \cdot |f|(\mathbf{e}')\right) \\
&= \sum_{\mathbf{e}\neq\mathbf{e}'} \mathbb{P}_{\mathbf{A}}\left(\mathbf{e}-\mathbf{e}' \in \mathrm{Im}(\mathbf{A})\right) \ |f|(\mathbf{e}) \cdot |f|(\mathbf{e}') \\
&\leq \frac{1}{q^{m-n}} \sum_{\mathbf{e}\neq\mathbf{e}'} |f|(\mathbf{e}) \cdot |f|(\mathbf{e}') \ . \qquad (2.24)
\end{aligned}$$

The last inequality follows from the union bound (over all elements in the image of $\mathbf{A}$) and the fact that q is prime. □

We are particularly interested in the case where $|f| = \sqrt{\vartheta_{\sigma,q}}$. The following lemma allows us to apply the above result on this particular function.

Lemma 23. *Let $m \geq 1$ and $q \geq 2$ integers, and σ a real number such that $2 \leq \sigma \leq q/\sqrt{8m \ln q}$. Then we have:*

$$\sum_{\mathbf{e}\neq\mathbf{e}'} \sqrt{\vartheta_{\sigma,q}(\mathbf{e})}\sqrt{\vartheta_{\sigma,q}(\mathbf{e}')} \leq q^{\frac{m}{2}} + 1 \ .$$

Proof. First, note that the summation can be rewritten in the following way:

$$\sum_{\mathbf{e}\neq\mathbf{e}'} \sqrt{\vartheta_{\sigma,q}(\mathbf{e})}\sqrt{\vartheta_{\sigma,q}(\mathbf{e}')} = \left(\sum_{\mathbf{e}} \sqrt{\vartheta_{\sigma,q}(\mathbf{e})}\right)^2 - \sum_{\mathbf{e}} \vartheta_{\sigma,q}(\mathbf{e}) \ .$$

By positivity of the second term, it suffices to find an upper bound for the first one. We

rely on Lemma 4 to approximate $\vartheta_{\sigma,q}$ with $D_{\mathbb{Z}^m,\sigma}$. We have

$$\begin{aligned}
\sum_{\mathbf{e}\in\mathbb{Z}^m\cap(-\frac{q}{2},\frac{q}{2}]^m}\sqrt{\vartheta_{\sigma,q}(\mathbf{e})} &\leq \sum_{\mathbf{e}\in\mathbb{Z}^m\cap(-\frac{q}{2},\frac{q}{2}]^m}\left(\sqrt{D_{\mathbb{Z}^m,\sigma}(\mathbf{x})}+\mathrm{e}^{-\frac{q^2}{8\sigma^2}}\right) \quad \text{(by Lemma 4)}\\
&= \sum_{\mathbf{e}\in\mathbb{Z}^m\cap(-\frac{q}{2},\frac{q}{2}]^m}\left(\frac{\rho_{\sqrt{2}\sigma}(\mathbb{Z}^m)}{\sqrt{\rho_\sigma(\mathbb{Z}^m)}}\,D_{\mathbb{Z}^m,\sqrt{2}\sigma}(\mathbf{x})+\mathrm{e}^{-\frac{q^2}{8\sigma^2}}\right)\\
&\leq \frac{\rho_{\sqrt{2}\sigma}(\mathbb{Z}^m)}{\sqrt{\rho_\sigma(\mathbb{Z}^m)}}+q^m\mathrm{e}^{-\frac{q^2}{8\sigma^2}}\\
&\leq \frac{(1+\sqrt{2}\sigma)^m}{\sqrt{\sigma^m}}+q^m\mathrm{e}^{-\frac{q^2}{8\sigma^2}} \quad \text{(by Lemma 3)}\\
&\leq (2\sqrt{\sigma})^m+q^m\mathrm{e}^{-\frac{q^2}{8\sigma^2}}\ .
\end{aligned}$$

Since $\sigma \leq q/\sqrt{8m\ln q}$, we have that the last term is ≤ 1. Finally, note that the same upper bound on σ also implies that $2\sqrt{\sigma}\leq\sqrt{q}$. □

We can now conclude, by combining Lemma 22 and Lemma 23 with $\delta = q^{-n}$.

Lemma 24. *Let $m \geq n \geq 1$, $q \geq 2$ integers, $\sigma > 0$ a real number and $f : \mathbb{Z}/q\mathbb{Z} \to \mathbb{C}$ as in Theorem 4. Assume that q is prime and $2 \leq \sigma \leq q/\sqrt{8m\ln q}$. Let $\mathbf{A}$ be sampled uniformly from $(\mathbb{Z}/q\mathbb{Z})^{m\times n}$, and let $Z_f(\mathbf{A})$ as per Definition 17. Then we have*

$$\mathbb{P}_{\mathbf{A}}\left(\left|\frac{Z_f(\mathbf{A})}{q^n}-1\right|\geq q^{-n}\right) \;\leq\; q^{2n-m}(q^{\frac{m}{2}}+1)\ .$$

2.6 On the security of some lattice-based SNARKs

Succinct Non-Interactive Arguments of Knowledge (SNARKs) are cryptographic schemes whose purpose is to prove NP statements with a succinct proof and fast verification, as a function of the statement size. They must satisfy the property of knowledge soundness: informally speaking, if a malicious prover manages to build a proof that passes verification, then one can extract from its description and execution a valid witness for the proved statement. Several candidate SNARKs based on lattices in the standard model [GMNO18, NYI+20, ISW21, SSEK22, CKKK23, GNSV23] have been proposed. As these constructions rely on assumptions related to lattices, they are often conjectured secure even against quantum adversaries. In terms of parameters, several of those SNARKs require an exponential gap between the noise and the modulus, i.e., a large q/σ. For example, one may choose $q/\sigma = 2^\lambda, \sigma = \mathsf{poly}(\lambda), q = 2^{\Theta(\lambda)}$, and $n = \Theta(\lambda^2/\log\lambda)$. For this parametrization, the runtime of the best known algorithm grows as $\exp(\Omega(\lambda))$.

All these suggestions assume the hardness of some type of knowledge assumption, i.e., an assumption that formalizes the intuition that an algorithm cannot achieve a given task without knowing a specific information. This intuition is formalized using extractor algorithms. The specific knowledge assumptions used in those schemes are typically defined in terms of LWE-based ciphertexts (also sometimes called encodings) of a symmetric encryption scheme.

To simplify the discussion, we now focus on the constructions from [ISW21, SSEK22, CKKK23]. The discussion can be adapted to the other schemes (see Section 2.6.4). The corresponding encryption scheme handles plaintexts defined modulo an integer p, with ciphertexts that are vectors modulo a much larger integer q, such that the scheme enjoys a linear homomorphism property: given $y_1, \ldots, y_m \in \mathbb{Z}/p\mathbb{Z}$ and ciphertexts $\mathbf{ct}_1, \ldots, \mathbf{ct}_m$ decrypting to $a_1, \ldots, a_m$, the vector $\sum_i y_i \mathbf{ct}_i$ decrypts to $\sum y_i a_i$. It is then assumed that the only way to compute a valid ciphertext is to take a linear combination of the available ciphertexts (variants may be used in different schemes). To obtain SNARKs, this is formalized in terms of the existence of an efficient extractor: given the $\mathbf{ct}_i$'s, the description of the algorithm producing a new ciphertext and its internal randomness, some efficient extractor recovers scalars y_i's modulo p such that $\mathbf{ct} = \sum_i y_i \mathbf{ct}_i$.

We observe that the knowledge assumptions involved in those schemes can be expressed in terms of the knapsack version of LWE. The knLWE problem asks to recover $\mathbf{e}$ from the input $(\mathbf{B}, \mathbf{Be})$ where $\mathbf{B}$ is a uniformly chosen matrix from $(\mathbb{Z}/q\mathbb{Z})^{(m-n)\times m}$, for some integers $m > n \geq 1$ and $q \geq 2$. We are in a regime of parameters where $\mathbf{e}$ is uniquely determined from $\mathbf{Be}$, with overwhelming probability over the uniform choice of $\mathbf{B}$. We identify the matrix $\mathbf{B}$ with the matrix $(\mathbf{ct}_1, \ldots, \mathbf{ct}_m)$. Note that it is not uniform, we can pretend it is as it is computationally indistinguishable from uniform under some LWE parametrization. We then argue that the knowledge assumption is quantumly broken, by observing that our witness-oblivious quantum LWE sampler can be turned into a witness-oblivious knLWE sampler by relying on the randomized Karp reduction from LWE to knLWE from [MM11]. As some of the considered schemes rely on algebraic variants of LWE, such as Ring-LWE [SSTX09, LPR10] or Module-LWE [BGV12, LS15], we extend the witness-oblivious LWE sampler to those settings. The analysis extends without difficulty, except for difficulties arising from the fact that the considered rings are not fields.

2.6.1 Module Learning With Errors

All the SNARK constructions mentioned above can be framed into an algebraic variant of LWE called MLWE, which captures LWE and the Ring Learning With Errors problem (RLWE) [SSTX09, LPR10]. To recall the definition of MLWE and adapt the results on oblivious LWE sampling to MLWE, we first provide some reminders.

Let $d \geq 1$ be a power-of-2 integer. The cyclotomic ring R of degree d is $\mathbb{Z}[x]/\langle x^d + 1\rangle$. Each element of R is a polynomial of degree at most $d-1$ with integer coefficients. We let $\phi : R \to \mathbb{Z}^d$ denote the map that sends each element $\sum_{i<d} a_i x^i \in R$ to the vector $(a_0, \ldots, a_{d-1})^\intercal \in \mathbb{Z}^d$. For every element $a \in R$, we define $\mathrm{rot}(a)$ as the matrix whose i-th column is $\phi(x^{i-1}a \bmod x^d + 1)$, for all $1 \leq i \leq d$. Then we have $\phi(a \cdot b) = \mathrm{rot}(a)\phi(b)$ for all $a, b \in R$. Let $q \geq 2$ be an integer. Both ϕ and rot are extended to the quotient ring R/qR. Similarly, we extend ϕ to $(R/qR)^m$ and rot to $(R/qR)^{m\times n}$ for any integers $m, n \geq 1$.

For a distribution χ over $\mathbb{Z}/q\mathbb{Z}$, we define $\chi^{\otimes d}$ as the distribution over R/qR obtained by independently sampling each coefficient from χ. The notation is extended to distributions over $(R/qR)^m$ for any $m \geq 1$.

Module Learning With Errors (MLWE) is a variant of LWE introduced and studied in [BGV12, LS15]. It is defined by replacing $\mathbb{Z}/q\mathbb{Z}$ by R/qR in the LWE definition.

Definition 20 (MLWE). *Let $m \geq n \geq 1, q \geq 2$ be integers, R be a cyclotomic ring of degree a power-of-2 integer d and χ be a distribution over $\mathbb{Z}/q\mathbb{Z}$. The parameters m, n, d, q and χ are functions of some security parameter λ. Let $\mathbf{A} \in (R/qR)^{m\times n}$, $\mathbf{s} \in (R/qR)^n$ be sampled uniformly and $\mathbf{e} \in (R/qR)^m$ be sampled from from $\chi^{\otimes dm}$. The search $\mathsf{MLWE}_{m,n,d,q,\chi}$ problem is to find $\mathbf{s}$ and $\mathbf{e}$ given the pair $(\mathbf{A}, \mathbf{As}+\mathbf{e})$. The vectors $\mathbf{s}$ and $\mathbf{e}$ are respectively called the secret and the noise.*

Whenever χ is equal to the folded discrete Gaussian distribution $\vartheta_{\sigma,q}$ for some $\sigma > 0$, we overwrite the notations as $\mathsf{MLWE}_{m,n,d,q,\sigma}$.

We now show how Theorem 4 can be extended to MLWE. The MLWE problem can be viewed as a special case of LWE. Concretely, an $\mathsf{MLWE}_{m,n,d,q,\chi}$ instance $(\mathbf{A}, \mathbf{b} = \mathbf{As}+\mathbf{e}) \in (R/qR)^{m\times n} \times (R/qR)^m$ is mapped to the $\mathsf{LWE}_{md,nd,q,\chi}$ instance

$$(\mathrm{rot}(\mathbf{A}), \phi(\mathbf{b}) = \mathrm{rot}(\mathbf{A})\phi(\mathbf{s}) + \phi(\mathbf{e})) \ \in \ (\mathbb{Z}/q\mathbb{Z})^{md\times nd} \times (\mathbb{Z}/q\mathbb{Z})^{md} \ .$$

Our goal is to use Theorem 3 with these specific matrices. By the identity above, one can observe that Conditions 1 and 2 are not impacted by the change from LWE to MLWE. Condition 3 is related to the Gaussian elimination subroutine of Algorithm 1. We note that there is no q such that R/qR is a field for $d > 2$ (as opposed to $\mathbb{Z}/q\mathbb{Z}$ with q). Instead, we choose q prime such that $q = 3 \bmod 8$. In that case, the ring R/qR is isomorphic to $\mathbb{F}_{q^{d/2}} \times \mathbb{F}_{q^{d/2}}$. For $m \geq n \cdot \omega(\log \lambda)$, a uniform $\mathbf{A} \in (R/qR)^{m\times n}$ has a set of n rows that form an invertible matrix, with probability $1 - \mathsf{negl}(\lambda)$. This allows us to adapt the Gaussian elimination subroutine of Algorithm 1 to the module setting when $d > 2$. Overall, for such a modulus q, Condition 3 is also not impacted by the change from LWE to MLWE.

We now focus on Condition 4, which was proved in Subsection 2.5.3 to be fulfilled in the LWE case for a specific choice of amplitude function (defined in Theorem 4). We keep the same amplitude function, and adapt Lemma 24 to the module setting.

Lemma 25. *Let $m \geq n \geq 1, q \geq 2$ integers, $\sigma > 0$ a real number, $f : \mathbb{Z}/q\mathbb{Z} \to \mathbb{C}$ as in Theorem 4 and R a cyclotomic ring of degree a power-of-2 integer d. Assume that $d > 2$, q is prime and satisfies $q = 3 \bmod 8$, and $2 \leq \sigma \leq \sqrt{q/(8m \ln q)}$. Let $\mathbf{A}$ be sampled uniformly from $(R/qR)^{m\times n}$, and let $Z_f(\mathrm{rot}(\mathbf{A}))$ as per Definition 17. Then we have*

$$\mathbb{P}_{\mathbf{A}}\left(\left|\frac{Z_f(\mathrm{rot}(\mathbf{A}))}{q^{nd}} - 1\right| \geq q^{-nd}\right) \ \leq \ q^{(2n-\frac{m}{2})d}(q^{\frac{md}{4}} + 1) \ .$$

Proof. We follow the proof of Lemma 24 in Subsection 2.5.3. Lemma 21 applies without any change. For Lemma 22, the only step that needs to be adapted is Equation (2.24). We have, for $\mathbf{e} \neq \mathbf{e}' \in (R/qR)^m$:

$$\mathbb{P}_{\mathbf{A}}\left(\mathbf{e} - \mathbf{e}' \in \mathrm{Im}(\mathbf{A})\right) \leq q^{dn} \max_{\mathbf{s}\in(R/qR)^n} \mathbb{P}_{\mathbf{A}}\left(\mathbf{e} - \mathbf{e}' = \mathbf{As}\right) \ \leq q^{dn} \cdot q^{-\frac{dm}{2}} \ .$$

where we used the union bound in the first inequality and considered only one of the components of $R/qR \simeq \mathbb{F}_{q^{d/2}} \times \mathbb{F}_{q^{d/2}}$ in the second inequality. As a result, the term "q^{m-n}" in statement of Lemma 22 is replaced by $q^{(m/2-n)d}$. The proof of Lemma 23 is unchanged, but we strengthen the upper bound on σ to $\sigma \leq \sqrt{q/(8m \ln q)}$ to be able to replace the term"$q^{m/2}$" in statement of Lemma 23 by $q^{md/4}$. This completes the proof of Lemma 25. □

Using the above, we obtain the following adaptation of Theorem 4.

Theorem 5. *Let $m, n, d, q, R, \sigma, \lambda$ as in Definition 20. Assume that $m, \log q \leq \mathsf{poly}(\lambda)$, $d > 2$, and q is prime with $q = 3 \bmod 8$. Assume further that the parameters satisfy the following conditions:*

$$m \geq n\sigma \cdot \omega(\log \lambda) \quad \text{and} \quad 2 \leq \sigma \leq \sqrt{\frac{q}{8m \ln q}} .$$

Then Algorithm 1 runs in time $\mathsf{poly}(\lambda)$ and, for a proportion $1 - \mathsf{negl}(\lambda)$ of matrices $\mathbf{A} \in (R/qR)^{m \times n}$, it outputs a quantum state $|\varphi\rangle$ such that $D_{\mathrm{tr}}(|\varphi\rangle, |\mathbf{0}\rangle \mathsf{C} |\mathsf{LWE}(\mathrm{rot}(\mathbf{A}))\rangle_{q,f} |0\rangle) = \mathsf{negl}(\lambda)$.

In particular, if $\mathsf{MLWE}_{m,n,d,q,\sigma}$ is hard, then there exists a $\mathsf{poly}(\lambda)$-time quantum witness-oblivious $\mathsf{MLWE}_{m,n,d,q,\sigma}$ sampler.

As in the LWE context, we could use Karp reductions from MLWE with one parametrization to MLWE with another parametrization to significantly extend the range of allowed MLWE parametrizations in Theorem 5. We could notably throw away superfluous samples, switch from one modulus to another [LS15] or trade modulus for dimension [AD17].

2.6.2 Knapsack MLWE

We generalize the knapsack variant of LWE from [MM11] to modules.

Definition 21 (knMLWE)**.** *Let m, n, d, q, R, χ as in Definition 20. Let $\mathbf{B} \in (R/qR)^{n \times m}$ and $\mathbf{e} \in (R/qR)^m$ be sampled from $\chi^{\otimes dm}$. The search $\mathsf{knMLWE}_{m,n,d,q,\chi}$ problem is to find $\mathbf{e}$ from $(\mathbf{B}, \mathbf{Be})$.*

Whenever χ is equal to the folded discrete Gaussian distribution $\vartheta_{\sigma,q}$, we overwrite the notation as $\mathsf{knMLWE}_{m,n,d,q,\sigma}$.

A Karp reduction from LWE to its knapsack form was given in [MM11, Le. 4.8]. To extend it to modules, one needs to be able to perform linear algebra efficiently and that uniform matrices over $(R/qR)^{m \times (m-n)}$ contain a subset of $m-n$ rows that is invertible with sufficiently high probability. These conditions were already required to obtain Theorem 5, so we can keep the same parameter constraints here. Using Theorem 5 and Lemma 8, we obtain the following result.

Theorem 6. *Let $m, n, d, q, R, \sigma, \lambda$ as in Definition 20. Assume that $m, \log q \leq \mathsf{poly}(\lambda)$, $d > 2$, and q is prime with $q = 3 \bmod 8$. Assume further that the parameters satisfy the following conditions:*

$$m \geq (m-n)\sigma \cdot \omega(\log \lambda) \quad \text{and} \quad 2 \leq \sigma \leq \sqrt{\frac{q}{8m \ln q}} .$$

Assume that $\mathsf{MLWE}_{m,m-n,d,q,\sigma}$ is hard. Then there exists a $\mathsf{poly}(\lambda)$-time algorithm that produces samples $(\mathbf{B}, \mathbf{Be})$ that are within statistical distance $\mathsf{negl}(\lambda)$ from those obtained by sampling $\mathbf{B}$ uniformly and $\mathbf{e}$ from $\vartheta_{\sigma,q}^{\otimes dm}$, and for which there exists no efficient extractor algorithm that would recover the witness $\mathbf{e}$.

We discuss some restrictions of Theorem 6. For a large number of columns m, the matrix $\mathbf{B}$ must be almost square for the condition $m \geq (m-n)\cdot\omega(\log\lambda)$ to be satisfied. If we are interested in much fewer rows than columns (which is the case in our applications), one may use Theorem 6 with a near-square matrix and then throw away the superfluous rows. This preserves obliviousness.

Another restriction of Theorem 6 is that the modulus q is required to be prime and to satisfy $q = 3 \bmod 8$. However, in most applications, the modulus is not of that form: for example, in [ISW21], the considered moduli are powers of 2. We want to use modulus switching for knMLWE, but there are two difficulties. First, modulus switching introduces a small rounding error. We make it part of the weight vector $\mathbf{e}$ by putting the matrix $\mathbf{B}$ in canonical form. Second, in our application, the matrix $\mathbf{B}$ is given as input to the instance sampler rather than generated by the sampler itself. For this aspect, we note that the sampler of Theorem 6 satisfies this property: given as input a uniform matrix $\mathbf{B}$, with probability $1 - \mathsf{negl}(\lambda)$, it outputs $\mathbf{Be}$ such that $\mathbf{e}$ is within $\mathsf{negl}(\lambda)$ statistical distance from $\vartheta_{\sigma,q}^{\otimes dm}$.

Algorithm 2 is designed to handle those aspects. Step 1 puts the input matrix in canonical form. Step 4 performs a modulus switch for the non-trivial component $\overline{\mathbf{B}}$ of the canonical form. The new modulus q' is prime and satisfies $q' = 3 \bmod 8$ (note that such primes are frequent). By choice of $\mathbf{E}$ and τ, the resulting matrix $\overline{\mathbf{B}}'$ is within negligible statistical distance from uniform (this may be proved using standard facts on discrete Gaussian distributions, such as done for example in [BLP+13]). Step 6 randomizes to hide the canonical form to obtain a uniform matrix $\overline{\mathbf{B}}$. Step 7 calls the algorithm from Theorem 6 to obtain $\overline{\mathbf{b}} = \overline{\mathbf{B}}\mathbf{e}$ for some unknown $\mathbf{e}$ (as discussed above, the algorithm from Theorem 6 satisfies the property that it outputs a vector for a given matrix, rather than sampling them together). Finally, Step 8 sends $\overline{\mathbf{b}}$ back to R/qR.

