Quantum Key-Revocable Dual-Regev Encryption, Revisited

Prabhanjan Ananth* UCSB
Zihan Hu† Shanghai Qi Zhi Institute
Zikuan Huang‡ Tsinghua University

Abstract

Quantum information can be used to achieve novel cryptographic primitives that are impossible to achieve classically. A recent work by Ananth, Poremba, Vaikuntanathan (TCC 2023) focuses on equipping the dual-Regev encryption scheme, introduced by Gentry, Peikert, Vaikuntanathan (STOC 2008), with key revocation capabilities using quantum information. They further showed that the key-revocable dual-Regev scheme implies the existence of fully homomorphic encryption and pseudorandom functions, with both of them also equipped with key revocation capabilities. Unfortunately, they were only able to prove the security of their schemes based on new conjectures and left open the problem of basing the security of key revocable dual-Regev encryption on well-studied assumptions.

In this work, we resolve this open problem. Assuming polynomial hardness of learning with errors (over sub-exponential modulus), we show that key-revocable dual-Regev encryption is secure. As a consequence, for the first time, we achieve the following results:

- Key-revocable public-key encryption and key-revocable fully-homomorphic encryption satisfying classical revocation security and based on polynomial hardness of learning with errors. Prior works either did not achieve classical revocation or were based on sub-exponential hardness of learning with errors.
- Key-revocable pseudorandom functions satisfying classical revocation from the polynomial hardness of learning with errors. Prior works relied upon unproven conjectures.

1 Introduction

Leveraging fundamental principles of quantum information to achieve cryptographic notions, that are otherwise impossible to achieve classically, is an exciting research direction. In the past few years, a dizzying variety of quantum cryptographic primitives, termed as *unclonable* primitives, have been studied. Underlying the unclonable primitives is the no-cloning principle of quantum mechanics [WZ82, Die82] which states that quantum states, unlike classical strings, cannot be copied. The recent surge in the development of unclonable primitives has resulted in innovative approaches to tackle many real-world security challenges, including protection against anti-piracy [Aar09], privacy concerns in blockchain technology [AGKZ20], and provable deletion of cryptographic data from the web [BI20, BL20].

We focus on the task of securely leasing or revoking cryptographic keys using the tools of quantum information. Before precisely stating the problem that we set out to address, let us consider

*prabhanjan@cs.ucsb.edu
†huzihan423@gmail.com
‡hzk21@mails.tsinghua.edu.cn

two scenarios: (a) Imagine a manager needing to temporarily delegate their duties, including access to sensitive encrypted data, to their subordinate by sharing cryptographic keys. The challenge is ensuring the subordinate's access is revoked upon the manager's return, a task that is impossible to achieve with classical keys, (b) If a cryptographic key is stolen from a device, unless the attacker has left a trace, it becomes challenging to detect such an attack and report it.

Quantum information presents a unique approach of tackling both of the above aforementioned problems.

Our Focus. A major focus of our work is on protecting decryption keys. Specifically, we focus on the popular dual-Regev public-key encryption scheme of [GPV08] (also, referred to as the GPV encryption scheme), which has inspired the design of many lattice-based cryptographic primitives [BGG+14, Mah18, BDGM20, Qua20]. A key-revocable dual-Regev public-key encryption scheme, first introduced in [APV23], is the same as the dual-Regev scheme except that we have the additional guarantee that the decryption keys can alternately be represented as quantum states. Any user in possession of the quantum decryption key can decrypt ciphertexts just the way he would have been able to do if he had a classical decryption key. The security guarantee stipulates that once the user returns the quantum decryption key, they will lose the ability to decrypt ciphertexts and in particular, we require that the semantic security of dual-Regev encryption still hold. We refer the reader to Section 1.1 for a more detailed description of the key-revocable dual Regev public-key encryption scheme.

Key-Revocable Security of Dual-Regev: Motivation. Proving the security of key-revocable dual-Regev encryption could lead to adding key revocation capabilities to other cryptographic primitives. Indeed, [APV23] showed that key-revocable dual-Regev encryption can be leveraged to prove the existence of fully homomorphic encryption and pseudorandom functions equipped with key revocation capabilities. The structure of dual-Regev encryption was crucially exploited in these applications.

There is also an aesthetic reason behind studying this problem. Dual-Regev public-key encryption is an elegant construction that is taught in most graduate classes on lattice-based cryptography. Understanding whether it satisfies key-revocable security is a natural theoretical question.

The work of [APV23] attempted to prove the key-revocable security of dual-Regev encryption. Unfortunately, they were only able to prove the security of this construction based on a new unfounded conjecture. They leave the problem of proving the key-revocable security of dual Regev encryption on concrete computational assumptions as an important open problem. In this same work, inspired by the literature on certified deletion [BI20, HMNY21, BK22], they define a stronger property called classical revocation: instead of the user being asked to return the state, they are only asked to return a classical string that certifies that the quantum decryption key has been deleted. After the state has been deleted, as before, we require the semantic security of dual-Regev encryption to still hold. [APV23] relied upon yet another new conjecture to show that dual-Regev encryption satisfied classical key-revocation security. The reliance on both these conjectures makes the current state of affairs rather unsatisfactory. [APV23] left open the problem of basing key-revocation security of dual-Regev encryption on well-studied cryptographic assumptions.