We can see that the output $\mathbf{b}$ is of the correct form. First, note that we have $\overline{\mathbf{T}}^{-1}\overline{\mathbf{b}} = (\mathbf{I} \mid \overline{\mathbf{B}}')\mathbf{e}$. Rounding from modulus q' to modulus q gives $\frac{q}{q'}(\mathbf{I} \mid \overline{\mathbf{B}}')\mathbf{e} + \mathbf{f}$ for some small-magnitude vector $\mathbf{f}$. Letting $\mathbf{e}_1$ denote the first n entries of $\mathbf{e}$ and $\mathbf{e}_2$ the remaining $m-n$, and using the definition of $\overline{\mathbf{B}}'$, we see that $\frac{q}{q'}(\mathbf{I} \mid \overline{\mathbf{B}}')\mathbf{e} + \mathbf{f}$ is of the form $\overline{\mathbf{B}}\mathbf{e}_2 + \mathbf{g}$ for some small magnitude vector $\mathbf{g}$. This can be rewritten as $(\mathbf{I} \mid \overline{\mathbf{B}})(\mathbf{g}^\intercal|\mathbf{e}_2^\intercal)^\intercal$. Multiplying by $\mathbf{T}$ gives that $\mathbf{b}$ is indeed of the correct form. Further, the transformation preserves obliviousness. Assume by contradiction that an extractor can recover $(\mathbf{g}^\intercal|\mathbf{e}_2^\intercal)^\intercal$. Then it can in particular recover $\mathbf{e}_2$. From $\mathbf{e}_2$, it can recover $\mathbf{e}_1$ as $\mathbf{e}_1 = \overline{\mathbf{T}}^{-1}\overline{\mathbf{b}} - \overline{\mathbf{B}}'\mathbf{e}_2$. This contradicts the fact that the algorithm from Theorem 6 is oblivious.

Finally, let us comment on the failure probability of Step 1 (as q' is prime and satisfies $q' = 3 \bmod 8$, the failure probability of Step 5 is very low). Depending on the arithmetic shape of q and the values of m and n, this value could possibly be non-negligible. Fortunately, in all the applications, the number of columns m is orders of magnitude higher than the number of rows n, so that, with overwhelming probability, we can find a subset of n columns that is invertible. It then suffices to apply Algorithm 2 after an appropriate reordering of the columns.

2.6.3 SNARKs from linear-only vector encryption

For constructing SNARKs, the authors of [ISW21, SSEK22, CKKK23] adapt the approaches of [BCI+13] and [BISW17] to the LWE setting (the possibility of adaptation

Algorithm 2 Witness-oblivious knMLWE sampler for arbitrary q

Parameters: m, n, q, d, σ and λ as in Definition 20.

Input: $\mathbf{B} \in (R/qR)^{n \times m}$.
Output: A vector $\mathbf{b} \in (R/qR)^n$.

1: Compute a matrix $\mathbf{T}$ such that $\mathbf{TB} = (\mathbf{I} \mid \overline{\mathbf{B}})$. If $\mathbf{T}$ does not exist, then abort.
2: Set q' as the smallest prime larger than q such that $q' = 3 \bmod 8$.
3: Set $\tau := q'/q \cdot \sqrt{\lambda}$.
4: Set $\overline{\mathbf{B}}' := \frac{q'}{q}\overline{\mathbf{B}} + \mathbf{E}$ with each entry of $\mathrm{rot}(\mathbf{E})$ sampled from $D_{\mathbb{Z}^d - \frac{q'}{q}\mathrm{rot}(\overline{\mathbf{B}}_{ij}),\tau}$ for all i, j.
5: Sample $\overline{\mathbf{T}} \in (R/q'R)^{n \times n}$. If it is not invertible, then abort.
6: Set $\overline{\mathbf{B}} := \overline{\mathbf{T}}(\mathbf{I} \mid \overline{\mathbf{B}}')$.
7: Apply the sampler from Theorem 6 with parameters m, n, q', d, σ on $\overline{\mathbf{B}}$ to obtain $\overline{\mathbf{b}} = \overline{\mathbf{B}}\mathbf{e}$.
8: Compute $\mathbf{b} := \mathbf{T}^{-1}\lfloor \frac{q}{q'}(\overline{\mathbf{T}}^{-1}\overline{\mathbf{b}} \bmod q') \rceil \bmod q$.
9: Return $\mathbf{b}$.

to LWE was actually suggested in [BCI+13], see Remark 5.19 therein). They use secret-key vector encryption schemes that are linear-only homomorphic. The plaintexts belong to an R/pR-module, whereas the ciphertexts belong to an R/qR-module for some integers $q > p \geq 2$ where $R = \mathbb{Z}[x]/\langle x^d + 1\rangle$ for some power-of-2 degree d. Such schemes allow the players to compute R/pR-linear functions of the ciphertexts but no other function than those ones. This is called to the linear-only property.

Definition 22 (Vector Encryption over Cyclotomic Fields)**.** *Let $\ell, m, n \geq 1$ be integers, $R = \mathbb{Z}[x]/\langle x^d + 1\rangle$ with a power-of-2 degree d, $q > p \geq 2$ be integers, and S be a subset of $(R/pR)^m$. All these are functions of the security parameter λ. A secret-key linearly-homomorphic vector encryption scheme with the message space $(R/pR)^\ell$ and the ciphertext space $(R/qR)^n$ is a tuple of algorithms $\Pi_{\mathsf{Enc}} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Add})$ with the following specifications.*

- $\mathsf{Gen}(1^\lambda) \mapsto (pp, sk)$*: Given the security parameter λ, it outputs public parameters pp and a secret key sk;*
- $\mathsf{Enc}(sk, \mathbf{v}) \mapsto \mathbf{ct}$*: Given the secret key sk and a vector $\mathbf{v} \in (R/pR)^\ell$, it outputs a ciphertext $\mathbf{ct} \in (R/qR)^n$;*
- $\mathsf{Dec}(sk, \mathbf{ct}) \mapsto \mathbf{v}/\perp$*: Given the secret key sk and a ciphertext $\mathbf{ct}$, it outputs a vector $\mathbf{v} \in (R/pR)^\ell$ or a special symbol $\perp$;*
- $\mathsf{Add}(pp, \{\mathbf{ct}_i\}_i, \{y_i\}_i) \mapsto \mathbf{ct}^*$*: Given the public parameters pp, a collection of ciphertexts $\{\mathbf{ct}_i\}_i$, and a collection of scalars $\{y_i\}_i$ from R/pR, it outputs a ciphertext $\mathbf{ct}^*$.*

Moreover, Algorithm Add *satisfies the following property:*

- *Additive homomorphism with respect to the set S: For all security parameters λ, all*

vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ *from* $(R/pR)^\ell$*, and* $(y_1, \ldots, y_m) \in S$*, it holds that*

$$\mathbb{P}\left(\mathsf{Dec}(sk, \mathbf{ct}^*) = \sum_{i=1}^{m} y_i \mathbf{v}_i \;\middle|\; \begin{array}{l} (pp, sk) \leftarrow \mathsf{Gen}(1^\lambda) \\ \mathbf{ct}_i \leftarrow \mathsf{Enc}(sk, \mathbf{v}_i) \\ \mathbf{ct}^* \leftarrow \mathsf{Add}(pp, \{\mathbf{ct}_i\}_i, \{y_i\}_i) \end{array}\right) = 1 - \mathsf{negl}(\lambda) \ .$$

The set S controls the level of homomorphic operations that are allowed. In [ISW21, Th. 3.12], it is showed that the proposed vector encryption scheme allows homomorphic operations with respect to the whole set $(R/pR)^m$ when q is chosen sufficiently large. In [SSEK22, Th. 2], this set is more restricted.

When using lattice problems, the functionality of Definition 22 is obtained as follows. One typically relies on an LWE/MLWE encryption scheme with plaintexts defined modulo p. Given ciphertexts $\mathbf{ct}_i$'s, which are vectors modulo q, and scalars y_i, the Add algorithm first computes the linear combination $\sum_i y_i \mathbf{ct}_i$ and then possibly adds some large amount of noise (a technique typically referred to as noise flooding or noise smudging) or rounds. These operations can be publicly implemented. In terms of security, the ciphertexts $\mathbf{ct}_i$ are designed to be computationally indistinguishable from uniform, under an appropriate LWE/MLWE parametrization, to ensure the IND-CPA security of the vector encryption scheme. We note that all the schemes we consider follow this blueprint.

In this work, we are particularly interested in the linear-only security property. Note that the adversary is allowed to be a quantum algorithm in the context of post-quantum cryptography. This is taken into account in the following definition.

Definition 23 (Linear-Only Against Quantum Adversaries)**.** *A vector encryption scheme* $\Pi_{\mathsf{Enc}} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Add})$ *is linear-only if for all QPT algorithms* $\mathcal{A}$*, there exists a valid QPT extractor* $\mathcal{E}$ *such that for all security parameters* λ*, auxiliary mixed states* ρ *over* $\mathbb{C}^{2^{\mathsf{poly}(\lambda)}}$*, and any QPT plaintext generator* $\mathcal{M}$*, it holds that*

$$\mathbb{P}\left(\mathsf{ExptLinearExt}_{\Pi_{\mathsf{Enc}}, \mathcal{A}, \mathcal{M}, \mathcal{E}, \rho}(1^\lambda) = 1\right) = \mathsf{negl}(\lambda) \ ,$$

where the experiment $\mathsf{ExptLinearExt}_{\Pi_{\mathsf{Enc}}, \mathcal{A}, \mathcal{M}, \mathcal{E}, \rho}(1^\lambda)$ *is defined as follows.*

1. *The challenger samples the public parameters and the secret key* $(pp, sk) \leftarrow \mathsf{Gen}(1^\lambda)$*, together with* m *vectors* $(\mathbf{v}_1, \ldots, \mathbf{v}_m) \leftarrow \mathcal{M}(1^\lambda, pp)$*. It computes the ciphertexts* $\mathbf{ct}_i \leftarrow \mathsf{Enc}(sk, \mathbf{v}_i)$ *for all* i*.*

2. *Then it runs the extraction process with the outputs as follows:*

$$\left((\mathbf{ct}'_1, \ldots, \mathbf{ct}'_k), \mathbf{\Pi}\right) \leftarrow \langle \mathcal{A}, \mathcal{E} \rangle (1^\lambda, |pp, \mathbf{ct}_1, \ldots, \mathbf{ct}_m\rangle \otimes \rho, |0\rangle) \ .$$

   Let $\mathbf{V}' = (\mathbf{v}_1 | \ldots | \mathbf{v}_m)\mathbf{\Pi}$*. The output of the experiment is* 1 *if there exists an* $i \leq m$ *such that* $\mathsf{Dec}(sk, \mathbf{ct}_i) \neq \perp$ *and* $\mathsf{Dec}(sk, \mathbf{ct}_i) \neq \mathbf{v}'_i$ *where* $\mathbf{v}'_i$ *is the* i*-th column of* $\mathbf{V}'$*. Otherwise, the experiment outputs* 0*.*

As discussed in [ISW21, Rem. 3.6], the requirement that the extractor must succeed for all auxiliary inputs ρ is too strong. In particular, no polynomial-time extractor exists if ρ is the output of a one-way function that the extractor must invert in order to analyze the behaviour of the sampler. In all cases that we consider, in the classical setting, the auxiliary inputs are sampled as uniform strings. In the quantum setting, such a string

can be simulated by Hadamard gates and projective measurements. Therefore, in our applications, we choose ρ to be null.

In Definition 23, the adversary is given m ciphertexts $\mathbf{C} := (\mathbf{ct}_1|\dots|\mathbf{ct}_m) \in (R/qR)^{n\times m}$ and is supposed to output k distinct small linear combinations of these vectors, namely $\mathbf{C}\pi_1, \dots, \mathbf{C}\pi_k$ where $\pi_i \in R^m$ is the i-th column of $\mathbf{\Pi}$, with each entry in $(-p/2, p/2]$. It asks the extractor to find the exact value of the matrix $\mathbf{\Pi}$.

We observe that $(\mathbf{C}, \mathbf{C}\pi_i)$ is a knMLWE instance, for all i. In [ISW21], the authors use MLWE with $d = 2$, whereas much larger degrees are considered in [SSEK22, CKKK23]. In all cases, the knMLWE number of columns is very large, of the order of 2^{20}, whereas the number of rows corresponds to MLWE-based ciphertexts is of the order of 2^{12}. The plaintext modulus p has a bit-size that is much smaller than the one of the ciphertext modulus q. The latter may have up to 100 bits in [ISW21].

We attack the linear-only property as follows. Let $\mathbf{C} = (\mathbf{ct}_1|\dots|\mathbf{ct}_m) \in (R/qR)^{n\times m}$ with $\mathbf{ct}_i = \mathsf{Enc}(\mathsf{sk}, \mathbf{v}_i)$ for all i. Consider a quantum knMLWE sampler as in Subsection 2.6.2. Note that $\mathbf{C}$ is not statistically uniform but only computationally indistinguishable from uniform. This assumption holds for all secret-key encryption schemes used in the considered SNARK constructions. We claim that the sampler is still oblivious in this situation: if an extractor exists for $\mathbf{C}$'s of this form, then we can distinguish $\mathbf{C}$ from uniform (note that one can efficiently verify the validity of the extracted witness). Now, let $\mathbf{Ce}$ be the output of the sampler, with $\mathbf{e} = (e_1, \dots, e_m)^\intercal$ small. It then holds that $e_1\mathbf{ct}_1 + \dots + e_m\mathbf{ct}_m$ decrypts to

$$e_1\mathbf{v}_1 + \dots + e_m\mathbf{v}_m = \mathbf{Ve} \bmod p \ ,$$

as Π_{Enc} is additively-homomorphic modulo p. Since $\mathbf{Ce}$ is a hard instance sampled obliviously, extracting $\mathbf{e}$ out of $\mathbf{Ce}$ is not possible for QPT extractors, except with negligible probability. This contradicts Condition 2 of Definition 23.

2.6.4 SNARKs from encoding schemes

In [GGPR13], specific encoding schemes were introduced to build SNARKs from assumptions related to the discrete logarithm problem. Later, the framework was applied to lattices for constructing presumably post-quantum SNARKs [GMNO18, NYI+20, GNSV23]. The constructions in [GMNO18, NYI+20] consider encodings for finite fields, while encodings for rings of the form R/pR are designed. Concretely, the message space is of the form R/pR for some integer p and ring of integers R of a number field, and the codeword space is $(R/qR)^n$ for some integers $n, q > p$. The ring R is usually chosen to be the ring of integers of a power-of-2 cyclotomic field. We recall the definition of encoding schemes, keeping only the parameters and properties that are relevant for our purposes.

Definition 24 (Encoding Schemes Over Cyclotomic Rings)**.** *Let $m, n \geq 1$ be integers, $R = \mathbb{Z}[x]/\langle x^d + 1\rangle$ with a power-of-2 degree d, and $q > p \geq 2$ be integers. All these are functions of the security parameter λ. An m-linearly-homomorphic encoding scheme with the message space R/pR and the codeword space $C \subseteq (R/qR)^n$ is a tuple of algorithms $\Pi_{\mathsf{Ecd}} = (\mathsf{Gen}, \mathsf{Encode}, \mathsf{Eval})$ with the following specifications.*

- $\mathsf{Gen}(1^\lambda) \mapsto (pp, sk)$*: Given the security parameter λ, it outputs public parameters pp and a secret key sk.*

- $\mathsf{Encode}(sk, a) \mapsto \mathbf{cw}$: *Given the secret key* sk *and a ring element* $a \in R/pR$*, it outputs a codeword* $\mathbf{cw} \in C$ *with the following property: the subsets* $\{C_a \mid a \in R/pR\}$ *partition* C *where* C_a *is the set of all possible encodings of* a*.*

- $\mathsf{Eval}(pp, \{\mathbf{cw}_1, \ldots, \mathbf{cw}_m\}, \{c_1, \ldots, c_m\}) \mapsto \mathbf{cw}^*$: *Given the public parameters* pp*,* m *codewords* $\{\mathbf{cw}_1, \ldots, \mathbf{cw}_m\}$*, and* m *scalars* $\{c_1, \ldots, c_m\}$ *in* R/pR*, it outputs a codeword* $\mathbf{cw}^*$*.*

Moreover, Algorithm Eval *satisfies the following property:*

- m*-linearly homomorphism: For all* $a_1, \ldots, a_m, c_1, \ldots, c_m \in (R/pR)^m$*, it holds that*

$$\mathbb{P}\left[\mathbf{cw}^* \in C_{\langle \mathbf{a}, \mathbf{c} \rangle} \;\middle|\; \begin{array}{l} (pp, sk) \leftarrow \mathsf{Gen}(1^\lambda) \\ \mathbf{cw}_i \leftarrow \mathsf{Encode}(sk, a_i) \\ \mathbf{cw}^* \leftarrow \mathsf{Eval}(pp, \{\mathbf{cw}_i\}_i, \{c_i\}_i) \end{array}\right] = 1 - \mathsf{negl}(\lambda) \ .$$

Algorithm Eval operates within the same framework as Algorithm Add of Definition 22. Moreover, the encodings are such that the codewords are computationally indistinguishable from random elements in the codeword space.

The m-power knowledge of exponent assumption (m-PKE) is a generalization of the knowledge of exponent assumption by [Dam91] to encoding schemes. We adapt this assumption to the quantum setting.

Definition 25 (m-PKE Against Quantum Adversaries)**.** *An encoding scheme* $\Pi_{\mathsf{Ecd}} = (\mathsf{Gen}, \mathsf{Encode}, \mathsf{Eval})$ *satisfies* m*-PKE assumption for the auxiliary input generator* $\mathcal{Z}$ *if for all QPT algorithms* $\mathcal{A}$*, there exists a valid QPT extractor* $\mathcal{E}$ *such that*

$$\mathbb{P}\left(\mathsf{ExptKnowledgeExt}_{\Pi_{\mathsf{Ecd}}, \mathcal{A}, \mathcal{Z}, \mathcal{E}, k}(1^\lambda) = 1\right) = \mathsf{negl}(\lambda) \ ,$$

where the experiment $\mathsf{ExptLinearExt}_{\Pi_{\mathsf{Ecd}}, \mathcal{A}, \mathcal{Z}, \mathcal{E}, k}(1^\lambda)$ *is defined as follows.*

1. *The challenger samples the public parameters and the secret key* $(pp, sk) \leftarrow \mathsf{Gen}(1^\lambda)$*, together with* α *and* s *sampled uniformly from* $(R/pR)^\times$ *and a fixed subset of* $(R/pR)^\times$*, respectively. It computes* σ *as follows:*

$$\begin{aligned} \sigma := \big(pp, &\mathsf{Encode}(sk, 1), \mathsf{Encode}(sk, s), \ldots, \mathsf{Encode}(sk, s^m), \\ &\mathsf{Encode}(sk, \alpha), \mathsf{Encode}(sk, \alpha s), \ldots, \mathsf{Encode}(sk, \alpha s^m)\big) \ . \end{aligned}$$

   It also computes $z \leftarrow \mathcal{Z}(\sigma)$*.*

2. *Then it runs the extraction process with the outputs, as follows:*

$$\big((\mathbf{cw}, \mathbf{cw}'), (a_0, \ldots, a_m)\big) \leftarrow \langle \mathcal{A}, \mathcal{E} \rangle (1^\lambda, |\sigma, z\rangle, |0\rangle) \ .$$

   The output of the experiment is 1 *if* $\mathbf{cw}' - \alpha \mathbf{cw} \in C_0$ *and* $\mathbf{cw} \notin C_S$ *where* $S = \sum_{i=0}^m a_i s^i$*. Otherwise, the output of the experiment is* 0*.*

In [GMNO18, NYI+20, GNSV23], it is assumed that $\mathcal{Z}$ is "benign", in the sense that the auxiliary information z is generated with a dependency on sk, s and α that is limited to the extent that it can be generated efficiently from σ. The extractor is also given the randomness of the adversary. In the quantum setting, we allow the extractor to have auxiliary inputs of the above type, while we omit the randomness of the adversary since it can be simulated by Hadamard gates and projective measurements.

In [GMNO18, NYI+20], the authors use LWE symmetric encryption (i.e., with $d = 1$) for the encoding scheme. The value of m is of order 2^{15}, which is significantly larger than the rank of the ciphertext space n (chosen around 2^{10}). The ciphertext modulus q can be as large as 736 bits, whereas the plaintext modulus p has 32 bits. The authors of [GNSV23] rely on high-degree MLWE, whereas the (module) rank of their ciphertext space is constant.

In Definition 25, the adversary is given $2(m+1)$ encodings of the powers of s. We use them to define the following matrix:

$$\mathbf{C} := \left(\begin{array}{c|c|c|c} \mathsf{Encode}(sk, 1) & \mathsf{Encode}(sk, s) & \cdots & \mathsf{Encode}(sk, s^m) \\ \mathsf{Encode}(sk, \alpha) & \mathsf{Encode}(sk, \alpha s) & & \mathsf{Encode}(sk, \alpha s^m) \end{array} \right) \in (R/qR)^{2n \times (m+1)} .$$

A small combination $\mathbf{Ce}$ of the columns gives a pair of ciphertext $(\mathbf{cw}, \mathbf{cw}')$ that satisfies $\mathbf{cw} - \alpha\mathbf{cw}' \in C_0$, by the $(m+1)$-linear homomorphism property of the scheme. The vectors $\mathbf{cw}$ and $\mathbf{cw}'$ respectively correspond to the first and second halves of $\mathbf{Ce}$. Note that the auxiliary input z does not help to recover $\mathbf{e}$. It contains codewords of the form $\mathsf{Encode}(sk, \beta v(s))$ where v is a publicly known polynomial and β is a uniformly sampled element from R/pR that is independent from all other parameters. This does not help the adversary to extract information about the matrix $\mathbf{C}$, as can be shown using a hybrid argument in which one replaces the plaintext $\beta v(s)$ with a garbage plaintext (using the fact that the codewords are indistinguishable from uniform). By adapting the arguments from Subsection 2.6.3, it can be seen that sampling $\mathbf{Ce}$ obliviously allows to break the security assumption of Definition 25.

Chapter 3

Analysis of Fiat-Shamir with Aborts

The results of this chapter are based on the collaboration of the author with Julien Devevey, Alain Passelègue, Damien Stehlé, and Keita Xagawa. The following article is related to this chapter.

[DFPS23] A Detailed Analysis of Fiat-Shamir with Aborts, with Julien Devevey, Alain Passelègue, and Damien Stehlé. In *CRYPTO 2023*.

The first set of results concerns the correctness and the runtime in Section 3.2.

The second set of results relates to the security analyses of FSwA. We provide two security analyses for FSwBA in the QROM: the history-free approach and the adaptive-reprogramming approach which are respectively detailed in section 3.3 and 3.4. We also provide a ROM analysis based on a classical version of the adaptive reprogramming approach in Section 3.4. The security analysis for FSwUA is presented in Section 3.5.

Finally, the analysis of the Σ-protocols whose simulator's quality is measured in terms of the Rényi divergence (rather than the statistical distance) is detailed in Section 3.6.

3.1 Preliminaries

We use code-based games to write the proofs. We use capital letters with fraktur font (e.g., $\mathfrak{L}$) to denote the list of objects. We let $\mathsf{Coll} : \mathfrak{L} \mapsto \{0,1\}$ be the function that takes as input a list and outputs 1 if and only if at least two of the elements of the list are equal. We sometimes abuse the notation and let $\mathsf{Coll}(\mathfrak{L})$ denote the event that it returns 1. To denote that a function f (or a database) is reprogrammed at input x to the value y we use the notation $f^{x \mapsto y}$.

3.1.1 Probabilities

Let X be a random variable over some finite space Ω. The min-entropy of X is

$$H_\infty(X) := -\log \max_{\omega \in \Omega} \mathbb{P}_X[\omega].$$

We recall an upper bound on the collision probability of i.i.d. random variables.

Lemma 26. *Let $\mathfrak{L}$ be a list of i.i.d. random variables $\{X_i\}_i$ over a finite set, each of which has min-entropy α. We have*

$$\mathbb{P}[\mathsf{Coll}(\mathfrak{L})] \leq |\mathfrak{L}|^2 \cdot 2^{-\alpha-1}.$$

Proof. Let ℓ denote the size of $\mathfrak{L}$. We bound this probability recursively:

$$\begin{aligned}
\mathbb{P}[\mathsf{Coll}(\mathfrak{L}) = 1] &= \mathbb{P}[\mathsf{Coll}(\{w_i\}_{i\in[\ell]}) = 1] \\
&\leq \mathbb{P}[\mathsf{Coll}(\{w_i\}_{i\in[\ell-1]}) = 1] \\
&\quad + \mathbb{P}[\mathsf{Coll}(\{w_i\}_{i\in[\ell-1]}) = 0 \wedge \mathsf{Coll}(\{w_i\}_{i\in[\ell]}) = 1] \\
&= \mathbb{P}[\mathsf{Coll}(\{w_i\}_{i\in[\ell-1]}) = 1] + (\ell-1)\cdot 2^{-\alpha} \\
&\qquad\qquad \vdots \\
&\leq (\ell-1)\cdot 2^{-\alpha} + (\ell-2)\cdot 2^{-\alpha} + \cdots + 2^{-\alpha} \\
&\leq |\ell|^2 \cdot 2^{-\alpha-1}.
\end{aligned}$$

□

Assuming now that $\mathsf{Supp}(X) \subseteq \mathsf{Supp}(Y)$, the Rényi divergence of infinite order is defined as follows:

$$R_\infty(X\|Y) := \max_{x\in\mathsf{Supp}(X)} \frac{\mathbb{P}_X(x)}{\mathbb{P}_Y(x)}.$$

We will use the same notations if X, Y are probability distributions. In the following, for the sake of simplicity, we restrict ourselves to discrete distributions. The definition above and our results involving the Rényi divergence carry over to continuous ones. The same holds for their applicability to Lyubashevsky's signature, as argued in [DFPS22]. Some background on the Rényi divergence are reminded below.

The following lemma borrowed from [LSS14] lists a few properties of the Rényi divergence. Proofs can be found in [vEH14].

Lemma 27. *Let P and Q be two discrete probability distributions such that we have $\mathsf{Supp}(P) \subseteq \mathsf{Supp}(Q)$. The following properties hold.*

- ***Log. Positivity:*** *$R_\infty(P\|Q) \geq R_\infty(P\|P) = 1$.*
- ***Data Processing Inequality:*** *$R_\infty(P^f\|Q^f) \leq R_\infty(P\|Q)$ for any probabilistic function f, where X^f denotes the distribution of $f(x)$ where $x \hookleftarrow X$.*
- ***Multiplicativity:*** *Let P and Q be two distributions of a pair of random variables X_1 and X_2 and P_i and Q_i denote the marginal distribution of X_i under P and Q, respectively. We have*

$$R_\infty(P\|Q) \leq R_\infty(P_1\|Q_1) \cdot \max_{x_1\in\mathsf{Supp}(P_1)} R_\infty((P_2|x_1)\|(Q_2|x_1)).$$

- ***Probability Preservation:*** *Let $E \subseteq \mathsf{Supp}(Q)$ be an arbitrary event. Then we have*

$$P(E) \leq Q(E)\cdot R_\infty(P\|Q).$$

3.1.2 Σ-protocols

We start by recalling various definitions pertaining to Σ-protocols.

Definition 26 (Σ-Protocol with Aborts)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ 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 $c \in \mathcal{C}$;*
- $\mathsf{P}_2 : (x, y, w, c, st) \to z$ *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. 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 a Σ-protocol.

Definition 27 (Correctness)**.** *Let $\gamma, \beta \geq 0$. A Σ-protocol $((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is (γ, β)-correct if for every $x \in \mathcal{L}$ and valid witness $y \in \mathcal{Y}$ the following holds.*

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

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

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

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

We also let β denote *the probability of aborting*. 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 will be pushed toward 0, whereas γ will stay close to 1.

We consider the following statistical Honest-Verifier Zero-Knowledge (HVZK) definition, which benefits from a simulator even for aborting transcripts of the Σ-protocol.

Definition 28 (Statistical Honest-Verifier Zero-Knowledge (HVZK)). *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 $z = \perp$.*

If Σ 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. n the rest of the paper, we limit ourselves to public-coin Σ-protocols.

Note that the zero-knowledge definition that is usually used in the literature of Fiat-Shamir with aborts is the one that only concerns the non-aborting transcripts which is a weaker requirement on the Σ-protocol. However, it is shown in [DFPS23] that Lyubashevsky's Σ-protocols, one of the principal applications of FSwA transform and the main concern of our analyses, satisfy this stronger notion.

We also consider a computational zero-knowledge definition. For having the equivalency between the one-transcript vs many-transcript variants, 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.

Definition 29 (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 advantage is $\leq \varepsilon_{zk}$:*

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

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

Note that in all the analyses we consider in this work, when we use the zero-knowledge property, the witness y is available to the challenger. 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.

For cryptographic purposes, one instantiates the Σ-protocol with hard samples. This notion is captured in the following definition.

Definition 30 (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* Gen.

A useful statistical property of a Σ-protocol is the min-entropy of the commitments. We borrow the following definition from [KLS18].

Definition 31 (Commitment Min-Entropy). *For $\alpha \geq 0$, we say that an identification scheme $((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ with instance generator* Gen *has commitment min-entropy α if $H_\infty[w|(w, st) \leftarrow \mathsf{P}_1(x, y)] \geq \alpha$, for all $(x, y) \leftarrow \mathsf{Gen}(1^\lambda)$.*

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

An identification protocol is said to be unique response if for every w, c there exists at most one response z such that the transcript (w, c, z) passes the verification. The following definition relaxes this notion against computationally bounded adversaries.

Definition 32 (Computational Unique Response)**.** *Let $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ be an identification scheme with instance generator Gen. For any quantum adversary $\mathcal{A}$, we define the following advantage function:*

$$\mathsf{Adv}^{\mathsf{CUR}}_{\Sigma}(\mathcal{A}) = \mathbb{P}_{(x,y)\leftarrow\mathsf{Gen}(1^\lambda)}\left[\begin{array}{c|c} z \neq z' & \\ \mathsf{V}_2(w,c,z)=1 & (w,c,z,z') \leftarrow \mathcal{A}(x) \\ \mathsf{V}_2(w,c,z')=1 & \end{array}\right].$$

3.1.3 Signatures

Here we briefly recall the formalism of digital signatures.

Definition 33 (Digital Signature)**.** *A signature scheme is a tuple of PPT algorithms $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ with the following specifications:*

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

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

where the probability is taken over the random coins of the signing algorithm. We say that it is correct in the (Q)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 remind the definition of existential unforgeability against chosen message attacks ($\mathsf{UF\text{-}CMA}$).

Definition 34 (Security)**.** *Let $T, \delta \geq 0$. A signature scheme $\mathsf{SIG} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is said to be (T, δ)-$\mathsf{UF\text{-}CMA}$ secure in the ROM if for any quantum adversary $\mathcal{A}$ with runtime $\leq T$ given (classical) access to the signing oracle and (quantum) access to a random oracle H, it holds that*

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

where the randomness is also taken over the random coins of $\mathcal{A}$. The adversary should also not have issued a sign query for μ^*. 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.

$\underline{\mathsf{KeyGen}(1^\lambda)}$:
1: $(x, y) \leftarrow \mathsf{Gen}(1^\lambda)$
2: $(vk, sk) = (x, (x, y))$
3: **return** (vk, sk)

$\underline{\mathsf{Sign}(sk, \mu)}$:
1: $\kappa := 1$
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: **if** $z = \perp$ **return** $\perp$
8: **return** $\sigma = (w, z)$

$\underline{\mathsf{Ver}(vk, \mu, \sigma)}$:
1: Parse $\sigma = (w, z)$
2: $c = H(w \| \mu)$
3: **return** $\mathsf{V}_2(vk, w, c, z)$

Figure 3.1: Signatures $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ and $\mathsf{SIG}_\infty = \mathsf{FS}_\infty[\Sigma, H]$. The signature SIG_B uses blocks highlighted with the blue color, whereas SIG_∞ does not.

*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* UF-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* sUF-CMA$_1$ *by taking the conjunction of* sUF-CMA *and* UF-CMA$_1$*.*

Note that for deterministic signatures, the UF-CMA$_1$ and UF-CMA security notions coincide.

3.1.4 Fiat-Shamir Transform

Let $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ be an identification protocol with an instance generator Gen 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 Σ. Then, for every positive integer B, one can construct a signature scheme $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ by applying the Fiat-Shamir transform with bounded aborts (FSwBA) as in Figure 3.1. 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 $\mathsf{SIG}_\infty = \mathsf{FS}_\infty[\Sigma, H]$ of the Fiat-Shamir transform for a Σ-protocol Σ as in Figure 3.1. Note that the signing algorithm of SIG_∞ may not be PPT as required in Definition 33. Ideally, it would still be expected polynomial-time.

In this work we show that sUF-CMA security (and sometimes sUF-CMA$_1$) of such signatures can be reduced to their UF-NMA security. Here, we briefly recall two possible ways to reduce UF-NMA security to the security of the underlying Σ-protocol. For more details, we refer the reader to prior works (e.g., [Lyu09, Lyu12, AFLT16, DFMS19, LZ19]).

- In [AFLT16, KLS18], the authors consider *lossy identification schemes* in which there exists another instance generator function $\mathsf{Gen}_{\mathsf{ls}}$ for the protocol that only outputs an instance x_{ls} without any witness. Moreover, its output distribution is computationally indistinguishable from the one of the real instance generator Gen. Further, it is said to be $\varepsilon_{\mathsf{ls}}$-sound if no cheating prover (even unbounded) can impersonate the real prover given x_{ls} as input and make the verifier to accept with probability

more than $\varepsilon_{\mathsf{ls}}$. They reduce UF-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 Gen and $\mathsf{Gen}_{\mathsf{ls}}$.

- In [DFMS19, LZ19] 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 Σ-protocol. Their reduction is less tight than the one of [KLS18].

3.1.5 Quantum computations

A quantum state $|\psi\rangle$ of a system is a unit vector in the Hilbert space $\mathbb{C}^d$. Each step of a quantum algorithm is either a unitary transformation or a quantum measurement over the states. A unitary transformation over the space $\mathbb{C}^d$ is a $d \times d$ matrix $\mathbf{U}$ such that $\mathbf{U}\mathbf{U}^* = \mathbf{I}_d$ where $\mathbf{U}^*$ is the conjugate-transpose of $\mathbf{U}$. Let $\{|b_i\rangle\}_{i\in[d]}$ be an orthonormal basis for $\mathbb{C}^d$. Measuring a state $|\psi\rangle$ with this basis returns a value i with probability $|\langle b_i|\psi\rangle|^2$, and the post-measurement state is $|b_i\rangle$.

Let $f : \{0,1\}^n \to \{0,1\}^m$ be an arbitrary function. Then the quantum oracle $|f(\cdot)\rangle$ is a unitary transformation, acting on the computational basis $\{|x\rangle|y\rangle : x \in \{0,1\}^n, y \in \{0,1\}^m\}$ as $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$, and extended by linearity. An oracle-aided quantum machine $\mathcal{A}^{\mathcal{O}}$ is allowed to use an oracle $\mathcal{O}$ as a black box by querying $\mathcal{O}$ in some quantum state.

Consider a scenario in which we have an array (a data structure) of N classical strings $x_1, \ldots, x_N$. Quantum Random Access Classical Memory (QRACM) allows us to load this data to a quantum register in superposition. More precisely, a QRACM operation is defined by

$$U_{QRACM} : |i\rangle|y_i\rangle \mapsto |i\rangle|y_i \oplus x_i\rangle$$

for all i and y_i, and is extended by linearity. The efficiency of quantum random access gates has been a point of debate [JS19]. In this work, we single out the results using QRACM because of its difference from the quantum circuit model.

For more details on quantum computations, we refer the reader to [NC11].

We need the following lemmas for the history-free approach. The first one is the one-sided O2H lemma.

Lemma 28 (One-Sided O2H [AHU19, Theorem 3], adapted)**.** *Let X, Y, S be three finite sets with $S \subseteq X$. Let $H, G : X \to Y$ be two functions such that $H(x) \neq G(x)$ if and only if $x \in S$. Let $\mathcal{A}$ be a quantum algorithm that distinguishes quantum oracles $|G\rangle$ and $|H\rangle$ with q queries and success probability $\varepsilon_{\mathcal{A}}$. Then, there exists a quantum algorithm $\mathcal{B}$ that, given access to the oracle $|H\rangle$ and $\mathcal{A}$, finds an element in S with success probability $\geq \varepsilon_{\mathcal{A}}^2/(4q^2)$.*

The next lemma links two notions of indistinguishability.

Lemma 29 (Oracle-Indistinguishability [Zha12a, Theorem 1.1])**.** *Let D_1 and D_2 be efficiently samplable distributions with supports contained in a finite set Y. Let X be an arbitrary finite set. Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be two functions from X to Y such that, on each*

Game $\mathsf{Reprogram}_b$:		$\underline{\mathsf{Reprogram}(x_2)}$:
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: $b' \leftarrow \mathcal{A}^{\lvert H_b\rangle,\ \mathsf{Reprogram}(\cdot)}$	▷ *Quantum*	3: $H_1 := H_1^{(x_1,x_2)\mapsto y}$
4: $b' \leftarrow \mathcal{A}^{H_b,\ \mathsf{Reprogram}(\cdot)}$	▷ *Classical*	4: **return** (x_1, x')
5: **return** b'		

Figure 3.2: The reprogramming game.

input $x \in X$, they output an independent sample in Y from D_1 and D_2, respectively. Let $\mathcal{A}$ be a quantum adversary that distinguishes two quantum oracles $|\mathcal{O}_1\rangle$ and $|\mathcal{O}_2\rangle$ with advantage ε by making q quantum queries. Then there exists a quantum algorithm $\mathcal{B}$ that distinguishes D_1 and D_2 with advantage $\geq (6q)^{-3}\varepsilon^2$.

The following lemma gives an upper bound on succeeding in a generic search game.

Lemma 30 (Adapted from [AHU19, Lem. 2])**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be two sets, and $H : \mathcal{X} \to \mathcal{Y}$ be a random function drawn from a distribution such that $\mathbb{P}[H(x) = 1] \leq \lambda$ for some fixed $\lambda \in [0, 1]$. Let $\mathcal{A}$ be an (unbounded) adversary making q quantum queries to H. Then it holds that $\mathbb{P}[H(x) = 1 | x \leftarrow \mathcal{A}^{|H\rangle}] \leq 4(q + 2)(q + 1)\lambda$.*

3.1.6 Adaptive Reprogramming in the QROM

We rely on the following lemma for one of our analyses in the QROM. Consider the following decision game: Assume the hash function takes inputs of the form (x_1, x_2), and an adversary (with quantum access to the hash function) has access to a reprogramming oracle which can be queried with any value x_2. On a query x_2, the oracle samples a value x_1 and either leaves the hash function unchanged or reprograms it on input (x_1, x_2) to a uniformly random value y from its range. It may also maintain a state x'. Given (x_1, x'), the adversary's goal is to decide whether the oracle reprograms the hash function or not. The following lemma proves this game to be hard even for quantum adversaries.

Lemma 31 (Adaptive Reprogramming [GHHM21, Proposition 2])**.** *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 3.2 and making q quantum queries to the random oracle and r classical queries to the* Reprogram *function. Then*

$$\left|\mathbb{P}[1 \Leftarrow \mathsf{Reprogram}_0^{\mathcal{A}}] - \mathbb{P}[1 \Leftarrow \mathsf{Reprogram}_1^{\mathcal{A}}]\right| \leq \frac{3r}{2}\sqrt{q \cdot 2^{-\alpha}},$$

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

We finally state the classical variant of Lemma 31.

Lemma 32 (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 3.2 and making q classical queries to the random oracle and r classical queries to the* Reprogram *function. Then*

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

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. □

3.2 Runtime of FSwUA and Correctness of FSwBA

In [DFPS23], a natural construction of Σ-protocol and an instantiation of a hash function are exhibited, such that in the corresponding signature obtained by the FSwA transform, the singning algorithm never halts.

Theorem 7. *Let H be a random oracle. There exists a public-coin identification protocol Σ with instance generator* Gen *that is* $(1, 1/100)$*-correct,* sc-HVZK *under the hardness of* LWE *problem , its final verification is deterministic, and has the following property: with overwhelming probability over the randomness of* Gen*, for every message μ, the expected runtime of the signing algorithm of* $\mathsf{SIG}_\infty := \mathsf{FS}_\infty[\Sigma, H]$*, over the randomness of H, is infinite.*

The above result puts forward an anomaly in the runtime (and correctness) of the corresponding signature in the random oracle model, implying that the definitions must be updated. In Subsection 3.2.1, we propose minor updates to the signature definition so that it supports such pathological behaviour. 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 Subsection 3.2.2, we will prove FSwUA yields signatures whose runtime satisfies the updated definition. Correctness of FSwBA is also addressed in Section 3.2.2 as a corollary.

3.2.1 Updated signature definition

As mentioned previously, 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 construction pertaining to Theorem 7, 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 33 and we only highlight the differences.

Definition 35 (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 33, except for the runtime of* Sign*, which we define below, and a minor tweak in the security game.*

- $\mathsf{Sign}^H : (sk, \mu) \to \sigma$ *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 $T_{\mathsf{Sign}^H(sk,\mu)}$ the runtime of* $\mathsf{Sign}(sk, \mu)$*.*

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 (vk, sk) in the range of KeyGen *and μ,*

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

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

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

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'-$\mathsf{UF\text{-}CMA}$ *security exactly as* $\mathsf{UF\text{-}CMA}$ *security in Definition 34, 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.*

3.2.2 Runtime and Correctness

Definition 35 does not forbid the situation described in the beginning of Subsection 3.2 from occurring but guarantees that it should be hard to find non-halting queries.