Main Result. In this work, we resolve this open problem. We show the following:

Theorem 1.1. *Assuming polynomial hardness of learning with errors over sub-exponential modu-*

lus[1]*, dual Regev encryption is key-revocable. Moreover, this scheme satisfies the classical revocation property.*

Applications. By combining the above theorem with the applications of key-revocable dual-Regev encryption in [APV23], we obtain the following results:

Main Application: We present the first result of *key-revocable pseudorandom functions* based on the polynomial hardness of learning with errors and also simultaneously satisfies classical revocation property. Prior work by [APV23] relied upon unproven conjectures.

Other Applications: We also achieve other applications that are in some aspects better than the previous works.

1. We present the first result of *key-revocable public-key encryption* that is based on polynomial hardness of learning with errors and simultaneously satisfies classical revocation property. Prior works by [AKN+23, CGJL23] satisfied one but not the other.

2. We present the first result of *key-revocable fully homomorphic encryption* that is based on polynomial hardness of learning with errors and simultaneously satisfies classical revocation property. Prior work by [CGJL23] achieved this result from sub-exponential hardness of learning with errors.

Main Technical Contribution: At the heart of our result is a new search-to-decision reduction that reduces a quantum distinguisher that breaks the semantic security of dual-Regev encryption into a quantum adversary that can solve an inhomogeneous short integer solution (ISIS) problem. Our search-to-decision reduction is qualitatively different from [APV23] who rely upon Goldreich-Levin reduction over large finite fields. In addition to the fact that [APV23] relies upon a conjecture, their reduction necessarily[2] incurs a loss that is inversely proportional to q, where q is the size of the field. Since they need to set q to be sub-exponential in the security parameter, this means that their reduction suffers from sub-exponential loss. On the other hand, our ISIS solver only incurs inverse polynomial loss, independent of q.

Related Works: It would be remiss not to discuss two other related prior works.

Chardouvelis, Goyal, Jain, Liu [CGJL23] present instantiations of key-revocable public-key encryption and fully homomorphic encryption. Moreover, their schemes satisfy classical key-revocation security[3]. *There are two advantages of our work over theirs:*

- *They do not have any results on pseudorandom functions,*
- *They assume sub-exponential hardness of learning with errors whereas we only assume polynomial hardness of learning with errors.*

[1] By aggressively setting the parameters, it would suffice to just assume polynomial hardness of learning with errors over quasi-polynomial modulus.

[2] Their starting point is the classical Goldreich Levin reduction over finite fields by Dodis et al. [DGT+10]. This reduction already suffers from a loss that is inversely proportional to q.

[3] In fact, they satisfy a much stronger property where the communication with the user can be completely classical.

Agrawal, Kitagawa, Nishimaki, Yamada, Yamakawa [AKN+23] present an instantiation of key-revocable public-key encryption based on the existence of any post-quantum secure public-key encryption scheme. They also present other key-revocable notions, such as functional encryption, that are not covered in this work. *There are two advantages of our work over theirs:*

- *They do not prove the classical key revocation security of their scheme,*
- *They also do not provide any positive results on either fully homomorphic encryption or pseudorandom functions.*

Both the works, [CGJL23] and [AKN+23], come up with arguably more involved constructions of key-revocable public-key encryption which make it unwieldy to extend their techniques to get new applications.

1.1 Technical Overview

In this section, we give an overview of the main ideas and techniques underlying our proofs.

Key-Revocable Dual-Regev Public-Key Encryption. We first recall the key-revocable dual-Regev constructions from [APV23]. This part has been reproduced verbatim from their work.

- $\mathsf{KeyGen}(1^\lambda)$: Sample a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a short trapdoor basis $\mathsf{td}_{\mathbf{A}}$ for it. The (quantum) decryption key is a Gaussian superposition of ISIS solutions, which is generated by the following procedure: Create a Gaussian superposition of short vectors $\mathbf{x}$, compute the image $\mathbf{A}\cdot\mathbf{x} \pmod q$ in the second register to get

$$|\psi\rangle = \sum_{\mathbf{x}\in\mathbb{Z}_q^m} \rho_\sigma(\mathbf{x})\,|\mathbf{x}\rangle \otimes |\mathbf{A}\cdot\mathbf{x} \pmod q\rangle$$

  where $\rho_\sigma(\mathbf{x}) = \exp(-\pi\|\mathbf{x}\|^2/\sigma^2)$ is the Gaussian measure, for some $\sigma > 0$, and measure the second register to the Gaussian coset state

$$|\psi_{\mathbf{y}}\rangle = \sum_{\substack{\mathbf{x}\in\mathbb{Z}_q^m \\ \mathbf{A}\mathbf{x}=\mathbf{y} \pmod q}} \rho_\sigma(\mathbf{x})\,|\mathbf{x}\rangle$$

  for some measurement outcome $\mathbf{y} \in \mathbb{Z}_q^n$.

  Finally we set $\mathsf{PK} = (\mathbf{A}, \mathbf{y})$, $\mathsf{MSK} = \mathsf{td}_{\mathbf{A}}$ and $\rho_{\mathsf{SK}} = |\psi_{\mathbf{y}}\rangle$.

- $\mathsf{Enc}(\mathsf{PK}, \mu)$: To encrypt a bit $\mu \in \{0,1\}$, sample a random string $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ together with discrete Gaussian errors $\mathbf{e} \in \mathbb{Z}^m$ and $e' \in \mathbb{Z}$, and output a classical ciphertext CT given by

$$\mathsf{CT} = \left(\mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal, \mathbf{s}^\intercal\mathbf{y} + e' + \mu\cdot\lfloor\frac{q}{2}\rfloor\right) \in \mathbb{Z}_q^m \times \mathbb{Z}_q.$$

- $\mathsf{Dec}(\rho_{\mathsf{SK}}, \mathsf{CT})$: First apply the unitary $U : |\mathbf{x}\rangle\,|0\rangle \to |\mathbf{x}\rangle\,|\mathsf{CT}\cdot(-\mathbf{x}, 1)^\intercal\rangle$ on input $\rho_{\mathsf{SK}} \otimes |0\rangle\langle 0|$, and then measure the second register in the computational basis. Because ρ_{SK} is supposed to be the Gaussian coset state $|\psi_{\mathbf{y}}\rangle$, which is a superposition of short vector $\mathbf{x}$ subject $\mathbf{A}\cdot\mathbf{x} = \mathbf{y}$, we obtain an approximation of $\mu\cdot\lfloor\frac{q}{2}\rfloor$ from which we can recover μ.

- $\mathsf{Revoke}(\mathsf{PK}, \mathsf{MSK}, \rho)$: Apply the projective measurement $\{|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}|, I - |\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}|\}$ onto ρ using the master secret key $\mathsf{td}_{\mathbf{A}}$[4]. Output Valid if the measurement succeeds, and output $\mathsf{Invalid}$, otherwise.

Consider an efficient adversary $\mathcal{A}$. It receives as input a state $|\psi_{\mathbf{y}}\rangle$ from the challenger and computes a state $\rho_{\mathrm{R,AUX}}$ on two registers R and AUX. Subsequently, the adversary returns system R to the challenger, while retaining system AUX as quantum advice for subsequent steps. Informally speaking, we say that the above scheme is secure if $\mathcal{A}$ wins both of the following events simultaneously only with negligible probability:

- Revoke on the system R outputs Valid.
- Using AUX, $\mathcal{A}$ can distinguish $\left(\mathbf{s}^{\intercal}\mathbf{A} + \mathbf{e}^{\intercal}, \mathbf{s}^{\intercal}\mathbf{y} + e' + \lfloor \frac{q}{2} \rfloor\right)$ versus $(\mathbf{s}^{\intercal}\mathbf{A} + \mathbf{e}^{\intercal}, \mathbf{s}^{\intercal}\mathbf{y} + e')$

Starting Point. Inspired by [APV23], we undertake the following approach. Suppose there did exist an efficient adversary $\mathcal{A}$ that is successful in violating the security of the above construction. We reduce $\mathcal{A}$ into an SIS solver $\mathcal{B}$, which is described as follows: it first runs $\mathcal{A}$ on input $(\mathbf{A}, \mathbf{y}, |\psi_{\mathbf{y}}\rangle)$ to obtain a state ρ on two registers R and AUX. Then, $\mathcal{B}$ needs to be cleverly designed in such a way that it recovers a short vector $\mathbf{x_0}$ from R and a short vector $\mathbf{x_1}$ from AUX satisfying the following properties:

- $\mathbf{A}\mathbf{x_0} = \mathbf{y}$, $\mathbf{A}\mathbf{x_1} = \mathbf{y}$ and,
- $\mathbf{x_0} \neq \mathbf{x_1}$.

Once both the vectors $\mathbf{x_0}$ and $\mathbf{x_1}$ are recovered then it simply sets the SIS solution to be $\mathbf{x_0} - \mathbf{x_1}$.

While [APV23] set out on this route, they only managed to show such a reduction based on a new conjecture. The core reason behind this is the fact that it is challenging to be able to *simultaneously* recover two *distinct* short solutions from *two potentially entangled* registers R and AUX. An attempt to recover $\mathbf{x_0}$ from R could invariably disturb the part of the state on AUX such that it is no longer possible to recover $\mathbf{x_1}$. Any approach we undertake should tackle this challenge.

Our Approach. We propose a three-step approach to prove the security of key-revocable dual-Regev encryption based on learning with errors.

- STEP 1. In the first step, we transform the intermediate state ρ (on R and AUX) produced by $\mathcal{A}$ into a "good state" ρ_{good}. This step doesn't need to always succeed. We require two guarantees here: (a) this step aborts with probability bounded away from 1 and, (b) conditioned on not abort, the output of this step is a good state ρ_{good} such that the revocation on R succeeds with non-negligible probability and Step 2 works.
- STEP 2. Suppose the output of Step 1 is ρ_{good}. We require that as long as ρ_{good} is a good state then, from AUX, we should be able to recover a short vector $\mathbf{x_1}$ such that $\mathbf{A}\mathbf{x_1} = \mathbf{y}$. More importantly, we should be able to recover $\mathbf{x_1}$ with overwhelming probability.
- STEP 3. We recover a short vector $\mathbf{x_0}$ from the register R such that $\mathbf{A}\mathbf{x_0} = \mathbf{y}$. Our hope is that $\mathbf{x_0}$ and $\mathbf{x_1}$ are distinct and if this is the case then $\mathbf{x_0} - \mathbf{x_1}$ is a non-trivial short solution in the kernel of $\mathbf{A}$.

[4] [APV23] showed how to implement this projective measurement efficiently with the trapdoor $\mathsf{td}_{\mathbf{A}}$.