Theorem 8 (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:*

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

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 $N = I_{\mathsf{Sign}^H(sk,\mu)}$. 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:

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

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

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

Let us first focus on $\mathbb{P}_H(M = k)$. The random variable M only depends on the w_i's, which are i.i.d.: we can bound the collision probability using Lemma 26. Hence $\mathbb{P}_H(M = k) \leq k^2 \cdot 2^{-\alpha-1}$. 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

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

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 $\mathbb{P}_H(N > i | N < M) = \beta^i$. Plugging everything together, we obtain

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

□

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 9. *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 $(sk, vk) \leftarrow \mathsf{KeyGen}$ and $\mu \in \mathcal{M}$. Conditioned on $\mathsf{Sign}(sk, \mu)$ 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 10. *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 $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and any message $\mu \in \mathcal{M}$, we have*

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

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

Proof. The result follows from Theorem 8. 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

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

□

3.3 Security of FSwBA: the History-free Approach

In this section we discuss the security of the Fiat-Shamir transform with bounded aborts. We first prove the $\mathsf{UF\text{-}CMA}_1$ security of the signature in the QROM based on the (flawed) proof in [KLS18]. Subsection 3.3.2 is devoted to extending the latter to obtain strong unforgeability. Finally, we extend the results from $\mathsf{UF\text{-}CMA}_1$ and $\mathsf{sUF\text{-}CMA}_1$ security to $\mathsf{UF\text{-}CMA}$ and $\mathsf{sUF\text{-}CMA}$, respectively.

3.3.1 $\mathsf{UF\text{-}CMA}_1$ Security of FSwBA

Below, we reduce the $\mathsf{UF\text{-}CMA}_1$ security to its $\mathsf{UF\text{-}NMA}$ security using the statistical zero-knowledge property of the Σ-protocol. One can see this proof as a correction of [KLS18]. We claim that the same approach applies to $\mathsf{UF\text{-}CMA}$ security in Section 3.3.3.

Theorem 11. *Let $\varepsilon_{zk}, \alpha, T_{\mathsf{Sim}} \geq 0$, $B \geq 0$, H and G hash functions modeled as random oracles. Assume that $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is an $(\varepsilon_{zk}, T_{\mathsf{Sim}})$-HVZK public-coin identification protocol and that the commitment message of the prover has min-entropy α. For any quantum adversary $\mathcal{A}$ against $\mathsf{UF\text{-}CMA}_1$ security of $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ that issues at most Q_H quantum queries to the random oracle H and Q_S classical queries to the signing oracle, there exists a quantum adversary $\mathcal{B}$ against $\mathsf{UF\text{-}NMA}$ security of SIG_B with*

$$\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + T_{\mathsf{Sim}} \cdot B \cdot (Q_S + Q_H) \ ,$$

and such that

$$\mathsf{Adv}^{\mathsf{UF\text{-}CMA}_1}_{\mathsf{SIG}_B}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) + 2^{\frac{-\alpha+3}{2}} \cdot B \cdot (Q_S + Q_H) + 30\sqrt{\varepsilon_{zk} \cdot B} \cdot (Q_S + Q_H)^{\frac{3}{2}} \ .$$

The reduction holds in the QROM and relies on $\mathcal{B}$ having access to a private random oracle H' with the same domain and range as H that is not accessible by $\mathcal{A}$.

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

Note that one could adjust the proof of the above statement (as well as those of the next statements) to replace access to the private random oracle by relying on a quantum pseudo-random function in the reduction [Zha12a].

Proof. The proof of Theorem 11 is based on a sequence of hybrid games. Recall that we assumed the reduction has access to another random oracle H' to which the adversary does not have access to, which serves to simulate the random oracle.

Game G_0. This is the genuine $\mathsf{UF\text{-}CMA}_1$ game, as described in Figure 3.3.

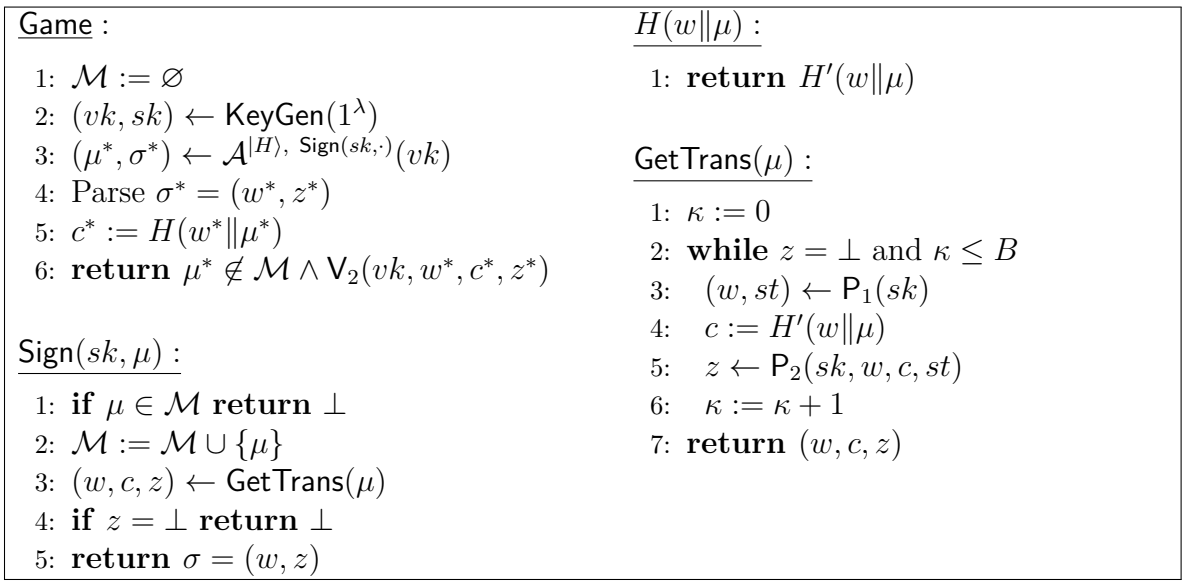


Figure 3.3: Game G_0

Game G_1. In this game, described in Figure 3.4, we record all the transcripts produced during GetTrans and return them as its output. The function Sign runs GetTrans on its input μ. Hence, we modify it to single out the last transcript of the recording and continue with it as before. Nothing else changes in this game. This change is only internal to the oracles and the adversary's view remains identical to that of G_0.

$\mathsf{Sign}(sk, \mu)$:	$\mathsf{GetTrans}(\mu)$:
1: **if** $\mu \in \mathcal{M}$ **return** $\perp$	1: $\kappa := 1,\ z^{(0)} := \perp$
2: $\mathcal{M} := \mathcal{M} \cup \{\mu\}$	2: **while** $z^{(\kappa-1)} = \perp$ and $\kappa \leq B$
3: $\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[\kappa]} \leftarrow \mathsf{GetTrans}(\mu)$	3: $(w^{(\kappa)}, st^{(\kappa)}) \leftarrow \mathsf{P}_1(sk)$
4: **if** $z^{(\kappa)} = \perp$ **return** $\perp$	4: $c^{(\kappa)} := H'(w^{(\kappa)} \| \mu)$
5: **return** $\sigma = (w^{(\kappa)}, z^{(\kappa)})$	5: $z^{(\kappa)} \leftarrow \mathsf{P}_2(sk, w^{(\kappa)}, c^{(\kappa)}, st^{(\kappa)})$
	6: $\kappa := \kappa + 1$
	7: **return** $\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[\kappa]}$

Figure 3.4: Game G_1

Game G_2. Its only difference with Game G_1 is that we replace the randomness of the prover in GetTrans with a uniform function $RF : \{0,1\} \times \mathcal{M} \times [B] \to \mathcal{R}$ which is hidden from adversary's view, to derandomize GetTrans. Note that it depends on the message μ and number of the round in the rejection sampling to ensure uniqueness of the random coin with respect to them. It only changes the GetTrans subroutine. Further, the function GetTrans becomes a deterministic function with respect to the message μ. We use subscripts to emphasize this fact in Figure 3.5. Although the signatures become deterministic, since we are only interested in $\mathsf{UF\text{-}CMA}_1$ security, the adversary's view remains unchanged. The changes are depicted in Figure 3.5.

$\mathsf{GetTrans}(\mu)$:
1: $\kappa := 1,\ z_\mu^{(0)} := \perp$
2: **while** $z_\mu^{(\kappa-1)} = \perp$ and $\kappa \leq B$
3: $(w_\mu^{(\kappa)}, st_\mu^{(\kappa)}) := \mathsf{P}_1(sk; RF(0\|\mu\|\kappa))$
4: $c_\mu^{(\kappa)} := H'(w_\mu^{(\kappa)} \| \mu)$
5: $z_\mu^{(\kappa)} := \mathsf{P}_2(sk, w_\mu^{(\kappa)}, c_\mu^{(\kappa)}, st_\mu^{(\kappa)}; RF(1\|\mu\|\kappa))$
6: $\kappa = \kappa + 1$
7: **return** $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]}$

Figure 3.5: Game G_2

Game G_3. In this game, described in Figure 3.6, we change the way that the random oracle queries are answered. Upon receiving an input $w\|\mu$, the oracle H queries the GetTrans function on input μ to receive a sequence of transcripts. Then if w is equal to one of the commitments in the transcripts, it returns its corresponding challenge. This is just a syntactic change and the adversary's view remains identical. The modifications can be seen in Figure 3.6.

Game G_4. Let $\mathfrak{L}_\mu$ be the list of commitments generated for the message μ in the $\mathsf{GetTrans}(\mu)$ function. In this game, we modify $\mathsf{GetTrans}(\mu)$ such that if $\mathsf{Coll}(\mathfrak{L}_\mu)$ occurs, then it returns

$H(w\|\mu)$:

1: $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]} := \mathsf{GetTrans}(\mu)$
2: **if** $\exists i : w = w_\mu^{(i)}$ **return** $c_\mu^{(i)}$
3: **return** $H'(w\|\mu)$

Figure 3.6: Game G_3

a special symbol Υ. We also change both Sign and H to return Υ if their call to GetTrans returns Υ. All these changes are reflected in Figure 3.7.

$\mathsf{Sign}(sk, \mu)$:

1: **if** $\mu \in \mathcal{M}$ **return** $\perp$
2: $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
3: **if** $\mathsf{GetTrans}(\mu) = \Upsilon$ **return** Υ
4: $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]} := \mathsf{GetTrans}(\mu)$
5: **if** $z_\mu^{(\kappa)} = \perp$ **return** $\perp$
6: **return** $\sigma_\mu = (w_\mu^{(\kappa)}, z_\mu^{(\kappa)})$

$H(w\|\mu)$:

1: **if** $\mathsf{GetTrans}(\mu) = \Upsilon$ **return** Υ
2: $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]} := \mathsf{GetTrans}(\mu)$
3: **if** $\exists i : w = w_\mu^{(i)}$ **return** $c_\mu^{(i)}$
4: **return** $H'(w\|\mu)$

$\mathsf{GetTrans}(\mu)$:

1: $\kappa := 1$, $z_\mu^{(0)} := \perp$
2: **while** $z_\mu^{(\kappa-1)} = \perp$ and $\kappa \le B$
3: $\quad (w_\mu^{(\kappa)}, st_\mu^{(\kappa)}) := \mathsf{P}_1(sk; RF(0\|\mu\|\kappa))$
4: $\quad c_\mu^{(\kappa)} := H'(w_\mu^{(\kappa)}\|\mu)$
5: $\quad z_\mu^{(\kappa)} :=$
$\qquad \mathsf{P}_2(sk, w_\mu^{(\kappa)}, c_\mu^{(\kappa)}, st_\mu^{(\kappa)}; RF(1\|\mu\|\kappa))$
6: $\quad \kappa := \kappa + 1$
7: $\mathfrak{L}_\mu := \{w_\mu^{(i)}\}_{i\in[\kappa]}$
8: **if** $\mathsf{Coll}(\mathfrak{L}_\mu)$ **return** Υ
9: **return** $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]}$

Figure 3.7: Game G_4. The differences from Game G_3 are depicted in blue.

Let C be the concatenation of two functions $H\|\mathsf{GetTrans}$ that sends $w\|\mu$ to the bit-string $H(w\|\mu)\|\mathsf{GetTrans}(\mu)$. The queries of the adversary (both sign queries and random oracle queries) can be answered by using quantum queries to C. Therefore, without loss of generality, we assume that the adversary makes $Q_S + Q_H$ quantum queries directly to the concatenation function. Let C_3 and C_4 be the concatenation functions in Game G_3 and Game G_4, respectively. If an adversary distinguishes G_3 from G_4, one can construct a wrapper around $\mathcal{A}$ distinguishing C_3 from C_4 since all the queries in the game can be simulated by the concatenation function as described above. They behave differently only on the inputs including a message that triggers Υ. Building on that, we use Lemma 28 to construct an algorithm $\mathcal{B}$ based on $\mathcal{A}$ which extracts a message μ triggering Υ as follows

$$\begin{aligned}&\left|\mathbb{P}[1 \leftarrow \mathcal{A}^{|\mathsf{C}_3\rangle}] - \mathbb{P}[1 \leftarrow \mathcal{A}^{|\mathsf{C}_4\rangle}]\right| \\ &\le 2(Q_S + Q_H)\sqrt{\mathbb{P}[\mu \text{ triggers } \Upsilon \mid \mu \leftarrow \mathcal{B}^{|\mathsf{C}_3\rangle}]}.\end{aligned}$$

Now, note that C_3 never outputs Υ. In fact, for every $w\|\mu$, the value of $\mathsf{C}_3(w\|\mu)$ is independent from $\mathsf{Coll}(\mathfrak{L}_\mu)$. Therefore, algorithm $\mathcal{B}$ can do nothing except a totally random guess. For each message μ, the probability of $\mathsf{Coll}(\mathfrak{L}_\mu)$ can be bounded by Lemma 26. Hence we have

$$\left|\mathbb{P}[1 \Leftarrow G_4^{\mathcal{A}}] - \mathbb{P}[1 \Leftarrow G_3^{\mathcal{A}}]\right| \le 2(Q_S + Q_H) \cdot B \cdot 2^{\frac{-\alpha-1}{2}}.$$

Game G_5. In this game, we let the challenges $c_\mu^{(i)}$'s in the GetTrans function be produced as in the Σ-protocol without using the random oracle and sampled from the uniform distribution. To make GetTrans deterministic, we use a uniform function $RF' : \mathcal{M} \times [B] \to \mathcal{C}$ as a function to sample the challenges. The domain $\mathcal{M} \times [B]$ of the function suffices for our purpose since within the UF-CMA$_1$ security the adversary is not allowed to query one message twice. Replacing the verifier V_1 with RF' is sufficient, since the identification protocol is public-coin. Note that both Sign and H change accordingly. Thanks to the Coll($\mathfrak{L}_\mu$) check, each invocation of H' can be treated independently. Thus, the distribution of GetTrans, and consequently those of Sign and H, remains identical to that of the previous game. In this game the rounds of the rejection sampling are finally independent and each one has the same distribution as the real transcript in the Σ-protocol. All these changes are reflected in Figure 3.8.

GetTrans(μ) :

1: $\kappa := 1$, $z_\mu^{(0)} := \bot$
2: **while** $z_\mu^{(\kappa-1)} = \bot$ and $\kappa \leq B$
3: $\quad (w_\mu^{(\kappa)}, st_\mu^{(\kappa)}) := \mathsf{P}_1(sk; RF(0\|\mu\|\kappa))$
4: $\quad c_\mu^{(\kappa)} := RF'(\mu\|\kappa)$
5: $\quad z_\mu^{(\kappa)} := \mathsf{P}_2(sk, w_\mu^{(\kappa)}, c_\mu^{(\kappa)}, st_\mu^{(\kappa)}; RF(1\|\mu\|\kappa))$
6: $\quad \kappa := \kappa + 1$
7: $\mathfrak{L}_\mu := \{w_\mu^{(i)}\}_{i\in[\kappa]}$
8: **if** Coll($\mathfrak{L}_\mu$) **return** Υ
9: **return** $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]}$

Figure 3.8: Game G_5. The difference from Game G_4 is depicted in blue.

Game G_6. In this game, we replace the transcripts with the simulated ones in each round of GetTrans. Let Sim be the zero-knowledge simulator of Σ. We use a new uniform function $RF'' : \mathcal{M} \times [B] \to \mathcal{R}$ as the randomness generator of Sim. Note that RF'' is not accessible by the adversary. Figure 3.9 updates GetTrans accordingly.

GetTrans(μ) :

1: $\kappa := 1$, $z_\mu^{(0)} := \bot$
2: **while** $z_\mu^{(\kappa-1)} = \bot$ and $\kappa \leq B$
3: $\quad c_\mu^{(\kappa)} := RF'(\mu\|\kappa)$
4: $\quad (w_\mu^{(\kappa)}, z_\mu^{(\kappa)}) := \mathsf{Sim}(vk, c_\mu^{(\kappa)}; RF''(\mu\|\kappa))$
5: $\quad \kappa := \kappa + 1$
6: $\mathfrak{L}_\mu := \{w_\mu^{(i)}\}_{i\in[\kappa]}$
7: **if** Coll($\mathfrak{L}_\mu$) **return** Υ
8: **return** $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]}$

Figure 3.9: Game G_6. The difference from Game G_5 is depicted in blue.

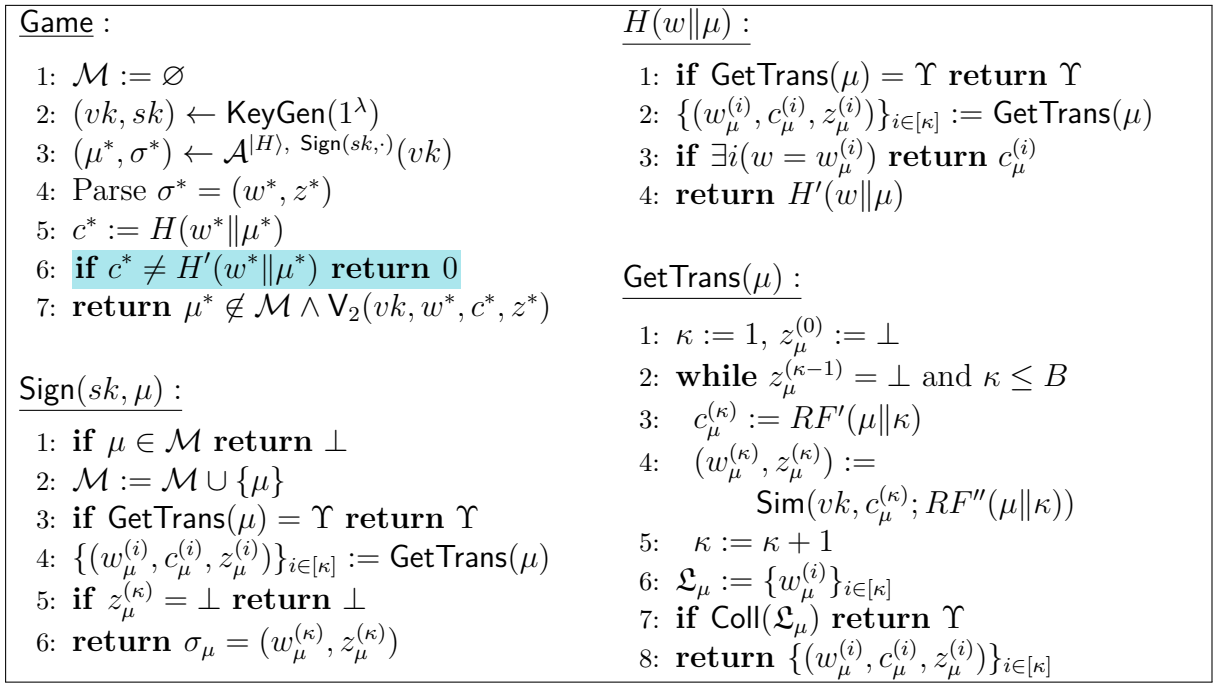
Let C_5 and C_6 be concatenation functions of $H\|$GetTrans in games G_5 and G_6. Without loss of generality, we allow the adversary to make $Q_S + Q_H$ direct quantum queries to

them and is tasked to distinguish C_5 and C_6. The distribution of the outcomes of C_5 and C_6 are statistically (or computationally in the case of sc-HVZK) $B \cdot \varepsilon_{zk}$-far from each other. Plugging C_5 and C_6 into Lemma 29 implies

$$\left|\mathbb{P}[1 \Leftarrow G_6^{\mathcal{A}}] - \mathbb{P}[1 \Leftarrow G_5^{\mathcal{A}}]\right| \leq (6Q_S + 6Q_H)^{\frac{3}{2}}\sqrt{B \cdot \varepsilon_{zk}}.$$

In the case of sc-HVZK, note that the distributions with which Lemma 29 is instantiated are indeed efficiently samplable, as the sc-HVZK definition lets the witness be known to the distinguisher.

Game G_7. In this game, we add one more condition for a valid signature in Line 6 of the game as shown in Figure 3.10. This step simplifies the reduction from the UF-NMA game.