The easiest step to realize is Step 3. Suppose we have the guarantee that we can recover $\mathbf{x_1}$ from Aux with overwhelming probability. By invoking *almost as good as new* lemma (Lemma 2.1), we can show that the state ρ after Step 2 is not disturbed by much. This means that Revoke still succeeds on R with inverse polynomial probability. This further implies that measuring the register R yields a short vector $\mathbf{x_0}$. Then using a simplified analysis of [APV23], we can argue that $\mathbf{x_0} \neq \mathbf{x_1}$, completing the proof.

We focus our attention on implementing Steps 1 and 2. Our main technical contribution will lie in Step 2.

Implementing Step 1: To implement Step 1, we rely upon the threshold implementation technique introduced by Zhandry [Zha20]. Threshold implementation is a technique employed to get an estimate of the success probability of a POVM on a state. In our context, we employ this technique to test whether the adversary acting upon AUX register of ρ is successful in violating the security of key-revocable dual-Regev encryption scheme. Formally, we define the threshold implementation operator $\mathsf{TI}_{\frac{1}{2}+\gamma}$, where γ is some inverse polynomial, with the following properties:

1. $\mathsf{TI}_{\frac{1}{2}+\gamma}$ is akin to a projector-like operator, collapsing the state to a γ-good state ρ_{good} capable of distinguishing between $(\mathbf{s}^\intercal\mathbf{A}+\mathbf{e}^\intercal, \mathbf{s}^\intercal\mathbf{y}+e')$ and $(\mathbf{u}, r)$ for $\mathbf{u}, r$ being sampled uniformly randomly with probability 2γ (referred to as a "γ-good state") when $\mathsf{TI}_{\frac{1}{2}+\gamma}$ outputs 1, or to some other state when $\mathsf{TI}_{\frac{1}{2}+\gamma}$ outputs 0.

2. For a successful adversary, applying $\mathsf{TI}_{\frac{1}{2}+\gamma}$ with an inverse polynomial γ on ρ_{Aux} results in an output of 1 with noticeable probability.

3. Upon applying $\mathsf{TI}_{\frac{1}{2}+\gamma}$ again on a γ-good state, it yields an output of 1 with probability 1.

To summarize, as long as $\mathcal{A}$ is a successful adversary, $\mathsf{TI}_{\frac{1}{2}+\gamma}$ collapses ρ into a good state ρ_{good} with inverse polynomial probability.

Implementing Step 2: As mentioned earlier, implementing Step 2 is our main technical contribution.

It was already shown by [APV23] that $\mathbf{x_1}$ can be extracted from AUX. However, the success probability of their extraction mechanism was only inverse polynomial which is insufficient for our purpose. Instead, we completely depart from [APV23] and propose a novel extraction method. This high-level approach is inspired by [CGJL23] although they study for a completely different construction.

At a high level, our extractor proceeds by guessing each entry of $\mathbf{x_1}$, where $\mathbf{x_1}$ is a short solution mapping $\mathbf{A}$ to $\mathbf{y}$, one coordinate at a time. For each coordinate, we try all possible values and using the distinguisher, test which of our guesses was correct. Recall that there are exponentially many short vectors that map $\mathbf{A}$ to $\mathbf{y}$. But once we apply the Gaussian collapsing lemma [Por22, LMZ23], we can replace the state $|\psi_{\mathbf{y}}\rangle$ with $|\mathbf{x_1}\rangle$. While recovering, say, the i^{th} coordinate of $\mathbf{x_1}$, we use the distinguisher on Aux to figure out whether the guess for the i^{th} coordinate was correct or not. However, this has to be handled with care. Since the distinguisher has quantum auxiliary advice, we cannot keep hoping to run the distinguisher again and again. After the first run, the state of the distinguisher could be damaged making it useless for future iterations. So we need to come up with a mechanism to check if a guess is correct or not while maintaining the quantum

state. Making crucial use of threshold implementation along with techniques from lattice-based cryptography, we show how to implement this.

Our extractor is described as follows:

1. Initialize $\mathbf{x} = \mathbf{0}$ as the output register.
2. For each position $i \in [m]$ and each guess g_i, we test whether the i-th entry $\mathbf{x}_1$ is g_i by:
   (a) Applying $\mathsf{TI}_{\frac{1}{2}+\gamma'}(i, g_i)$ on system $\textsc{Aux}$, where $\mathsf{TI}_{\frac{1}{2}+\gamma'}(i, g_i)$ is a threshold implementation that 'tests' whether the state is γ'-good at distinguishing between $\left(\mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal + c \cdot \hat{\mathbf{i}}, \mathbf{s}^\intercal\mathbf{y} + c \cdot g_i + e'\right)$ (where $c \xleftarrow{\$} \mathbb{Z}_q$ and $\hat{\mathbf{i}}$ is the unit vector on the i-th dimension) and $(\mathbf{u}, r)$ (where $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m, r \xleftarrow{\$} \mathbb{Z}_q$).
   (b) If the output is 1, set $\mathbf{x}_i = g_i$.
   (c) If the output is 0, skip to the next iteration.
3. Output $\mathbf{x}$.

We argue that our extractor outputs $\mathbf{x}_1$ with nearly perfect probability if $\mathsf{TI}_{\frac{1}{2}+\gamma}$ on $\rho_{\textsc{Aux}}$ outputs 1. Zhandry [Zha20] demonstrates that for two threshold implementations concerning computationally indistinguishable tasks (e.g., distinguishing $(\mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal, \mathbf{s}^\intercal\mathbf{y} + e')$ from $(\mathbf{u}, r)$, and distinguishing $(\mathbf{u}, \mathbf{u}^\intercal\mathbf{x}_1 + e')$ from $(\mathbf{u}, r)$), their outputs are closely related. Now, considering each guess g_i for position i:

- If the guess is correct (i.e., the i-th entry of $\mathbf{x}_1$ is g_i), the distribution $\left(\mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal + c \cdot \hat{\mathbf{i}}, \mathbf{s}^\intercal\mathbf{y} + c \cdot g_i + e'\right)$ is computationally indistinguishable from the distribution $(\mathbf{u}, \mathbf{u}^\intercal\mathbf{x}_1 + e')$, and thus also from $(\mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal, \mathbf{s}^\intercal\mathbf{y} + e')$. Given $\rho'_{\textsc{Aux}}$ is a γ-good state, $\mathsf{TI}_{\frac{1}{2}+\gamma'}(i, g_i)$ outputs 1 with $1 - \mathsf{negl}$ probability if all other threshold implementations are ignored (i.e., applied $\mathsf{TI}_{\frac{1}{2}+\gamma'}(i, g_i)$ just after $\mathsf{TI}_{\frac{1}{2}+\gamma}$).
- If the guess is incorrect, the distribution $\left(\mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal + c \cdot \hat{\mathbf{i}}, \mathbf{s}^\intercal\mathbf{y} + c \cdot g_i + e'\right)$ is computationally indistinguishable from $(\mathbf{u}, r)$. Consequently, any state provides no advantage as advice, and $\mathsf{TI}_{\frac{1}{2}+\gamma'}$ outputs 1 with negl probability.

Finally, we apply the quantum union bound to all measurements to demonstrate that the probability of no error occurring during our testing procedure is $1 - \mathsf{negl}$.

In the above proof, we omitted a major issue. Recall that in Step 1, we implement threshold implementation to project the state ρ onto a good state ρ_{good}. Moreover, this threshold implementation is designed to check if the adversary can distinguish between the distributions $(\mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal, \mathbf{s}^\intercal\mathbf{y} + e')$ and $(\mathbf{u}, r)$. As discussed above, at some point, in the intermediate hybrids we need to change these distributions. Once we switch the distributions, the threshold implementation might only work with negligible probability. Our hope, in some cases invoking learning with errors, is to argue that this does not happen. However, it is not clear how to carry out this reduction. After all, the threshold implementation as defined by [Zha20] operates on a superposition of exponentially many samples from a distribution and so, given just one sample from a distribution, it is not possible to perform threshold implementation. We present a useful lemma (in Section 5) where we argue that operationally, the guarantees of threshold implementation (including the output and

the residual state) are not affected when one distribution is replaced with another computationally indistinguishable distribution.

Technical Sections: To maintain the clarity of presentation in the technical sections, we present the proof of security in a different order than the one discussed in the overview, although the main ideas of the proof have been conveyed above.

- Steps 1, 2 and 3 are not presented in order. Section 5 and Section 6 are concerned with implementing Step 2. Section 7 is concerned with implementing Steps 1 and 3.
- Regarding Step 2, we present the useful lemma related to threshold implementation in Section 5. We then discuss the mechanism to extract $\mathbf{x_1}$ from Aux in Section 6.

2 Preliminaries

We denote the security parameter by λ. In this work, when we add/multiply two elements from $\mathbb{Z}_q$, we mean the addition/multiplication in $\mathbb{Z}_q$ by default (i.e. for $a, b \in \mathbb{Z}_q$, $a+b$ means $a+b \pmod q$ and ab means $ab \pmod q$ by default).

2.1 Quantum Computing

We recall some important lemmas from quantum information theory.

Lemma 2.1 ("Almost As Good As New" Lemma, [Aar16])**.** *Let $\rho \in \mathcal{D}(\mathcal{H})$ be a density matrix over a Hilbert space $\mathcal{H}$. Let U be an arbitrary unitary and let $(\mathbf{\Pi}_0, \mathbf{\Pi}_1 = \mathbf{I} - \mathbf{\Pi}_0)$ be projectors acting on $\mathcal{H} \otimes \mathcal{H}_{\mathsf{aux}}$. We interpret $(U, \mathbf{\Pi}_0, \mathbf{\Pi}_1)$ as a measurement performed by appending an ancillary system in the state $|0\rangle\langle 0|_{\mathsf{aux}}$, applying the unitary U and subsequently performing the two-outcome measurement $\{\mathbf{\Pi}_0, \mathbf{\Pi}_1\}$ on the larger system. Suppose that the outcome corresponding to $\mathbf{\Pi}_0$ occurs with probability $1-\varepsilon$, for some $\varepsilon \in [0,1]$. In other words, it holds that $\mathrm{Tr}[\mathbf{\Pi}_0(U\rho \otimes |0\rangle\langle 0|_{\mathsf{aux}} U^\dagger)] = 1-\varepsilon$. Then,*

$$\mathrm{TD}(\rho, \widetilde{\rho}) \leq \sqrt{\varepsilon},$$

where $\widetilde{\rho}$ is the state after performing the measurement and applying $U^\dagger$, and after tracing out $\mathcal{H}_{\mathsf{aux}}$:

$$\widetilde{\rho} = \mathrm{Tr}_{\mathsf{aux}}\left[U^\dagger\left(\mathbf{\Pi}_0 U(\rho \otimes |0\rangle\langle 0|_{\mathsf{aux}})U^\dagger \mathbf{\Pi}_0 + \mathbf{\Pi}_1 U(\rho \otimes |0\rangle\langle 0|_{\mathsf{aux}})U^\dagger \mathbf{\Pi}_1\right) U\right].$$