Game :

1: $\mathcal{M} := \varnothing$
2: $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
3: $(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{|H\rangle,\ \mathsf{Sign}(sk,\cdot)}(vk)$
4: Parse $\sigma^* = (w^*, z^*)$
5: $c^* := H(w^* \| \mu^*)$
6: **if** $c^* \neq H'(w^* \| \mu^*)$ **return** 0
7: **return** $\mu^* \notin \mathcal{M} \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

$\mathsf{Sign}(sk, \mu)$:

1: **if** $\mu \in \mathcal{M}$ **return** $\perp$
2: $\mathcal{M} := \mathcal{M} \cup \{\mu\}$
3: **if** $\mathsf{GetTrans}(\mu) = \Upsilon$ **return** Υ
4: $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]} := \mathsf{GetTrans}(\mu)$
5: **if** $z_\mu^{(\kappa)} = \perp$ **return** $\perp$
6: **return** $\sigma_\mu = (w_\mu^{(\kappa)}, z_\mu^{(\kappa)})$

$H(w\|\mu)$:

1: **if** $\mathsf{GetTrans}(\mu) = \Upsilon$ **return** Υ
2: $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]} := \mathsf{GetTrans}(\mu)$
3: **if** $\exists i (w = w_\mu^{(i)})$ **return** $c_\mu^{(i)}$
4: **return** $H'(w\|\mu)$

$\mathsf{GetTrans}(\mu)$:

1: $\kappa := 1,\ z_\mu^{(0)} := \perp$
2: **while** $z_\mu^{(\kappa-1)} = \perp$ and $\kappa \leq B$
3: $\quad c_\mu^{(\kappa)} := RF'(\mu\|\kappa)$
4: $\quad (w_\mu^{(\kappa)}, z_\mu^{(\kappa)}) :=$
$\qquad \mathsf{Sim}(vk, c_\mu^{(\kappa)}; RF''(\mu\|\kappa))$
5: $\quad \kappa := \kappa + 1$
6: $\mathfrak{L}_\mu := \{w_\mu^{(i)}\}_{i\in[\kappa]}$
7: **if** $\mathsf{Coll}(\mathfrak{L}_\mu)$ **return** Υ
8: **return** $\{(w_\mu^{(i)}, c_\mu^{(i)}, z_\mu^{(i)})\}_{i\in[\kappa]}$

Figure 3.10: Game G_7. The difference from Game G_6 is depicted in blue.

An adversary $\mathcal{A}$ distinguishes G_7 from G_6 only if it can find $(\mu^*, (w^*, z^*))$ such that

$$H(w^*\|\mu^*) \neq H'(w^*\|\mu^*) \wedge \mu^* \notin \mathcal{M} \wedge \mathsf{V}_2(vk, w^*, c^*, z^*).$$

For all i, we define the game $\widetilde{G}_i$ as the same as G_i except that the adversary wins if it finds a triple $(\mu^*, (w^*, z^*))$ such that $H(w^*\|\mu^*) \neq H'(w^*\|\mu^*)$. If $\mathcal{A}$ distinguishes G_7 and G_6, one can build a wrapper $\mathcal{R}$ around $\mathcal{A}$ that wins the game $\widetilde{G}_6$. Hence,

$$\left|\mathbb{P}[1 \Leftarrow G_7^{\mathcal{A}}] - \mathbb{P}[1 \Leftarrow G_6^{\mathcal{A}}]\right| \leq \mathbb{P}[1 \Leftarrow \widetilde{G}_6^{\mathcal{R}}].$$

Since $\widetilde{G}_i$ has the same oracles and interaction rules as in G_i, one can bound the winning probability of $\mathcal{R}$ in $\widetilde{G}_6$ with a similar argument as in the game transitions from G_2 to G_6.

$$\mathbb{P}[1 \Leftarrow \widetilde{G}_6^{\mathcal{R}}] \leq \mathbb{P}[1 \Leftarrow \widetilde{G}_2^{\mathcal{R}}] + (6Q_S + 6Q_H)^{\frac{3}{2}}\sqrt{B \cdot \varepsilon_{zk}} + 2(Q_S + Q_H) \cdot B \cdot 2^{\frac{-\alpha-1}{2}}.$$

This inequality gives an explicit upper bound on the distinguishing advantage of $\mathcal{A}$ since the winning probability of $\mathcal{R}$ in $\widetilde{G}_2$ is zero.

It remains to reduce the UF-NMA game of SIG_B to G_7. The adversary $\mathcal{B}^{|H'\rangle}$ can perfectly simulate the signing oracle and H for $\mathcal{A}$, and the random functions RF' and RF'' using another hash function G that is modeled as a random oracle. Whenever $\mathcal{A}$ outputs a forgery (μ^*, σ^*), it would also be a valid forgery for $\mathcal{B}$ and pass the verification thanks to Line 6 in Game G_7.

Runtime. For each signing or random oracle query, the reduction runs the HVZK simulator B times. To simulate the random functions RF' and RF'', one can use the private random oracle G that is accessible to the reduction (it is also possible to replace G with a quantum pseudo-random function). Therefore, the runtime of the reduction is essentially $\mathsf{Time}(\mathcal{A}) + T_{\mathsf{Sim}} \cdot B \cdot (Q_S + Q_H)$. □

3.3.2 Strong Unforgeability

We now analyze the strong unforgeability security of the signatures obtained by Fiat-Shamir with bounded aborts. Contrary to the previous results, the strong unforgeability security relies on both computational unique response and correctness of the underlying sigma protocol. Although the proofs of the theorems in this section share strong resemblances, they have delicate differences. We fully detail those differences.

In the next theorem, we reduce the $\mathsf{sUF\text{-}CMA}_1$ security of a signature obtained by FSwBA to its UF-NMA security. The proof is based on that of Theorem 11. A similar result holds for the sUF-CMA security and is formally stated in Section 3.3.3.

Theorem 12. *Let $\varepsilon_{zk}, \alpha, \beta, \gamma, T_{\mathsf{Sim}} \geq 0$, $B \geq 0$, H and G hash functions modeled as random oracles. Assume that $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is an $(\varepsilon_{zk}, T_{\mathsf{Sim}})$-HVZK and (γ, β)-correct public-coin identification protocol, and that the commitment message of the prover has min-entropy α. For any quantum adversary $\mathcal{A}$ against $\mathsf{sUF\text{-}CMA}_1$ security of $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ that issues at most Q_H quantum queries to the random oracle H and Q_S classical queries to the signing oracle, there exist quantum adversaries $\mathcal{B}$ and $\mathcal{C}$ respectively against UF-NMA security of SIG_B and the computational unique response property of Σ, with*

$$\mathsf{Time}(\mathcal{B}), \mathsf{Time}(\mathcal{C}) \approx \mathsf{Time}(\mathcal{A}) + T_{\mathsf{Sim}} \cdot B \cdot (Q_S + Q_H)$$

and such that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{sUF\text{-}CMA}_1}_{\mathsf{SIG}_B}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) &+ \mathsf{Adv}^{\mathsf{CUR}}_{\Sigma}(\mathcal{C}) \\ &+ 2^{\frac{-\alpha+3}{2}} \cdot B \cdot (Q_S + Q_H) \\ &+ 60\sqrt{\varepsilon_{zk} \cdot B} \cdot (Q_S + Q_H)^{\frac{3}{2}} \\ &+ 4(Q_H + 2)^2(1-\gamma) + \beta^B + B \cdot 2^{-\alpha} \ .\end{aligned}$$

Our reduction holds in the QROM and relies on $\mathcal{B}$ and $\mathcal{C}$ having access to a private random oracle H' with the same domain and range as H that is not accessible by $\mathcal{A}$.

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

Proof. The proof is based on similar hybrid games as in the proof of Theorem 11. We modify the games as follows. Instead of maintaining the list $\mathcal{M}$ of messages that were queried by the adversary via the signature oracle, the challenger also keeps the corresponding signatures to these messages. Let $\mathcal{MS}$ be the new list of message-signature pairs. Each game, at its final step, also checks whether the forgery $(\mu^*, (w^*, z^*))$ belongs to this list or not, and returns 0 if it does.

In terms of the advantage of the adversary, with these modifications, everything remains the same up to G_6. The adversary succeeds to distinguish the last game transition only if it can detect the following event:

$$H(w^* \| \mu^*) \neq H'(w^* \| \mu^*) \wedge (\mu^*, (w^*, z^*)) \notin \mathcal{MS} \wedge \mathsf{V}_2(vk, w^*, c^*, z^*) \ .$$

Since $H(w^* \| \mu^*) \neq H'(w^* \| \mu^*)$, the adversary must have queried μ^* before and w^* must have appeared in at least one of the transcripts produced during the rejection sampling of $\mathsf{Sign}(\mathsf{sk}, \mu^*)$. In the $\mathsf{UF\text{-}CMA}_1$ security, the adversary is only allowed to query μ^* once. Let $(w_{\mu^*}, c_{\mu^*}, z_{\mu^*})$ be the corresponding transcript of this query. We proceed by analyzing different cases.

- Event_1 : The transcript $(w_{\mu^*}, c_{\mu^*}, z_{\mu^*})$ is the final transcript produced during the execution of $\mathsf{Sign}(\mathsf{sk}, \mu^*)$ and it gets accepted by the verifier, namely, it holds that $\mathsf{V}_2(vk, w_{\mu^*}, c_{\mu^*}, z_{\mu^*}) = 1$. It violates the computational unique response of Σ since $(w^*, c^*, z^*) \neq (w^*, c^*, z_{\mu^*})$ and they both get accepted by the verifier. Therefore, the adversary $\mathcal{A}$ can be turned into another adversary $\mathcal{C}$ (that observes $\mathcal{A}$ and outputs the two transcripts) against the computational unique response property of Σ with an advantage that is at most $\mathsf{Adv}^{\mathsf{CUR}}_{\Sigma}(\mathcal{C})$. Note that $\mathcal{C}$ has the same runtime as $\mathcal{A}$.

- Event_2 : The transcript $(w_{\mu^*}, c_{\mu^*}, z_{\mu^*})$ is either the final transcript produced during the execution of $\mathsf{Sign}(\mathsf{sk}, \mu^*)$ that gets rejected by the verifier, or it is not the final transcript. We reverse the hybrid games back to G_5, where the transcripts are honestly-generated. Following the same technique in the proof of Theorem 11, we have

  $$\left| \mathbb{P}[\mathsf{Event}_2 \mid G_6^{\mathcal{A}}] - \mathbb{P}[\mathsf{Event}_2 \mid G_5^{\mathcal{A}}] \right| \leq (6Q_S + 6Q_H)^{\frac{3}{2}} \sqrt{B \cdot \varepsilon_{zk}} \ .$$

  Assume that the transcript is the final one and it gets rejected by the verifier. In G_5 the transcripts are all generated honestly with independent challenges in each iteration. Since Σ is (γ, β)-correct, we have $\mathbb{P}[z_{\mu^*} = \perp] \leq \beta^B$. Conditioned on $z_{\mu^*} \neq \perp$, the adversary would attmept to find a message μ^* that produces an incorrect transcript, i.e., $z_{\mu^*} \neq \perp$ such that $(w_{\mu^*}, c_{\mu^*}, z_{\mu^*})$ gets rejected by the verifier. The hash queries of the adversary may provide some information to find such a message. However, the success probability can be bounded by $4(Q_H + 2)(Q_H + 1)(1 - \gamma)$ using Lemma 30, where we used the fact that a non-aborting honestly-generated transcript is incorrect with probability at most $1 - \gamma$. Therefore, the probability of this case is bounded by

  $$4(Q_H + 2)(Q_H + 1)(1 - \gamma)(1 - \beta^B) + \beta^B \leq 4(Q_H + 2)^2(1 - \gamma) + \beta^B \ .$$

Now, assume that it is not the final transcript. Recall that we have $w_{\mu^*} = w^*$. This transcript has not been revealed to the adversary since it is not the final transcript. Moreover, revealing the final transcript does not reduce the min-entropy of w_{μ^*} since it is independent from $(w_{\mu^*}, c_{\mu^*}, z_{\mu^*})$. In G_5, the distribution of w_{μ^*} has min-entropy α, therefore, the probability of $w^* = w_{\mu^*}$ is at most $2^{-\alpha}$. Note that at most B transcripts are produced during the game. Therefore, the probability of this case is at most $B \cdot 2^{-\alpha}$.

By putting the two analyses together, we obtain

$$\mathbb{P}[\mathsf{Event}_2 \mid G_6^{\mathcal{A}}] \leq (6Q_S + 6Q_H)^{\frac{3}{2}}\sqrt{B \cdot \varepsilon_{zk}} + 4(Q_H+2)^2(1-\gamma) + \beta^B + B \cdot 2^{-\alpha} \ .$$

Finally, one can bound the distinguishing advantage of the adversary, using the union bound, as follows:

$$\begin{aligned}|\mathbb{P}[1 \Leftarrow G_7^{\mathcal{A}}] - \mathbb{P}[1 \Leftarrow G_6^{\mathcal{A}}]| &\leq \mathbb{P}[\mathsf{Event}_1 \mid G_6^{\mathcal{A}}] + \mathbb{P}[\mathsf{Event}_2 \mid G_6^{\mathcal{A}}] \\ &\leq \mathsf{Adv}_{\Sigma}^{\mathsf{CUR}}(\mathcal{C}) + (6Q_S + 6Q_H)^{\frac{3}{2}}\sqrt{B \cdot \varepsilon_{zk}} \\ &\quad + 4(Q_H+2)^2(1-\gamma) + \beta^B + B \cdot 2^{-\alpha} \ .\end{aligned}$$

The adversary $\mathcal{B}^{|H'\rangle}$ can perfectly simulate the signing oracle and H for $\mathcal{A}$, and the random functions RF' and RF'' using another hash function G that is modeled as a random oracle. To reduce the $\mathsf{UF\text{-}NMA}$ game of SIG_B to G_7, the reduction plays in G_7 against $\mathcal{A}$, and whenever $\mathcal{A}$ outputs a forgery (μ^*, σ^*), it would also be a valid forgery for $\mathcal{B}$ that passes the verification check. □

3.3.3 From $\mathsf{UF\text{-}CMA}_1$ and $\mathsf{sUF\text{-}CMA}_1$ to UF-CMA and sUF-CMA

In this section, we extend the results of Theorem 11 and 12 to the (s)UF-CMA security.

Theorem 13. *Let $\varepsilon_{zk}, \alpha, \beta, \gamma, T_{\mathsf{Sim}} \geq 0$, $B \geq 0$, H and G hash functions modeled as random oracles. Assume that $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is an $(\varepsilon_{zk}, T_{\mathsf{Sim}})$-HVZK and (γ, β)-correct public-coin identification protocol, and that the commitment message of the prover has min-entropy α. For any quantum adversary $\mathcal{A}$ against UF-CMA (or sUF-CMA) security of $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ that issues at most Q_H quantum queries to the random oracle H and Q_S classical queries to the signing oracle, there exist quantum adversaries $\mathcal{B}$ and $\mathcal{C}$ respectively against UF-NMA security of SIG_B and the computational unique response property of Σ, with $\mathsf{Time}(\mathcal{B}), \mathsf{Time}(\mathcal{C}) \approx \mathsf{Time}(\mathcal{A}) + T_{\mathsf{Sim}} \cdot B \cdot Q_S \cdot Q_H$ as follows:*

- *For the UF-CMA security, it holds that*

$$\mathsf{Adv}_{\mathsf{SIG}_B}^{\mathsf{UF\text{-}CMA}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathsf{SIG}_B}^{\mathsf{UF\text{-}NMA}}(\mathcal{B}) + 2^{\frac{-\alpha+3}{2}} \cdot B \cdot Q_S \cdot (Q_S + Q_H) + 30\sqrt{\varepsilon_{zk} \cdot B} \cdot (Q_S + Q_H)^{\frac{3}{2}} \ .$$

- *For the* sUF-CMA *security, it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{sUF\text{-}CMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{CUR}}_{\Sigma}(\mathcal{C}) \\ + 2^{\frac{-\alpha+3}{2}} \cdot B \cdot Q_S \cdot (Q_S + Q_H) \\ + 60\sqrt{\varepsilon_{zk} \cdot B} \cdot (Q_S + Q_H)^{\frac{3}{2}} \\ + Q_S \cdot \Big(4(Q_H+2)^2(1-\gamma) + \beta^B + B \cdot 2^{-\alpha}\Big).\end{aligned}$$

Both reductions hold in the QROM and rely on $\mathcal{B}$ and $\mathcal{C}$ having access to a private random oracle H' with the same domain and range as H that is not accessible by $\mathcal{A}$.

Proof. We first discuss the UF-CMA security. Since most of the proof is similar to that of Theorem 11, we just give a sketch. Consider an imaginary two-dimensional data structure (for example a table) that has $|\mathcal{M}|$ columns each one indexed by one message μ, that contains all the transcripts generated during the rejection sampling process in $\mathsf{Sign}(sk, \mu)$. Part of this table contains the view of the adversary. In the proof of Theorem 11, in the first two hybrid games we derandomized each cell of the data structure using a random function which takes as input the coordinate of the cell; its message and the row number (the iteration number). In the $\mathsf{UF\text{-}CMA}_1$ game, the adversary is supposed to choose Q_S columns (messages) and receive some information of each column (the signatures) and output a forgery. As long as the adversary is not allowed to query a message twice, this derandomization does not change the view of the adversary. This is not the case in the UF-CMA game. Moreover, we do not know the messages on which the adversary will query the signing oracle, and so we cannot assign appropriate randomness to the queries a priori. Instead, we consider a three-dimensional data structure such that each cell is uniquely determined by a message, an iteration number in $[B]$, and a query number in $[Q_S]$. One can see this three-dimensional table as the previous table that each column has expanded to Q_S columns. This new table contains the view of the adversary in the UF-CMA game and if we derandomize it with a random function that takes as input the coordinate of the cell, it does not change the view of the adversary. Now, the whole proof of Theorem 11 can be similarly repeated here with a small modification that each time we look into the two-dimensional table in the $\mathsf{UF\text{-}CMA}_1$ proof, we replace it with the three-dimensional one. We mention further details for the sake of completeness.

In the $\mathsf{UF\text{-}CMA}_1$ game, to consistently answer the random oracle query on input $w\|\mu$, we output some uniform element from the range of the function, unless the column indexed by μ contains a transcript with the commitment w in which case we output its corresponding challenge in the transcript. In the UF-CMA game, we search over the whole section of the message μ which contains roughly $B \cdot Q_S$ cells. This lookup in the table costs roughly $B \cdot Q_S$ operations.

In order to replace the real transcripts with the simulated ones, we take care of the collisions in the outputs of the random oracle (the challenges of the transcripts) in the table. This issue stems from the fact that in the simulated transcripts, all the challenges will be replaced by fresh random elements if there is any collision, they have to be updated accordingly. Recall that each challenge is evaluated as $H(w\|\mu)$. In the $\mathsf{UF\text{-}CMA}_1$ game, since there is no repeating message, the possible collisions only appear in the same column which has size at most B. This probability of collision was captured in the fourth hybrid game in the proof of Theorem 11. In the UF-CMA game, the possible collisions are

spread over the whole section of the message μ. One can update the fourth hybrid game accordingly and compute the probability of success similarly.

After handling the collisions, we change the real transcripts with simulated ones. The only issue that requires to be taken care of is that the forged signature $(\mu^*, (w^*, z^*))$ by the adversary must not intersect with the reprogrammed ones. The proof is similar to that of Theorem 11 in the last hybrid up to replacing the list $\mathfrak{L}_{\mu^*}$ which is the column indexed by μ^* with the whole section of the message μ^* in the three-dimensional table.

For the sUF-CMA, we modify the games as follows. Instead of maintaining the list $\mathcal{M}$ of messages that were queried by the adversary via the signature oracle, the challenger also keeps the corresponding signatures to these messages. Let $\mathcal{MS}$ be the new list of message-signature pairs. Each game, at its final step, also checks whether the forgery $(\mu^*, (w^*, z^*))$ belongs to this list or not, and returns 0 if it does.

In terms of the advantage of the adversary, with these modifications, everything remains the same up to the last game hop. Let G_s and G_f be the last two games. The adversary succeeds to distinguish the last game hop only if it can detect the following event:

$$H(w^* \| \mu^*) \neq H'(w^* \| \mu^*) \wedge (\mu^*, (w^*, z^*)) \notin \mathcal{MS} \wedge \mathsf{V}_2(vk, w^*, c^*, z^*) \ .$$

Since $H(w^* \| \mu^*) \neq H'(w^* \| \mu^*)$, the adversary must have queried μ^* before and w^* must have appeared in at least one of the transcripts produced during the rejection sampling of $\mathsf{Sign}(\mathsf{sk}, \mu^*)$. Let $\mathfrak{S}$ be the list of indices where the adversary asked for such queries. We proceed case by case.

- Event_1 : There exists at least one signature query in $\mathfrak{S}$ such that the value of w^* has appeared in the final transcript produced during the execution of $\mathsf{Sign}(\mathsf{sk}, \mu^*)$ such that its corresponding transcript gets accepted by the verifier. Similar to the proof of Theorem 12, the probability of this case can be bounded by $\mathsf{Adv}^{\mathsf{CUR}}_{\Sigma}(\mathcal{C})$ where $\mathcal{C}$ is an algorithm with the same runtime as $\mathcal{A}$.

- Event_2 : For every signature query in $\mathfrak{S}$, if the value of w^* has appeared in the final transcript, then the final transcript gets rejected by the verifier. Let $\mathfrak{S}_{\text{final}}$ be the subset corresponding to these queries. We first replace the simulated transcripts with the honestly-generated ones. Let G_h denote this transition. Following the same technique in the proof of Theorem 11, we have

$$\left| \mathbb{P}[\mathsf{Event}_2 \mid G_s^{\mathcal{A}}] - \mathbb{P}[\mathsf{Event}_2 \mid G_h^{\mathcal{A}}] \right| \leq (6Q_S + 6Q_H)^{\frac{3}{2}} \sqrt{B \cdot \varepsilon_{zk}} \ .$$

  In G_h the transcripts are all generated honestly with independent challenges in each iteration. For each index $i \in \mathfrak{S}_{\text{final}}$ in game G_h, we have

$$
\begin{aligned}
&\mathbb{P}[\mathsf{Ver}(vk, \mathsf{Sign}(sk, \mu^*)) = 0 \mid i\text{-th query}] \\
&\leq \mathbb{P}[\mathsf{Sign}(sk, \mu^*) = (\cdot, \perp) \mid i\text{-th query}] \\
&\qquad + \mathbb{P}[\mathsf{Ver}(vk, \mathsf{Sign}(sk, \mu^*)) = 0 \mid i\text{-th query} \wedge \mathsf{Sign}(sk, \mu^*) \neq (\cdot, \perp)] \\
&\qquad\quad \cdot \mathbb{P}[\mathsf{Sign}(sk, \mu^*) \neq (\cdot, \perp)] \\
&\leq \beta^B + \mathbb{P}[\mathsf{Ver}(vk, \mathsf{Sign}(sk, \mu^*)) = 0 \mid i\text{-th query} \wedge \mathsf{Sign}(sk, \mu^*) \neq (\cdot, \perp)] \\
&\qquad\quad \cdot (1 - \beta^B) \\
&\leq \beta^B + 4(Q_H + 2)(Q_H + 1)(1 - \gamma)(1 - \beta^B) \quad \text{(by Lemma 30)} \\
&\leq \beta^B + 4(Q_H + 2)^2(1 - \gamma),
\end{aligned}
$$

where we used the fact that Σ is (γ, β)-correct. Note that this is an upper bound on the probability of occurring of each signature query in $\mathfrak{S}_{\text{final}}$.