Lemma 2.2 (Quantum Union Bound, [Gao15])**.** *Let $\rho \in \mathcal{D}(\mathcal{H})$ be a state and let $\mathbf{\Pi}_1, \ldots, \mathbf{\Pi}_n \geq 0$ be sequence of (orthogonal) projections acting on $\mathcal{H}$. Suppose that, for every $i \in [n]$, it holds that $\mathrm{Tr}[\mathbf{\Pi}_i \rho] = 1-\varepsilon_i$, for $\varepsilon_i \in [0,1]$. Then, if we sequentially measure ρ with projective measurements $\{\mathbf{\Pi}_1, \mathbf{I}-\mathbf{\Pi}_1\}, \ldots, \{\mathbf{\Pi}_n, \mathbf{I}-\mathbf{\Pi}_n\}$, the probability that all measurements succeed is at least*

$$\mathrm{Tr}[\mathbf{\Pi}_n \cdots \mathbf{\Pi}_1 \rho \mathbf{\Pi}_1 \cdots \mathbf{\Pi}_n] \geq 1 - 4\sum_{i=1}^{n} \varepsilon_i.$$

2.2 Lattices and Cryptography

We adapt notations from [APV23] and keep it same as much as we can. The following subsection is copied verbatim from [APV23]. Let $n, m, p, q \in \mathbb{N}$ be positive integers. The rounding operation for $q \geq p \geq 2$ is the function

$$\lfloor \cdot \rceil_p \ : \ \mathbb{Z}_q \to \mathbb{Z}_p \ : \ x \mapsto \lfloor (p/q) \cdot x \rceil \pmod{p}.$$

A *lattice* $\Lambda \subset \mathbb{R}^m$ is a discrete subgroup of $\mathbb{R}^m$. Given a lattice $\Lambda \subset \mathbb{R}^m$ and a vector $\mathbf{t} \in \mathbb{R}^m$, we define the coset with respect to vector $\mathbf{t}$ as the lattice shift $\Lambda - \mathbf{t} = \{\mathbf{x} \in \mathbb{R}^m : \mathbf{x} + \mathbf{t} \in \Lambda\}$. Note that many different shifts $\mathbf{t}$ can define the same coset.

In this work, we mainly consider *q-ary lattices* Λ that that satisfy $q\mathbb{Z}^m \subseteq \Lambda \subseteq \mathbb{Z}^m$, for some integer modulus $q \geq 2$. Specifically, we consider the lattice generated by a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for some $n, m \in \mathbb{N}$ that consists of all vectors which are perpendicular to the rows of $\mathbf{A}$, namely

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{x} = \mathbf{0} \pmod{q}\}.$$

For any *syndrome* $\mathbf{y} \in \mathbb{Z}_q^n$ in the column span of $\mathbf{A}$, we also consider the coset $\Lambda_q^{\mathbf{y}}(\mathbf{A})$ given by

$$\Lambda_q^{\mathbf{y}}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \pmod{q}\} = \Lambda_q^{\perp}(\mathbf{A}) + \mathbf{c},$$

where $\mathbf{c} \in \mathbb{Z}^m$ is an arbitrary integer solution to the equation $\mathbf{Ac} = \mathbf{y} \pmod{q}$.

Gaussian Distribution. The *Gaussian measure* ρ_σ with parameter $\sigma > 0$ is defined as

$$\rho_\sigma(\mathbf{x}) = \exp(-\pi \|\mathbf{x}\|^2 / \sigma^2), \qquad \forall \mathbf{x} \in \mathbb{R}^m.$$

Let $\Lambda \subset \mathbb{R}^m$ be a lattice and let $\mathbf{t} \in \mathbb{R}^m$. We define the *Gaussian mass* of $\Lambda - \mathbf{t}$ as the quantity

$$\rho_\sigma(\Lambda - \mathbf{t}) = \sum_{\mathbf{y} \in \Lambda} \rho_\sigma(\mathbf{y} - \mathbf{t}).$$

The *discrete Gaussian distribution* $D_{\Lambda - \mathbf{t}, \sigma}$ assigns probability proportional to $e^{-\pi \|\mathbf{x}\|^2 / \sigma^2}$ to every vector $\mathbf{x} \in \Lambda - \mathbf{t}$. In other words, we have

$$D_{\Lambda - \mathbf{t}, \sigma}(\mathbf{x}) = \frac{\rho_\sigma(\mathbf{x})}{\rho_\sigma(\Lambda - \mathbf{t})}, \qquad \forall \mathbf{x} \in \Lambda - \mathbf{t}.$$

In particular, for any coset $\Lambda_q^{\mathbf{y}}(\mathbf{A})$ with $\mathbf{y} \in \mathbb{Z}_q^n$, the discrete Gaussian $D_{\Lambda_q^{\mathbf{y}}(\mathbf{A}), \sigma}$ (centered around the origin) assigns probability proportional to $e^{-\pi \|\mathbf{x}\|^2 / \sigma^2}$ to every vector $\mathbf{x} \in \Lambda_q^{\mathbf{y}}(\mathbf{A})$, and 0 otherwise.

Let $\mathcal{B}^m(\mathbf{0}, r) = \{\mathbf{x} \in \mathbb{R}^m : \|\mathbf{x}\| \leq r\}$ denote the m-dimensional ball of radius $r > 0$. We use of the following tail bound for the Gaussian mass of a lattice [Ban93, Lemma 1.5 (ii)].

Lemma 2.3. *For any m-dimensional lattice Λ, shift $\mathbf{t} \in \mathbb{R}^m$, $\sigma > 0$ and $c \geq (2\pi)^{-\frac{1}{2}}$ it holds that*

$$\rho_\sigma\left((\Lambda - \mathbf{t}) \setminus \mathcal{B}^m(\mathbf{0}, c\sqrt{m}\sigma)\right) \leq (2\pi e c^2)^{\frac{m}{2}} e^{-\pi c^2 m} \rho_\sigma(\Lambda).$$

A consequence of Lemma 2.3 is that the Gaussian distribution $D_{\mathbb{Z}^m, \sigma}$ is essentially only supported on the finite set $\{\mathbf{x} \in \mathbb{Z}^m : \|\mathbf{x}\| \leq \sigma\sqrt{m}\}$, which suggests the use of *truncation*.

Definition 2.4 (Truncated discrete Gaussian distribution)**.** *Let $m \in \mathbb{N}$, $q \geq 2$ be an integer modulus and let $\sigma > 0$ be a parameter. Then, the* truncated *discrete Gaussian distribution $D_{\mathbb{Z}_q^m,\sigma}$ with finite support $\{\mathbf{x} \in \mathbb{Z}^m \cap (-\frac{q}{2}, \frac{q}{2}]^m : \|\mathbf{x}\| \leq \sigma\sqrt{m}\}$ is defined as the density*

$$D_{\mathbb{Z}_q^m,\sigma}(\mathbf{x}) = \frac{\rho_\sigma(\mathbf{x})}{\sum\limits_{\mathbf{y} \in \mathbb{Z}_q^m, \|\mathbf{y}\| \leq \sigma\sqrt{m}} \rho_\sigma(\mathbf{y})}.$$

Finally, we recall the following *noise smudging* property.

Lemma 2.5 (Noise smudging, [DGT+10])**.** *Let $y, \sigma > 0$. Then, the statistical distance between the distribution $D_{\mathbb{Z},\sigma}$ and $D_{\mathbb{Z},\sigma+y}$ is at most y/σ.*

We use the following technical lemma on the min-entropy of the truncated discrete Gaussian distribution, which we prove below.

Lemma 2.6 (min-entropy of the truncated discrete Gaussian, [APV23], Lemma 2.10)**.** *Let $n \in \mathbb{N}$ and let q be a prime with $m \geq 2n \log q$. Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be a matrix whose columns generate $\mathbb{Z}_q^n$. Then, for any $\sigma \geq \omega(\sqrt{\log m})$, there exists a negligible $\varepsilon(m)$ such that*

$$\max_{\mathbf{y} \in \mathbb{Z}_q^n} \max_{\substack{\mathbf{x} \in \mathbb{Z}_q^m, \|\mathbf{x}\| \leq \sigma\sqrt{m} \\ \mathbf{A}\mathbf{x} = \mathbf{y} \pmod q}} \left\{ \frac{\rho_\sigma(\mathbf{x})}{\sum\limits_{\substack{\mathbf{z} \in \mathbb{Z}_q^m, \|\mathbf{z}\| \leq \sigma\sqrt{m} \\ \mathbf{A}\mathbf{z} = \mathbf{y} \pmod q}} \rho_\sigma(\mathbf{z})} \right\} \leq 2^{-m+1} \cdot \frac{1+\varepsilon}{1-\varepsilon}.$$

The Short Integer Solution problem. The *Short Integer Solution* (SIS) problem was introduced by Ajtai [Ajt96] in his seminal work on average-case lattice problems.

Definition 2.7 (Short Integer Solution problem, [Ajt96])**.** *Let $n, m \in \mathbb{N}$, $q \geq 2$ be a modulus and let $\beta > 0$ be a parameter. The Short Integer Solution problem ($\mathsf{SIS}^m_{n,q,\beta}$) problem is to find a short solution $\mathbf{x} \in \mathbb{Z}^m$ with $\|\mathbf{x}\| \leq \beta$ such that $\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \pmod q$ given as input a matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$.*

Micciancio and Regev [MR07] showed that the SIS problem is, on the average, as hard as approximating worst-case lattice problems to within small factors. Subsequently, Gentry, Peikert and Vaikuntanathan [GPV07] gave an improved reduction showing that, for parameters $m = \text{poly}(n)$, $\beta = \text{poly}(n)$ and prime $q \geq \beta \cdot \omega(\sqrt{n \log q})$, the average-case $\mathsf{SIS}^m_{n,q,\beta}$ problem is as hard as approximating the shortest independent vector problem (SIVP) problem in the worst case to within a factor $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$. We assume that $\mathsf{SIS}^m_{n,q,\beta}$, for $m = \Omega(n \log q)$, $\beta = 2^{o(n)}$ and $q = 2^{o(n)}$, is hard against quantum adversaries running in time $\text{poly}(q)$ with success probability $\text{poly}(1/q)$.