Now, take a signature query in $\mathfrak{S} \setminus \mathfrak{S}_{\text{final}}$. Let $(w_{\mu^*}, c_{\mu^*}, z_{\mu^*})$ denote the transcript during the signing algorithm with $w_{\mu^*} = w^*$. This transcript has not been revealed to the adversary since it is not the final transcript. Moreover, revealing the final transcript does not reduce the min-entropy of w_{μ^*} since it is independent from $(w_{\mu^*}, c_{\mu^*}, z_{\mu^*})$. In G_h, the distribution of w_{μ^*} has min-entropy α, therefore, the probability of $w^* = w_{\mu^*}$ is at most $2^{-\alpha}$. Note that at most B transcripts are produced during the game. Hence, each signature query in $\mathfrak{S} \setminus \mathfrak{S}_{\text{final}}$ occurs with probability at most $B \cdot 2^{-\alpha}$.

Putting the two analyses together with the fact that $|\mathfrak{S}| \leq Q_S$, we obtain

$$
\begin{aligned}
\mathbb{P}[\mathsf{Event}_2 \mid G_s^{\mathcal{A}}] \leq (6Q_S + 6Q_H)^{\frac{3}{2}} \sqrt{B \cdot \varepsilon_{zk}} \\
+ Q_S\Big(4(Q_H + 2)^2(1 - \gamma) + \beta^B + B \cdot 2^{-\alpha}\Big) .
\end{aligned}
$$

Finally, one can bound the distinguishing advantage of the adversary, using the union bound, as follows:

$$
\begin{aligned}
|\mathbb{P}[1 \Leftarrow G_f^{\mathcal{A}}] - \mathbb{P}[1 \Leftarrow G_s^{\mathcal{A}}]| &\leq \mathbb{P}[\mathsf{Event}_1 \mid G_s^{\mathcal{A}}] + \mathbb{P}[\mathsf{Event}_2 \mid G_s^{\mathcal{A}}] \\
&\leq \mathsf{Adv}_{\Sigma}^{\mathsf{CUR}}(\mathcal{C}) + (6Q_S + 6Q_H)^{\frac{3}{2}} \sqrt{B \cdot \varepsilon_{zk}} \\
&\quad + Q_S\Big(4(Q_H + 2)^2(1 - \gamma) + \beta^B + B \cdot 2^{-\alpha}\Big).
\end{aligned}
$$

The remaining of the proof is similar to that of Theorem 12. □

3.4 Security of FSwBA: the Adaptive Reprogramming Approach

We show how to reduce UF-CMA security of the signature to UF-NMA security, separately in the ROM and QROM. Our proof in the ROM yields a tighter bound compared to our QROM proof.

We use similar frameworks for adaptive reprogramming (Lemmas 32 and 31) in the ROM and the QROM. Also, we note that our proof is crucially based on the zero-knowledge simulators in Definition 28 and 29.

Theorem 14. *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 an $(\varepsilon_{zk}, T_{\mathsf{Sim}})$-HVZK public-coin identification protocol, and that the commitment message of the prover has min-entropy α. Let $\mathcal{A}$ be any arbitrary 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 classical queries to the signing oracle. There exists a quantum adversary $\mathcal{B}$ against UF-NMA security of SIG_B as follows:*

- *In the ROM, the runtime of $\mathcal{B}$ is $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H) \log(B \cdot Q_S + Q_H))$, and*

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

- *In the QROM, the runtime of $\mathcal{B}$ is $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H) \log(B \cdot Q_S))$ with QRACM, and $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H) \cdot (B \cdot Q_S))$ without QRACM, and*

$$\mathsf{Adv}^{\mathsf{CMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) + 2^{-\frac{\alpha}{2}} \cdot \frac{3B \cdot Q_S}{2} \cdot \sqrt{(B \cdot Q_S + Q_H + 1)} + \varepsilon_{zk} \cdot B \cdot Q_S \ .$$

  The reduction in the QROM relies on $\mathcal{B}$ having access to a private random oracle H' with the same domain and range as H that is not accessible by $\mathcal{A}$.

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

In the ROM, our reduction simulates the random oracle using the lazy sampling method. We note that one can also use a private random oracle H' instead. Although it would make the proof conceptually simpler by handling both cases in the same way, it increases the runtime of the reduction.

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

Game G_0. The first game is the UF-CMA security game (Figure 3.11).

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 3.12.

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 3.13.

Note that if $b = 0$ in Figure 3.13, then $\mathcal{D}$ perfectly simulates G_0, and otherwise it perfectly simulates G_1. Therefore,

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

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 32 bounds the advantage of $\mathcal{D}$

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 3.11: 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 3.12: Game G_1. The difference from G_0 is highlighted in blue.

by $B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1)2^{-\alpha}$. In the QROM, using Lemma 31, it follows that the advantage of $\mathcal{D}$ is bounded by

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

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 3.14.

We would like to bound the distance between games G_1 and G_2 using the zero-knowledge property. First we discuss the QROM case. Suppose that we are given a random oracle H' and $B \cdot Q_S$ transcripts that are either sampled honestly or sampled by the simulator. We use them to simulate G_1 or G_2, respectively. Note that in both games, after each transcript, the random oracle is reprogrammed according to the transcript. In

$\mathcal{D}^{H_b,\mathsf{Reprogram}}$:

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** $\mu^* \notin \mathcal{M} \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

$\mathsf{Reprogram}(\mu, sk)$:

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)

$\mathsf{Sign}(sk, \mu)$:

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

Figure 3.13: The distinguisher $\mathcal{D}$.

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** $\mu^* \notin \mathcal{M} \wedge \mathsf{V}_2(vk, w^*, c^*, z^*)$

$\mathsf{Sign}(sk, \mu)$:

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

$\mathsf{GetTrans}(\mu)$:

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

Figure 3.14: Game G_2. The difference from G_1 is highlighted in blue.

order to simulate the reprogrammed random oracle perfectly, we keep track of a list $\mathfrak{D}$ of the classical values in which the random oracle must be reprogrammed. We describe the details in Figure 3.15.

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

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

The ROM case is similar except that instead of using the private random oracle H' to simulate H, we use the lazy sampling method. We obtain

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

Game G_3. The signing algorithm does not use the signing key anymore and uses the zero-

$\underline{\mathcal{C}^{|H'\rangle}(\{w_{i,\kappa}, c_{i,\kappa}, z_{i,\kappa}\}_{i\in[Q_S],\kappa\in[B]}) :}$

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

knowledge simulator to answer the sign queries. The 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 an element freshly sampled from the range of H and we add it in the database. In the QROM, we cannot simulate the random oracle with the lazy sampling method since the access to it is quantum Therefore, the challenger uses a private random oracle H' to simulate the hash queries of the adversary but also keeps a database of reprogrammed inputs. Whenever it receives a hash query, it first searches the database for a match and, if there is none, it returns the evaluation of the query with the private random oracle H'. We refer the reader to Figure 3.16 for the details of this game. We note that this is only a syntactic modification since the reprogramming has already been carried out in G_1.

Game G_4. This game only concerns the QROM. We add one more statement to the winning conditions. Let $(\mu^*, (w^*, z^*))$ be the forgery. The game aborts if $H(w^*\|\mu^*) \neq H'(w^*\|\mu^*)$ where H' is the random oracle used to simulate the hash queries of the adversary in the previous game. The adversary can distinguish this modification if the value $w^*\|\mu^*$ has been programmed during the game. This occurs only 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 view of the adversary view is identical to that of the previous one.

It remains to reduce the UF-NMA game of SIG_B to the last hybrid game (G_3 in the ROM and G_4 in the QROM). In the QROM, using the UF-NMA game and its random oracle, one can perfectly simulate G_4 for the adversary. If the adversary $\mathcal{A}$ finds a forgery (μ^*, σ^*), then the random oracle has not been reprogrammed at this value during the course of G_4 since it has not been queried before. Hence, it would be a valid signature for the UF-NMA game. In the ROM, the reduction from G_3 to UF-NMA works as follows. The reduction uses the random oracle of the UF-NMA game to answer direct random oracle queries. More precisely, let H' be the random oracle of the UF-NMA game, then the reduction modifies H as in Figure 3.17. Note that this perfectly simulates the view of the adversary.

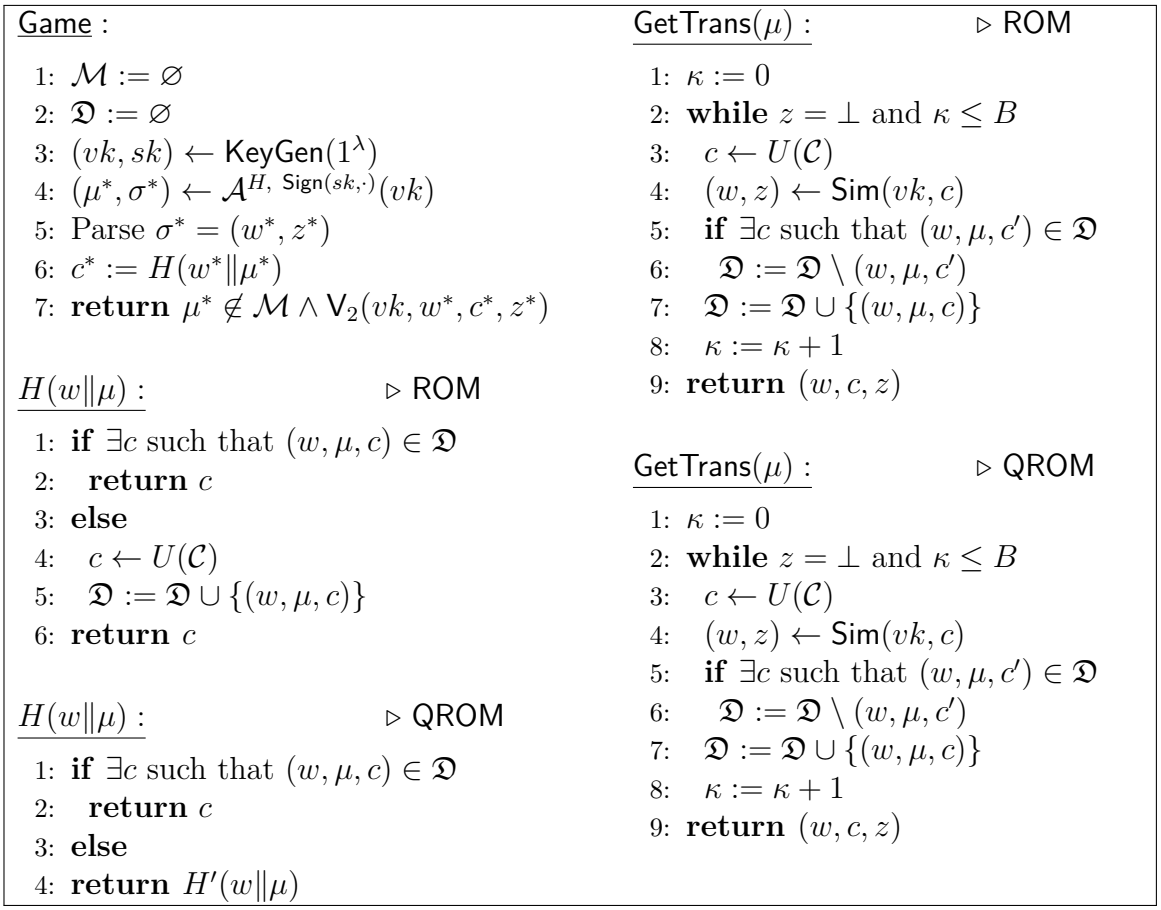


Figure 3.16: Random oracle simulation in Game G_3.

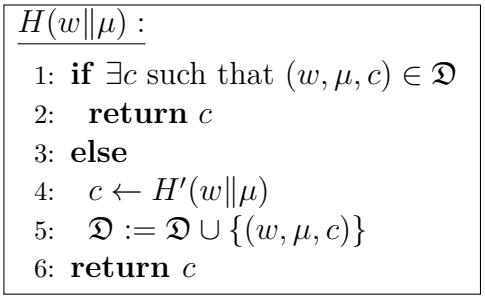


Figure 3.17: The simulation of the random oracle in G_3 by the reduction from G_3 to UF-NMA in the ROM.

A forged signature (μ^*, σ^*) in G_3 is a forged signature in UF-NMA if $c^* = H'(w^* \| \mu^*)$ where H' is the random oracle of UF-NMA. This holds if the random oracle has not been reprogrammed on input $w^* \| \mu^*$ during the game, which holds since the adversary is not allowed to ask for a signature of μ^*.

Runtime. We discuss two cases separately.

- 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 $\mathsf{Time}(\mathcal{A}) + \mathcal{O}(T_{\mathsf{Sim}} \cdot (B \cdot Q_S + Q_H) \cdot \log(B \cdot Q_S + Q_H))$.

- In the QROM. We split the runtime analysis in two different models depending on whether we have access to QRACM or not. To answer the hash and sign queries properly, the reduction maintains a database of reprogrammed input-outputs, and at each query, it searches over the database to find a match. Note that it is being carried out in superposition. The size of the database is at most $B \cdot Q_S$, and a naive exhaustive search takes $B \cdot Q_S$. Moreover, for each sign query, the reduction runs the zero-knowledge simulator at most B times. Thus, the runtime would be $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H)(B \cdot Q_S))$. With QRACM, the reduction has the advantage to maintain a sorted database and quantumly search over the database. It reduces the search time to $\log(B \cdot Q_S)$. It yields the runtime $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H) \log(B \cdot Q_S))$.

□

3.4.1 Strong Unforgeability

The following theorem reduces the sUF-CMA security to the UF-NMA security for a signature obtained by FSwBA.

Theorem 15. *Let $\varepsilon_{zk}, \alpha, \beta, \gamma, T_{\mathsf{Sim}} \geq 0$, $B \geq 0$, 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 an $(\varepsilon_{zk}, T_{\mathsf{Sim}})$-HVZK and (γ, β)-correct public-coin identification protocol, and that the commitment message of the prover has min-entropy α. Let $\mathcal{A}$ be any arbitrary adversary against sUF-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 classical queries to the signing oracle. There exist quantum adversaries $\mathcal{B}$ and $\mathcal{C}$ respectively against UF-NMA security of SIG_B and the computational unique response property of Σ as follows:*

- *In the ROM, both $\mathcal{B}$ and $\mathcal{C}$ run in $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H) \log(B \cdot Q_S + Q_H))$, and*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{sCMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) &+ \mathsf{Adv}^{\mathsf{CUR}}_{\Sigma}(\mathcal{C}) \\ &+ 2^{-\alpha} \cdot B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1) \\ &+ 2 \cdot \varepsilon_{zk} \cdot B \cdot Q_S \\ &+ Q_S \cdot \left((1-\gamma) + \beta^B + B \cdot 2^{-\alpha}\right) .\end{aligned}$$

- *In the QROM, both $\mathcal{B}$ and $\mathcal{C}$ run in $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H) \log(B \cdot Q_S))$ with QRACM, and $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_{\mathsf{Sim}} \cdot B \cdot Q_S + Q_H) \cdot (B \cdot Q_S))$ without QRACM, and*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{sCMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) &+ \mathsf{Adv}^{\mathsf{CUR}}_{\Sigma}(\mathcal{C}) \\ &+ 2^{-\frac{\alpha}{2}} \cdot \frac{3B \cdot Q_S}{2} \cdot \sqrt{(B \cdot Q_S + Q_H + 1)} \\ &+ 2 \cdot \varepsilon_{zk} \cdot B \cdot Q_S \\ &+ Q_S \cdot \left((1-\gamma) + \beta^B + B \cdot 2^{-\alpha}\right) .\end{aligned}$$

The reduction in the QROM relies on $\mathcal{B}$ and $\mathcal{C}$ having access to a private random oracle H' with the same domain and range as H that is not accessible by $\mathcal{A}$.

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

Proof. We base the proof on the hybrid games in the proof of Theorem 14. We modify them as follows. Here, the challenger maintains the list $\mathcal{MS}$ of message-signature pairs that were queried by the adversary via the signature oracle. Each game, at its final step, also checks whether the forgery $(\mu^*, (w^*, z^*))$ belongs to this list or not, and returns 0 if it does. With these modifications, everything remains the same up to G_3.

We first discuss the QROM case. The two games G_3 and G_4 behave differently only if we have the following conditions:

$$H(w^* \| \mu^*) \neq H'(w^* \| \mu^*) \wedge (\mu^*, (w^*, z^*)) \notin \mathcal{MS} \wedge \mathsf{V}_2(vk, w^*, c^*, z^*) \ .$$

Since $H(w^* \| \mu^*) \neq H'(w^* \| \mu^*)$, the adversary must have queried μ^* before and w^* must have appeared in at least one of the transcripts produced during the rejection sampling of $\mathsf{Sign}(\mathsf{sk}, \mu^*)$. Let $\mathfrak{S}$ be the list of indices where the adversary asked for such queries. We proceed case by case.

- Event_1 : There exists at least one signature query in $\mathfrak{S}$ such that the value of w^* has appeared in the final transcript produced during the execution of $\mathsf{Sign}(\mathsf{sk}, \mu^*)$ such that its corresponding transcript gets accepted by the verifier. This can be bounded by the computational unique response of Σ. Let $(w_{\mu^*}, z_{\mu^*}) \leftarrow \mathsf{Sign}(\mathsf{sk}, \mu^*)$ be the output of this particular signature query and $c_{\mu^*} := H(w_{\mu^*} \| \mu^*)$. It holds that $\mathsf{V}_2(vk, w_{\mu^*}, c_{\mu^*}, z_{\mu^*}) = 1$. Moreover, the forgery (w^*, c^*, z^*) passes the verification check. Therefore, the adversary $\mathcal{A}$ can be turned into another adversary $\mathcal{C}$ (that observes $\mathcal{A}$ and outputs the two transcripts) against the computational unique response property of Σ with an advantage at most $\mathsf{Adv}^{\mathsf{CUR}}_{\Sigma}(\mathcal{C})$. Note that $\mathcal{C}$ has the same runtime as $\mathcal{A}$.

- Event_2 : For every signature query in $\mathfrak{S}$, if the value of w^* has appeared in the final transcript, then the final transcript gets rejected by the verifier. Let $\mathfrak{S}_{\text{final}}$ be the subset corresponding to these queries. We reverse the hybrid games back to G_1, where the transcripts are honestly-generated. Following the same technique in the proof of Theorem 14, we have

$$\left| \mathbb{P}[\mathsf{Event}_2 \mid G_2^{\mathcal{A}}] - \mathbb{P}[\mathsf{Event}_2 \mid G_1^{\mathcal{A}}] \right| \leq B \cdot Q_S \cdot \varepsilon_{zk} \ .$$

  In G_1 the transcripts are all generated honestly with independent challenges in each iteration. The Σ-protocol is (γ, β)-correct. Therefore, each query in $\mathfrak{S}_{\text{final}}$ occurs with probability at most

$$(1 - \gamma)(1 - \beta^B) + \beta^B \leq (1 - \gamma) + \beta^B \ .$$

  Further, we show that a signature query of type $\mathfrak{S} \setminus \mathfrak{S}_{\text{final}}$ is unlikely to occur. Let $(w_{\mu^*}, c_{\mu^*}, z_{\mu^*})$ be the corresponding transcript of such a query with $w_{\mu^*} = w^*$. This transcript has not been revealed to the adversary during the game since it was

not the final transcript. Moreover, revealing the final transcript does not reduce the min-entropy of w_{μ^*} since it is independent from $(w_{\mu^*}, c_{\mu^*}, z_{\mu^*})$. The distribution of w_{μ^*} has min-entropy α in G_5 which implies $\mathbb{P}[w^* = w_{\mu^*}] \leq 2^{-\alpha}$. Note that at most B transcripts are produced during the game. Hence, each signature query in $\mathfrak{S} \setminus \mathfrak{S}_{\text{final}}$ occurs with probability at most $B \cdot 2^{-\alpha}$.

By putting the two bounds above together, we obtain

$$\mathbb{P}[\mathsf{Event}_2 \mid G_2^{\mathcal{A}}] \leq B \cdot Q_S \cdot \varepsilon_{zk} + Q_S\Big((1-\gamma) + \beta^B + B \cdot 2^{-\alpha}\Big) ,$$

where we used $|\mathfrak{S}| \leq Q_S$.

Finally, using the union bound, it holds that:

$$\begin{aligned} |\mathbb{P}[1 \Leftarrow G_3^{\mathcal{A}}] - \mathbb{P}[1 \Leftarrow G_2^{\mathcal{A}}]| &\leq \mathbb{P}[\mathsf{Event}_1 \mid G_2^{\mathcal{A}}] + \mathbb{P}[\mathsf{Event}_2 \mid G_2^{\mathcal{A}}] \\ &\leq \mathsf{Adv}_{\Sigma}^{\mathsf{CUR}}(\mathcal{C}) + B \cdot Q_S \cdot \varepsilon_{zk} \\ &\quad + Q_S\Big((1-\gamma) + \beta^B + B \cdot 2^{-\alpha}\Big) . \end{aligned}$$

The reduction from the UF-NMA game to the sUF-CMA game is similar to that of Theorem 14.

For the ROM case, we reduce the UF-NMA game to G_3 as follows. One can use the random oracle of the UF-NMA game to answer the direct hash queries of the adversary in G_3. This preserves the view of the adversary. Therefore, showing that a valid forgery for G_3 is also a valid forgery for the UF-NMA game completes the proof. A similar argument and upper bound, as in the QROM, holds here. □

3.5 Security of FSwUA

We finally prove the security of the unbounded version of the Fiat-Shamir transform in both ROM and QROM. We note that our proof in the ROM is tighter. We reduce the T'-UF-CMA security of the unbounded signature scheme to the UF-CMA security of the bounded one in the QROM.

Theorem 16. *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'-UF-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 UF-CMA security of $\mathsf{SIG}_B = \mathsf{FS}_B[\Sigma, H]$ is such that $|\mathsf{Adv}^{T'\text{-UF-CMA}}_{\mathsf{SIG}_\infty}(\mathcal{A}) - \mathsf{Adv}^{\mathsf{UF-CMA}}_{\mathsf{SIG}_B}(\mathcal{A})|$ is bounded as*

$$Q_S \cdot \beta^B + \frac{\beta^B \cdot 2^{-\alpha}}{(1-\beta)^3} + \begin{cases} 2^{-\alpha} \cdot B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1) & \text{in the ROM,} \\ 2^{-\frac{\alpha}{2}} \cdot \frac{3B \cdot Q_S}{2} \cdot \sqrt{(B \cdot Q_S + Q_H + 1)} & \text{in the QROM.} \end{cases}$$

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

Proof. We proceed with three hybrid games.

Game G_0. We define Game G_0 as the UF-CMA security of SIG_B.

Game G_1. Let Game G_1 be game T'-UF-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. The similar statement holds in the QROM. 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

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

We aim at bounding the winning probability of $\mathcal{R}$. Remember G_1 from Figure 3.12, which we rename G'_0 in this proof. In Theorem 14, we proved that

$$\left|\mathbb{P}[1 \Leftarrow G_0^{\mathcal{A}}] - \mathbb{P}[1 \Leftarrow G_0'^{\mathcal{A}}]\right| \leq \frac{3B \cdot Q_S}{2} \cdot \sqrt{(B \cdot Q_S + Q_H + 1)2^{-\alpha}},$$

in the QROM, and

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

in the ROM. It follows that we can replace Game G_0 in $\mathbb{P}[\mathsf{win}(\mathcal{R}^{G_0,\mathcal{A}})]$ 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 8 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. □

3.6 Security of FSwBA with the Rényi Divergence

As [DFPS22] mentions, defining a version of HVZK that relies on the Rényi divergence instead of the statistical distance allows to prove the security of a larger class of Fiat-Shamir signatures. In some cases, this allows to achieve smaller signature sizes. For the sake of simplicity, we restrict ourselves to the case of Rényi divergence of infinite order. We need the following definition.

Definition 36 (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 3.18*

$\mathsf{Sim}(x)$:

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 3.18: Simulator decomposition.

It is shown in [DFPS23] that there exists a decomposable simulator as above for Lyubashevsky's Σ-protocol. With this formalism, we are able to extend the HVZK definition to the Rényi divergence.

Definition 37 (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 a $(\varepsilon_{zk}, T_\perp)$-*HVZK *(or* sc-HVZK*) simulator for the Σ-protocol with 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 distribution of a transcript (w', c', z') of the Σ-protocol conditioned on $z' \neq \perp$ satisfy*

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

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 14 and its proof to this new setting.

Theorem 17. *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. Assume that $\Sigma = ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is an $(\varepsilon_{zk}, T_\perp, R_{zk}, T_{\not\perp})$-*DDHVZK *public-coin identification protocol with a p-decomposable simulator, then we have the following updates on Theorem 14.*

- *In the ROM:*
  the runtime of $\mathcal{B}$ is $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_\perp(B-1)Q_S + T_{\not\perp}Q_S)\log(B \cdot Q_S + Q_H))$*, and its advantage satisfies*

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

- *In the QROM:*
  the runtime of $\mathcal{B}$ *is* $\mathsf{Time}(\mathcal{A})+\mathcal{O}((T_\perp(B-1)Q_S+T_{\not\perp}Q_S)\log(B\cdot Q_S))$ *with QRACM, and* $\mathsf{Time}(\mathcal{A})+\mathcal{O}((T_\perp(B-1)Q_S+T_{\not\perp}Q_S)\cdot(B\cdot Q_S))$ *without QRACM, and its advantage satisfies*

$$\mathsf{Adv}^{\mathsf{CMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq R_{zk}^{Q_S}\cdot\left(\mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B})+(\varepsilon_{zk}+|p-\beta|)\cdot B\cdot Q_S\right) + 2^{-\frac{\alpha}{2}}\cdot\frac{3B\cdot Q_S}{2}\cdot\sqrt{(B\cdot Q_S+Q_H+1)}\ .$$

  The reduction in the QROM relies on $\mathcal{B}$ *having access to a private random oracle* H' *with the same domain and range as* H *that is not accessible by* $\mathcal{A}$*.*

Proof. The proof is similar to the one of Theorem 14. 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 14.

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

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

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 $(w,z)\leftarrow\mathsf{Sim}_\perp(vk,c)$, and if $b=0$ run $(w,z)\leftarrow\mathsf{Sim}_{\not\perp}(vk,c)$. 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 14. □

Finally, we adapt the above analysis to the strong unforgeability case as follows.

Theorem 18. *Let* $R_{zk}\geq 1,\varepsilon_{zk},T_\perp,T_{\not\perp},\gamma,\beta\geq 0,p\in[0,1]$*, 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 an* $(\varepsilon_{zk},T_\perp,R_{zk},T_{\not\perp})$-DDHVZK *public-coin* (γ,β)*-correct identification protocol with a* p*-decomposable simulator, then we have the following updates on Theorem 15.*

- *In the ROM,* $\mathcal{B}$ *and* $\mathcal{C}$ *run in* $\mathsf{Time}(\mathcal{A})+\mathcal{O}((T_\perp(B-1)Q_S+T_{\not\perp}Q_S)\log(B\cdot Q_S+Q_H))$*, and*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{sCMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq R_{zk}^{Q_S} \cdot \Big(&\mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{CUR}}_{\Sigma}(\mathcal{C}) \\ &+ 2 \cdot (\varepsilon_{zk} + |p - \beta|) \cdot B \cdot Q_S \\ &+ Q_S\Big((1 + \varepsilon_{zk})(1 - \gamma) + B \cdot 2^{-\alpha}\Big)\Big) \\ &+ 2^{-\alpha} \cdot B \cdot Q_S \cdot (B \cdot Q_S + Q_H + 1) \ .\end{aligned}$$

- *In the QROM, $\mathcal{B}$ and $\mathcal{C}$ run in $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_\perp(B-1)Q_S + T_{\not\perp}Q_S)\log(B \cdot Q_S))$ with QRACM, and $\mathsf{Time}(\mathcal{A}) + \mathcal{O}((T_\perp(B-1)Q_S + T_{\not\perp}Q_S) \cdot (B \cdot Q_S))$ without QRACM, and*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{sCMA}}_{\mathsf{SIG}_B}(\mathcal{A}) \leq R_{zk}^{Q_S} \cdot \Big(&\mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{SIG}_B}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{CUR}}_{\Sigma}(\mathcal{C}) \\ &+ 2 \cdot (\varepsilon_{zk} + |p - \beta|) \cdot B \cdot Q_S \\ &+ Q_S\Big((1 + \varepsilon_{zk})(1 - \gamma) + B \cdot 2^{-\alpha}\Big)\Big) \\ &+ 2^{-\frac{\alpha}{2}} \cdot B \cdot \left(\frac{3Q_S}{2} \cdot \sqrt{(B \cdot Q_S + Q_H + 1)} + 2^{-\frac{\alpha}{2}} R_{zk}^{Q_S}\right) \ .\end{aligned}$$

  The reduction in the QROM relies on $\mathcal{B}$ and $\mathcal{C}$ having access to a private random oracle H' with the same domain and range as H that is not accessible by $\mathcal{A}$.

Proof. We only discuss the QROM case. The ROM proof is similar. We use the same sequence of hybrid games as in the proof of Theorem 17. We modify the hybrid games as follows. Each game maintains a list $\mathcal{MS}$ of all message-signature pairs generated during the course of the game. We update the winning condition of the game by asking the forged message-signature pair to not belong to $\mathcal{MS}$. The advantage of the adversary between G_0 and G_3 can be bounded from above by a similar argument as in Theorem 17. For the last game hop, note that the adversary can distinguish between G_3 and G_4 if it can detect the following event:

$$H(w^* \| \mu^*) \neq H'(w^* \| \mu^*) \wedge (\mu^*, (w^*, z^*)) \notin \mathcal{MS} \wedge \mathsf{V}_2(vk, w^*, c^*, z^*) \ .$$

We separately analyze the following cases.

Let $\mathfrak{S}$ be the list of indices where the adversary asked for such queries. We proceed case by case.

- Event_1 : There exists at least one signature query in $\mathfrak{S}$ such that the value of w^* has appeared in the final transcript produced during the execution of $\mathsf{Sign}(\mathsf{sk}, \mu^*)$ such that the corresponding transcript passes the verification. In this case, one can construct an adversary $\mathcal{C}$ that attacks the computational unique response of Σ by observing $\mathcal{A}$ and outputting the corresponding transcript and the forgery of $\mathcal{A}$. Note that both transcripts pass the verification, therefore, the output of $\mathcal{C}$ breaks the computational unique response of Σ.

- Event_2 : For every signature query in $\mathfrak{S}$, if the value of w^* has appeared in the final transcript, then the final transcript does not pass the verification. Let $\mathfrak{S}_{\text{final}}$ be the subset corresponding to these queries. It seems that one could reverse the hybrid games back to G_0 to obtain honestly-generated transcripts. However, since the Rényi divergence is not symmetric, we do not know how to bound the distinguishing advantage of the adversary from above in the reverse direction. We show how one can bound $\mathbb{P}[\mathsf{Event}_2]$ in $G_{2,1}$. Therefore, reversing the games back to $G_{2,1}$ would be sufficient. Following the same argument as in Theorem 17 that from G_3 to $G_{2.1}$ the cost is additive and is equal to:

$$\left|\mathbb{P}[\mathsf{Event}_2 \mid G_3^{\mathcal{A}}] - \mathbb{P}[\mathsf{Event}_2 \mid G_{2.1}^{\mathcal{A}}]\right| \leq (\varepsilon_{zk} + |p-\beta|) \cdot B \cdot Q_S \ .$$

  In $G_{2,1}$, the transcripts except the valid one (if there exists any it would be the last transcript) are honestly sampled according to Σ. In particular, the challenges are sampled uniformly and independently from each other. According to the correctness of Σ, the probability of obtaining a transcript where $z \neq \perp$ is $1-\beta^B$. Recall that a valid ($z \neq \perp$) and honestly-generated transcript is also a valid signature with probability at least γ. In $G_{2,1}$, the valid signature is replaced by a simulated one. Therefore, we have

$$\begin{aligned}
&\mathbb{P}[\mathsf{V}_2(vk, w_{\mu^*}, c_{\mu^*}, z_{\mu^*}) = 0 \mid G_{2.1}^{\mathcal{A}} \wedge z_{\mu^*} \neq \perp] \\
&\leq (1+\varepsilon_{zk}) \cdot \mathbb{P}[\mathsf{V}_2(vk, w_{\mu^*}, c_{\mu^*}, z_{\mu^*}) = 0 \mid G_1^{\mathcal{A}} \wedge z_{\mu^*} \neq \perp] \\
&\leq (1+\varepsilon_{zk})(1-\gamma)(1-\beta^B) \\
&\leq (1+\varepsilon_{zk})(1-\gamma) \ .
\end{aligned}$$

  Let $(w_{\mu^*}, c_{\mu^*}, z_{\mu^*})$ be the transcript of a signature query in $\mathfrak{S} \setminus \mathfrak{S}_{\text{final}}$ with $w_{\mu^*} = w^*$. This transcript has not been revealed to the adversary since it is not the final transcript. Moreover, revealing the final transcript does not reduce the min-entropy of w_{μ^*} since it is independent from $(w_{\mu^*}, c_{\mu^*}, z_{\mu^*})$. In $G_{2.1}$, the distribution of w_{μ^*} has min-entropy α implying that $w^* = w_{\mu^*}$ occurs with probability $\leq 2^{-\alpha}$. Note that at most B transcripts are produced during the game. Therefore, each signature query of type $\mathfrak{S} \setminus \mathfrak{S}_{\text{final}}$ occurs with probability $\leq B \cdot 2^{-\alpha}$.

  The probability of Event_2 can be bounded as

$$\begin{aligned}
\mathbb{P}[\mathsf{Event}_2 \mid G_3^{\mathcal{A}}] \leq\ & (\varepsilon_{zk} + |p-\beta|) \cdot B \cdot Q_S \\
& + Q_S\Big((1+\varepsilon_{zk})(1-\gamma) + B \cdot 2^{-\alpha}\Big) \ .
\end{aligned}$$

Finally, using the union bound, one obtains

$$\begin{aligned}
|\mathbb{P}[1 \Leftarrow G_7^{\mathcal{A}}] - \mathbb{P}[1 \Leftarrow G_6^{\mathcal{A}}]| &\leq \mathbb{P}[\mathsf{Event}_1 \mid G_6^{\mathcal{A}}] + \mathbb{P}[\mathsf{Event}_2 \mid G_6^{\mathcal{A}}] \\
&\leq \mathsf{Adv}_{\Sigma}^{\mathsf{CUR}}(\mathcal{C}) + (\varepsilon_{zk} + |p-\beta|) \cdot B \cdot Q_S \\
&\quad + Q_S\Big((1+\varepsilon_{zk})(1-\gamma) + B \cdot 2^{-\alpha}\Big) \ .
\end{aligned}$$

The rest of the proof is similar to that of Theorem 17. □

Conclusion

In this section, we provide a wider perspective for our contributions and discuss future directions.

Oblivious sampling

In Chapter 2, we proved Theorem 1 by constructing an algorithm, that given a uniformly sampled matrix $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, obliviously generates an $\mathsf{LWE}_{m,n,q,\vartheta_{\sigma,q}}$ instance $(\mathbf{A}, \mathbf{b})$. The algorithm uses a subroutine to unambiguously distinguish the following coordinate states:

$$\forall j \in \mathbb{Z}/q\mathbb{Z}, \quad |\psi_j\rangle := \sum_{e=0}^{q-1} f(e)\,|j + e \bmod q\rangle \ ,$$

where f is proportional to $\sqrt{\vartheta_{\sigma,q}} \cdot u$ and u is the sign function. The subroutine is used to extract linear equations from the following state:

$$\sum_{\mathbf{s}} |\mathbf{s}\rangle \left(\bigotimes_{i\leq m} \left|\psi_{\langle \mathbf{a}_i,\mathbf{s}\rangle}\right\rangle \right) \ .$$

Then the superposition of secrets $\mathbf{s}$ in the first register is uncomputed by coherently applying Gaussian elimination, under the condition that sufficiently many linear equations are extracted above. The success probability of the algorithm depends on the quantity $q \cdot \min_x |\widehat{f}(x)|^2$, i.e, the success probability of the subroutine. The larger this quantity is, the more linear equations can be obtained. The sign function u as the local phases is used for increasing this quantity. This choice gives $\min_x |\widehat{f}(x)|^2 = 1/(q\sigma)$. It allows achieving the relaxed condition $m \geq n\sigma \cdot \omega(\log \lambda)$ compared to the exponential lower bound when one is not considering u. We note that the choice of the sign function may not be optimal. We believe that it is an intriguing question to search for the optimal choice of u.

Question 1. *Does there exist a choice of u, that is efficiently implementable in $\log q$, such that $\min_x |\widehat{f}(x)|^2 = \Omega(1/q)$?*

A positive answer to this question would relax the condition of Theorem 1 to $m \geq \Omega(n) \cdot \omega(\log \lambda)$. This would allow obtaining more flexible samplers under possibly weaker assumptions. For instance, it would imply an oblivious sampler for $q = 2^\lambda, \sigma = 2^{\sqrt{\lambda}}$, and $n = \lambda^2$, under the hardness of LWE with the same parameters. With the current version of Theorem 1, such a sampler can be achieved via the modulus-switching reduction,

under the hardness of LWE with parameters $q = 2^{\lambda-\sqrt{\lambda}} \cdot \mathsf{poly}(\lambda), \sigma = \mathsf{poly}(\lambda)$, and $n = \lambda^2$. When the noise-to-modulus ratio σ/q is fixed, the hardness of LWE can be roughly measured by the quantity $n \log q$ (see, e.g, [BLP+13]). Therefore, the latter assumption is supposedly stronger. We also believe that studying a wider class of phase functions can shed more light on our understanding of the hardness of the LWE problem. For instance, the author of [Che24] aims at solving the LWE problem in polynomial time, using the phase function $u(x) := \exp(-i\pi x^2/\tau^2)$. Although this approach fails due to a technical error, we believe that such phases deserve further exploration.

It seems that the algorithm for Theorem 1 does not require any special property of the matrix $\mathbf{A}$. In fact, it is not necessary for $\mathbf{A}$ to be uniform. An interesting open question is to devise oblivious samplers for the bounded distance decoding problem. Informally speaking, this problem asks, given a basis for a lattice L and a vector $\mathbf{t} \in \mathbb{R}^m$ that is guaranteed to be close to L, to find the closest point in L to $\mathbf{t}$. The LWE problem can be viewed as a distributional variant of the bounded distance decoding problem (see, e.g., the discussions in [GPV08, SSTX09]) over the following lattice.

$$L_q(\mathbf{A}) := \{\mathbf{As} \bmod q \mid \mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n\} + q\mathbb{Z}^m \ .$$

We leave the following problem for future work.

Question 2. *Which classes of lattices admit oblivious bounded distance decoding samplers, under the hardness of the generated instance?*

In the Common Reference String (CRS) model, in the beginning of the cryptographic protocols, all parties obtain a string sampled from a fixed distribution by a trusted authority. The distribution may have a specific structure such as that of the LWE problem, which in this case the trusted authority knows the underlying secret. A malicious authority might later use this information to attack the protocol. For this reason, when it is possible, one would rather rely on unstructured strings, i.e., the uniform distribution. This is considered to be a relatively weaker-to-achieve model. The CRS model has applications in designing Non-Interactive Zero-Knowledge (NIZK) proofs [DSMP88], for which numerous constructions were found based on various structured strings [BFM88, GO94, FLS99, GOS12, PS19]. When a classical oblivious sampler $\mathcal{C}$ for a structured distribution is available, it can be generically used to transform the structured CRS to the uniform one. The uniform CRS is the randomness used by $\mathcal{C}$ for generating the structured sample. Since the circuit description of $\mathcal{C}$ is publicly available, anyone can compute the structured string using the description and the randomness. A particularly interesting case is the NIZK proof of [PS19] for all NP languages. Their structured CRS has the shape of an LWE instance. If a classical oblivious LWE sampler was available, it could be used to transform their structured setting to a uniform setting. We are not aware of such a sampler. On the other hand, our quantum oblivious sampler cannot be used for this purpose. The reason is that our sampler is purely quantum and its randomness is inherently induced by the quantum measurements. Therefore, it is not possible to run it twice and obtain the same string (unless with negligible probability). We believe that studying mixed classical/quantum oblivious samplers would provide more perspective in this regard.

Question 3. *Does there exist a quantum* LWE *sampler that upon receiving a uniformly sampled string r, outputs an oblivious* LWE *instance $(\mathbf{A}_r, \mathbf{b}_r)$?*

Here, obliviousness requires that no extractor can find the witness even by having r. A positive answer to this question would provide a deterministic quantum algorithm when r is fixed, which can be used for the above purpose.

Analysis of FSwA

In Chapter 3, we provided detailed security analyses of FSwBA and FSwUA. In particular, by combining Theorem 16 with any of our results for FSwBA, one obtains a reduction from the UF-CMA security to the UF-NMA security in the unbounded regime. In the QROM, the adaptive-reprogramming approach of Theorem 14 yields the most optimized composition with respect to the reduction loss and runtime. With QRACM, up to a constant factor, its reduction loss is bounded from above by roughly

$$Q_S\beta^B + \frac{\beta^B 2^{-\alpha}}{(1-\beta)^3} + 2^{-\frac{\alpha}{2}} B Q_S Q_H^{\frac{1}{2}} + B Q_S \varepsilon_{zk} \ ,$$

where Q_S is the number of sign queries, Q_H is the number of hash queries, B is the upper bound on the number of repetitions in the signing algorithm, β is the probability of rejection in the underlying identification protocol, α is the min-entropy of the first message in the identification protocol, and ε_{zk} is the error for the zero-knowledge simulator. The runtime overhead of this reduction is BQ_H up to a constant factor. In the statistical zero-knowledge setting, the parameters are typically chosen such that $\varepsilon_{zk} = 2^{-\Omega(\lambda)}$. Moreover, Q_H is typically orders of magnitude larger than Q_S in practice, since hash evaluations can be made without restriction whereas sign queries require interaction with the signer. Therefore, the term $BQ_S\varepsilon_{zk}$ is relatively small compared to the others and can be ignored for the sake of simplicity. When B is chosen as large as the security parameter λ, the first two terms can also be ignored due to their small contribution in the loss. With these considerations, the leading term in the reduction loss is $2^{-\alpha/2}\lambda Q_S Q_H^{1/2}$, and the runtime overhead is λQ_H. An interesting question is to study the tightness of this reduction with respect to the runtime overhead and the security loss.

Question 4. *How tight is the provided* UF-CMA*-to-*UF-NMA *security reduction for FSwUA signatures?*

In particular, the runtime overhead is caused by the number of times one reprograms the random oracle. Can we use a different technique so that only one reprogramming per sign query suffices as in the ROM? In other words, can we decrease the runtime overhead from λQ_H to λQ_S? If not, does there exist a matching attack that distinguishes the UF-CMA game from the UF-NMA game, in the unbounded regime? It is known that in the adaptive reprogramming lemma, i.e, Lemma 31, the upper bound is tight up to a constant for all adversaries running in time polynomial in q [GHHM21, Theorem 7]. We note that the above security loss is dominated by the upper bound obtained by this lemma. Therefore, we believe that the security loss is tight, however, it is not clear to us how to extend the tightness of Lemma 31 to our setting.

In the statistical zero-knowledge setting, our reduction uses the HVZK property of the underlying identification protocol. As discussed earlier, it is a stronger notion compared to a simulator that only fakes the non-aborting transcripts. Whether it is possible to replace the HVZK simulator with the weaker one remains an intriguing open question.

Question 5. *Does there exist a* UF-CMA-*to*-UF-NMA *security reduction for FSwUA signatures with the same quality as above but only using the weaker zero-knowledge simulator?*

In [BBD+23, Theorem 2], the authors provide such a reduction. Their overall reduction loss is bounded from above by roughly

$$2^{-\frac{\alpha}{2}} Q_S Q_H^{\frac{1}{2}} + 2^{-\frac{\alpha}{2}} Q_H Q_S^{\frac{1}{2}} + Q_S \varepsilon_{zk} \ ,$$

with runtime overhead approximately being Q_S. With the same considerations as above, the leading term in their security loss is $2^{-\alpha/2} Q_H Q_S^{1/2}$. We note that their reduction has worse loss but better runtime overhead than ours. Is it possible to enhance their technique to obtain a smaller loss while keeping their runtime overhead? We leave this question for future work.

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.

[AD17] Martin R. Albrecht and Amit Deo. Large modulus ring-LWE $\geq$ module-LWE. In *ASIACRYPT*, 2017.

[AFLT16] M. Abdalla, P.-A. Fouque, V. Lyubashevsky, and M. Tibouchi. Tightly secure signatures from lossy identification schemes. *J. Cryptol.*, 2016.

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *ICALP*, 2011.

[AHU19] A. Ambainis, M. Hamburg, and D. Unruh. Quantum security proofs using semi-classical oracles. In *CRYPTO*, 2019.

[Ajt98] Miklós Ajtai. The shortest vector problem in ℓ_2 is NP-hard for randomized reductions (extended abstract). In *STOC*, 1998.

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, 2001.

[AR05] Dorit Aharonov and Oded Regev. Lattice problems in NP $\cap$ coNP. *J. ACM*, 2005.

[ASY22] S. Agrawal, D. Stehlé, and A. Yadav. Round-optimal lattice-based threshold signatures, revisited. In *ICALP*, 2022.

[Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Math. Ann.*, 1993.

[BBD+23] Manuel Barbosa, Gilles Barthe, Christian Doczkal, Jelle Don, Serge Fehr, Benjamin Grégoire, Yu-Hsuan Huang, Andreas Hülsing, Yi Lee, and Xiaodi Wu. Fixing and mechanizing the security proof of fiat-shamir with aborts and dilithium. In *CRYPTO*, 2023.

[BCI+13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In *TCC*, 2013.

[BD20] Zvika Brakerski and Nico Döttling. Hardness of LWE on general entropic distributions. In *EUROCRYPT*, 2020.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *ASIACRYPT*, 2011.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *STOC*, 1988.

[BGK15] Razvan Barbulescu, Pierrick Gaudry, and Thorsten Kleinjung. The Tower Number Field Sieve. In *ASIACRYPT*, 2015.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, 2012.

[BISW17] Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. Lattice-based SNARGs and their application to more efficient obfuscation. In *EUROCRYPT*, 2017.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *STOC*, 2013.

[BP14] Razvan Barbulescu and Cécile Pierrot. The multiple number field sieve for medium- and high-characteristic finite fields. *LMS J. Comput. Math.*, 2014.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS*, 1993.

[BS23] Dan Boneh and Victor Shoup. *A Graduate Course in Applied Cryptography.* 2023. Available at https://toc.cryptobook.us/book.pdf.

[CB98] Anthony Chefles and Stephen M. Barnett. Optimum unambiguous discrimination between linearly independent symmetric states. *Phys. Lett. A*, 1998.

[CDPR16] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In *EUROCRYPT*, 2016.

[Che24] Yilei Chen. Quantum algorithms for lattice problems. Cryptology ePrint Archive, Paper 2024/555, 2024.

[CKKK23] Heewon Chung, Dongwoo Kim, Jeong Han Kim, and Jiseung Kim. Amortized efficient zk-SNARK from linear-only RLWE encodings. *J. Comm. Netw.*, 2023.

[CLZ22] Yilei Chen, Qipeng Liu, and Mark Zhandry. Quantum algorithms for variants of average-case lattice problems via filtering. In *EUROCRYPT*, 2022.

[CN98] J.-Y. Cai and A. Nerurkar. Approximating the SVP to within a factor is NP-hard under randomized reductions. In *COCO*, 1998.

[CT23] André Chailloux and Jean-Pierre Tillich. The quantum decoding problem. Cryptology ePrint Archive, Paper 2023/1686, 2023.

[Dam91] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In *CRYPTO*, 1991.

[DFMS19] J. Don, S. Fehr, C. Majenz, and C. Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In *CRYPTO*, 2019.

[DFPS22] J. Devevey, O. Fawzi, A. Passelègue, and D. Stehlé. On rejection sampling in Lyubashevsky's signature scheme. In *ASIACRYPT*, 2022.

[DFPS23] Julien Devevey, Pouria Fallahpour, Alain Passelègue, and Damien Stehlé. A detailed analysis of Fiat-Shamir with aborts. In *CRYPTO*, 2023.

[DFS24] Thomas Debris-Alazard, Pouria Fallahpour, and Damien Stehlé. Quantum oblivious LWE sampling and insecurity of standard model lattice-based SNARKs. Cryptology ePrint Archive, Paper 2024/030, 2024.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 1976.

[DJ92] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proc. R. Soc. Lond. A*, 1992.

[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. *TCHES*, 2018.

[DRT21] Thomas Debris-Alazard, Maxime Remaud, and Jean-Pierre Tillich. Quantum reduction of finding short code vectors to the decoding problem. Cryptology ePrint Archive, Paper 2021/752, 2021.

[DSMP88] Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge proof systems. In *CRYPTO*, 1988.

[dW23] Ronald de Wolf. Quantum computing: Lecture notes. arXiv:1907.09415, 2023.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J, Comput.*, 1999.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO*, 1999.

[FS86] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, 1986.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In *EUROCRYPT*, 2013.

[GHHM21] A. B. Grilo, K. Hövelmanns, A. Hülsing, and C. Majenz. Tight adaptive reprogramming in the QROM. In *ASIACRYPT*, 2021.

[GKPV10] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In *ICS*, 2010.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *STOC*, 1982.

[GMNO18] Rosario Gennaro, Michele Minelli, Anca Nitulescu, and Michele Orrù. Lattice-based ZK-SNARKs from square span programs. In *CCS*, 2018.

[GNSV23] Chaya Ganesh, Anca Nitulescu, and Eduardo Soria-Vazquez. Rinocchio: SNARKs for ring arithmetic. *J. Cryptol.*, 2023.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptol.*, 1994.

[Gor93] Daniel Gordon. Discrete logarithms in GF(P) using the number field sieve. *SIAMDM*, 1993.

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 2012.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.

[GR02] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. arXiv:quant-ph/0208112, 2002.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *STOC*, 1996.

[HKM18] Gottfried Herold, Elena Kirshanova, and Alexander May. On the asymptotic complexity of solving LWE. *Des. Codes and Cryptogr.*, 2018.

[ISW21] Yuval Ishai, Hang Su, and David J. Wu. Shorter and faster post-quantum designated-verifier zkSNARKs from lattices. In *CCS*, 2021.

[JLSV06] Antoine Joux, Reynald Lercier, Nigel Smart, and Frederik Vercauteren. The number field sieve in the medium prime case. In *CRYPTO*, 2006.

[JS19] S. Jaques and J. M. Schanck. Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. In *CRYPTO*, 2019.

[Kat21] S. Katsumata. A new simple technique to bootstrap various lattice zero-knowledge proofs to QROM secure NIZKs. In *CRYPTO*, 2021.

[Kho03] Subhash Khot. Hardness of approximating the shortest vector problem in high ℓ_p norms. *JCSS FOCS 2003 Special Issue*, 2003.

[Kho05] Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *J. ACM*, 2005.

[KLS18] E. Kiltz, V. Lyubashevsky, and C. Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In *EUROCRYPT*, 2018.

[Lam10] Leslie Lamport. Constructing digital signatures from a one way function. *IEEE HICSS*, 2010.

[LLL82] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 1982.

[LMSV12] Jake Loftus, Alexander May, Nigel P. Smart, and Frederik Vercauteren. On CCA-secure somewhat homomorphic encryption. In *SAC*, 2012.

[LMZ23] Jiahui Liu, Hart Montgomery, and Mark Zhandry. Another round of breaking and making quantum money: How to not build it from lattices, and more. In *EUROCRYPT*, 2023.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, 2010.

[LS15] Adeline Langlois and Damien 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 *EUROCRYPT*, 2014.

[Lyu09] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, 2009.

[Lyu12] V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, 2012.

[Lyu16] V. Lyubashevsky. Digital signatures based on the hardness of ideal lattice problems in all rings. In *ASIACRYPT*, 2016.

[LZ19] Q. Liu and M. Zhandry. Revisiting post-quantum Fiat-Shamir. In *CRYPTO*, 2019.

[Mat03] D. V. Matyukhin. On asymptotic complexity of computing discrete logarithms over GF(p). *Discrete Math. Appl.*, 2003.

[Mic98] Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. FOCS, 1998.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM J. Comput.*, 2000.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In *CRYPTO*, 2011.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2011.

[NYI+20] Ken Naganuma, Masayuki Yoshino, Atsuo Inoue, Yukinori Matsuoka, Mineaki Okazaki, and Noboru Kunihiro. Post-quantum zk-SNARK for arithmetic circuits using QAPs. In *AsiaJCIS*, 2020.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, 2009.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In *CRYPTO*, 2019.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 2009.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 1978.

[Sch87] C.P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 1987.

[Sha49] C. E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 1949.

[Sho94] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *FOCS*, 1994.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 1997.

[SSEK22] Ron Steinfeld, Amin Sakzad, Muhammed F. Esgin, and Veronika Kuchta. Private re-randomization for module LWE and applications to quasi-optimal ZK-SNARKs. Cryptology ePrint Archive, Paper 2022/1690, 2022.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In *ASIACRYPT*, 2009.

[Unr17] Dominique Unruh. Post-quantum security of Fiat-Shamir. In *ASIACRYPT*, 2017.

[vEH14] T. van Erven and P. Harremoës. Rényi divergence and Kullback-Leibler divergence. *IEEE Trans. Inf. Theory*, 2014.

[YZ21] Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. In *EUROCRYPT*, 2021.

[Zha12a] M. Zhandry. How to construct quantum random functions. In *FOCS*, 2012.

[Zha12b] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In *CRYPTO*, 2012.

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *CRYPTO*, 2019.

Appendix

Positive operator-valued measures

Positive Operator-Valued Measures (POVM) are defined as follows. They are the most general measurements allowed within quantum information theory.

Definition 38 (POVM measurements)**.** *A* POVM *is a set* $\{\mathbf{E}_i\}_{i\in\mathcal{I}}$ *of positive operators where* $\mathcal{I}$ *is the set of measurement outcomes and the operators satisfy* $\sum_i \mathbf{E}_i = \mathbf{Id}$*. A measurement upon a quantum state* $|\psi\rangle$ *outputs* i *with probability* $\langle\psi|\,\mathbf{E}_i\,|\psi\rangle$*.*

POVMs are sometimes considered in the following situation: given a set of quantum states $|\psi_1\rangle, \ldots, |\psi_N\rangle$, devise a POVM that when applied over $|\psi_j\rangle$, it either outputs the correct index j or some special symbol $\perp$ representing the "unknown" answer. In other words, the measurement never makes an error when it succeeds to identify the prepared state and we say that it *unambiguously* distinguishes the states $|\psi_j\rangle$'s. The probability of error is defined as the probability that the measurement outputs $\perp$, when it is maximized over all possible input states:

$$p_\perp := \max_k \langle\psi_k|\,\mathbf{E}_\perp\,|\psi_k\rangle$$

where $\mathbf{E}_\perp$ corresponds to the outcome $\perp$.

Discrimination of coordinate states

We now describe the POVM from [CB98], which is known to be optimal to unambiguously distinguish the $|\psi_k\rangle$'s (as given in Definition 19). Namely, it minimizes the error parameter $p_\perp$ over all possible choice of POVMs. This optimality is enabled by the fact that the $|\psi_k\rangle$'s verify the following "symmetry" condition:

$$\forall k \in \mathbf{Z}/q\mathbb{Z}, \quad \mathbf{T}\,|\psi_k\rangle = |\psi_{k+1 \bmod q}\rangle \ ,$$

where $\mathbf{T}$ denotes the translation operator, i.e., $\mathbf{T}\,|a\rangle = |a+1 \bmod q\rangle$ for all a, and from the fact that they are linearly independent (which is ensured by $\widehat{f}(x) \neq 0$ for all x, as is the case for our instantiation with the folded Gaussian distribution).

Theorem 19 (Adapted from [CB98])**.** *Let* q *be an integer and* $f : \mathbb{Z}/q\mathbb{Z} \to \mathbb{C}$ *be an amplitude function such that* $\widehat{f}\,(y) \neq 0$ *for every* $y \in \mathbb{Z}/q\mathbb{Z}$*. Let*

$$\left|\psi_j^\perp\right\rangle := \frac{1}{\sqrt{N}} \sum_{y\in\mathbb{Z}/q\mathbb{Z}} \overline{\widehat{f}(-y)^{-1}}\ \omega_q^{-jy}\,|\chi_y\rangle\,, \ \textit{where}\ \ N := \sum_{y\in\mathbb{Z}/q\mathbb{Z}}^{q-1} |\widehat{f}(y)|^{-2}\ , \tag{3.1}$$

and

$$\forall j \in \mathbb{Z}/q\mathbb{Z}, \quad \mathbf{E}_j := \frac{1}{\lambda_+} \left|\psi_j^\perp\right\rangle\!\left\langle\psi_j^\perp\right|, \text{ and } \mathbf{E}_\perp := \mathbf{I} - \sum_{j \in \mathbb{Z}/q\mathbb{Z}} \mathbf{E}_j \ ,$$

where λ_+ *is the maximum eigenvalue of* $\sum_{j \in \mathbb{Z}/q\mathbb{Z}} \left|\psi_j^\perp\right\rangle\!\left\langle\psi_j^\perp\right|$. *Then the set* $\{\mathbf{E}_j\}_{j \in (\mathbb{Z}/q\mathbb{Z}) \cup \{\perp\}}$ *is a* POVM *that unambiguously distinguishes the coordinate states with success probability* p *as follows (it is independent of* j*):*

$$p = \langle\psi_j| \mathbf{E}_j |\psi_j\rangle = q \cdot \min_{y \in \mathbb{Z}/q\mathbb{Z}} \left|\widehat{f}(y)\right|^2 \ .$$

Representating the coordinate states in the Fourier basis is helpful to approach the problem. The first lemma shows that $\left|\psi_i^\perp\right\rangle$ defined as in Equation (3.1) is a quantum state orthogonal to all $|\psi_j\rangle$'s where $i \neq j$.

Lemma 33. *Using the notations of Theorem 19, we have:*

$$\forall i, j \in \mathbb{Z}/q\mathbb{Z}, \quad \left\langle\psi_i^\perp\middle|\psi_j\right\rangle = \begin{cases} \frac{q}{\sqrt{N}} & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \ .$$

Proof. Let us write the $|\psi_j\rangle$'s in the Fourier basis. We have for all $j \in \mathbb{Z}/q\mathbb{Z}$:

$$\begin{aligned} |\psi_j\rangle &= \sum_{e \in \mathbb{Z}/q\mathbb{Z}} f(e) |j + e \bmod q\rangle \\ &= \frac{1}{\sqrt{q}} \sum_{e \in \mathbb{Z}/q\mathbb{Z}} f(e) \sum_{x \in \mathbb{Z}/q\mathbb{Z}} \omega_q^{-(j+e)x} |\chi_x\rangle \quad \text{(by Lemma 1)} \\ &= \sum_{x \in \mathbb{Z}/q\mathbb{Z}} \left(\frac{1}{\sqrt{q}} \sum_{e \in \mathbb{Z}/q\mathbb{Z}} f(e) \omega_q^{-xe} \right) \omega_q^{-jx} |\chi_x\rangle \\ &= \sum_{x \in \mathbb{Z}/q\mathbb{Z}} \widehat{f}(-x) \omega_q^{-jx} |\chi_x\rangle \ . \end{aligned}$$

We thus have, for all $i \in \mathbb{Z}/q\mathbb{Z}$:

$$\left\langle\psi_i^\perp\middle|\psi_j\right\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}/q\mathbb{Z}} \omega_q^{x(i-j)} = \begin{cases} \frac{q}{\sqrt{N}} & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \ .$$

This completes the proof. □

We now consider the maximum eigenvalue λ_+ of $\sum_{j \in \mathbb{Z}/q\mathbb{Z}} \left|\psi_j^\perp\right\rangle\!\left\langle\psi_j^\perp\right|$.

Lemma 34. *Using notations of Theorem 19, we have:*

$$\lambda_+ = \frac{q}{N} \frac{1}{\min_{x \in \mathbb{Z}/q\mathbb{Z}} \left|\widehat{f}(x)\right|^2} \ .$$

Proof. We have the following equalities:

$$\begin{aligned}\sum_{j\in\mathbb{Z}/q\mathbb{Z}} \left|\psi_j^{\perp}\right\rangle\!\left\langle\psi_j^{\perp}\right| &= \frac{1}{N}\sum_{j\in\mathbb{Z}/q\mathbb{Z}}\left(\sum_{x\in\mathbb{Z}/q\mathbb{Z}} \overline{\widehat{f}(-x)^{-1}}\,\omega_q^{-jx}\,|\chi_x\rangle\right)\left(\sum_{y\in\mathbb{Z}/q\mathbb{Z}} \widehat{f}(-y)^{-1}\,\omega_q^{jy}\,\langle\chi_y|\right)\\ &= \frac{1}{N}\sum_{x,y\in\mathbb{Z}/q\mathbb{Z}}\left(\sum_{j\in\mathbb{Z}/q\mathbb{Z}} w_q^{j(y-x)}\right)\,\overline{\widehat{f}(-x)^{-1}}\,\widehat{f}(-y)^{-1}\;|\chi_x\rangle\!\langle\chi_y|\\ &= \frac{q}{N}\sum_{x\in\mathbb{Z}/q\mathbb{Z}} |\widehat{f}(-x)|^{-2}\,|\chi_x\rangle\!\langle\chi_x|\;.\end{aligned}$$

Therefore, as the $|\chi_x\rangle$'s define an orthonormal basis of the underlying Hilbert space, we obtain

$$\lambda_+ = \frac{q}{N}\,\frac{1}{\min\limits_{x\in\mathbb{Z}/q\mathbb{Z}}\left|\widehat{f}\,(x)\right|^2}\;.$$

This completes the proof. □

Proof of Theorem 19. The fact that $\{\mathbf{E}_j\}_{j\in(\mathbb{Z}/q\mathbb{Z})\cup\{\perp\}}$ defines a POVM follows from the definition of λ_+: they are positive operators and sum to the identity.

By Lemma 33, the state $\left|\psi_i^{\perp}\right\rangle$ is orthogonal to $|\psi_j\rangle$ for all $j\neq i$. Therefore, given $|\psi_j\rangle$, the probability to successfully measure j with the POVM $\{\mathbf{E}_i\}_{i\in(\mathbb{Z}/q\mathbb{Z})\cup\{\perp\}}$ is given by

$$p = \langle\psi_j|\,\mathbf{E}_j\,|\psi_j\rangle = \frac{1}{\lambda_+}\left|\left\langle\psi_j^{\perp}\middle|\psi_j\right\rangle\right|^2 = \frac{q^2}{\lambda_+\,N} = q\cdot\min_{y\in\mathbb{Z}/q\mathbb{Z}}\left|\widehat{f}\,(y)\right|^2,$$

where the two last equalities follow from Lemmas 33 and 34. □