The Learning with Errors problem. The *Learning with Errors* problem was introduced by Regev [Reg05] and serves as the primary basis of hardness of post-quantum cryptosystems. The problem is defined as follows.

Definition 2.8 (Learning with Errors problem, [Reg05])**.** *Let $n, m \in \mathbb{N}$ be integers, let $q \geq 2$ be a modulus and let $\alpha \in (0,1)$ be a noise ratio parameter. The (decisional) Learning with Errors ($\mathsf{LWE}^m_{n,q,\alpha q}$) problem is to distinguish between the following samples*

$$(\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{s}^\intercal \mathbf{A} + \mathbf{e}^\intercal \pmod q) \quad \text{and} \quad (\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m),$$

where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ is a uniformly random vector and where $\mathbf{e} \sim D_{\mathbb{Z}^m,\alpha q}$ is a discrete Gaussian error vector. We rely on the quantum $\mathsf{LWE}^m_{n,q,\alpha q}$ assumption which states that the samples above are computationally indistinguishable for any QPT algorithm.

As shown in [Reg05], the $\mathsf{LWE}^m_{n,q,\alpha q}$ problem with parameter $\alpha q \geq 2\sqrt{n}$ is at least as hard as approximating the shortest independent vector problem (SIVP) to within a factor of $\gamma = \widetilde{O}(n/\alpha)$ in worst case lattices of dimension n. In this work we assume the subexponential hardness of $\mathsf{LWE}^m_{n,q,\alpha q}$ which relies on the worst case hardness of approximating short vector problems in lattices to within a subexponential factor. We assume that the $\mathsf{LWE}^m_{n,q,\alpha q}$ problem, for $m = \Omega(n \log q)$, $q = 2^{o(n)}$, $\alpha = 1/2^{o(n)}$, is hard against quantum adversaries running in time $\mathrm{poly}(q)$. We note that this parameter regime implies $\mathsf{SIS}^m_{n,q,\beta}$ [SSTX09].

Trapdoors for lattices. We use the following *trapdoor* property for the LWE problem.

Theorem 2.9 ([MP11], Theorem 5.1)**.** *Let $n, m \in \mathbb{N}$ and $q \in \mathbb{N}$ be a prime with $m = \Omega(n \log q)$. There exists a randomized algorithms with the following properties:*

- $\mathsf{GenTrap}(1^n, 1^m, q)$*: on input $1^n, 1^m$ and q, returns a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathsf{td}_\mathbf{A}$ such that the distribution of $\mathbf{A}$ is negligibly (in the parameter n) close to uniform.*
- $\mathsf{Invert}(\mathbf{A}, \mathsf{td}_\mathbf{A}, \mathbf{b})$*: on input $\mathbf{A}$, $\mathsf{td}_\mathbf{A}$ and $\mathbf{b} = \mathbf{s}^\intercal \cdot \mathbf{A} + \mathbf{e}^\intercal \pmod q$, where $\|\mathbf{e}\| \leq q/(C_T\sqrt{n \log q})$ and $C_T > 0$ is a universal constant, returns $\mathbf{s}$ and $\mathbf{e}$ with overwhelming probability over $(\mathbf{A}, \mathsf{td}_\mathbf{A}) \leftarrow \mathsf{GenTrap}(1^n, 1^m, q)$.*

We use $\approx_c$ to denote two distributions or two quantum states are computationally indistinguishable. We use $\approx_s$ to denote two distributions or two quantum states are statistically indistinguishable.

2.3 Gaussian Coset States

In Algorithm 1 ([APV23], Algorithm 1), they give a procedure called $\mathsf{GenGauss}$ that, on input $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\sigma > 0$, generates a Gaussian superposition state of the form

$$|\psi_\mathbf{y}\rangle = \sum_{\substack{\mathbf{x}\in\mathbb{Z}_q^m: \\ \mathbf{A}\mathbf{x}=\mathbf{y} \pmod q}} \rho_\sigma(\mathbf{x}) |\mathbf{x}\rangle .$$

It is computationally hard to distinguish whether a Gaussian Coset State is measured in the computational basis or not.

Algorithm 1: $\mathsf{GenGauss}(\mathbf{A}, \sigma)$

Input : Matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and parameter $\sigma = \Omega(\sqrt{m})$.
Output: Gaussian state $|\psi_{\mathbf{y}}\rangle$ and $\mathbf{y} \in \mathbb{Z}_q^n$.

1 Prepare a Gaussian superposition in system X with parameter $\sigma > 0$:

$$|\psi\rangle_{XY} = \sum_{\mathbf{x}\in\mathbb{Z}_q^m} \rho_\sigma(\mathbf{x}) \, |\mathbf{x}\rangle_X \otimes |\mathbf{0}\rangle_Y .$$

2 Apply the unitary $U_{\mathbf{A}} : |\mathbf{x}\rangle |\mathbf{0}\rangle \to |\mathbf{x}\rangle |\mathbf{A}\cdot\mathbf{x} \bmod q\rangle$ on system X and Y:

$$|\psi\rangle_{XY} = \sum_{\mathbf{x}\in\mathbb{Z}_q^m} \rho_\sigma(\mathbf{x}) \, |\mathbf{x}\rangle_X \otimes |\mathbf{A}\cdot\mathbf{x} \bmod q\rangle_Y .$$

3 Measure system Y in the computational basis, resulting in the state

$$|\psi_{\mathbf{y}}\rangle_{XY} = \sum_{\substack{\mathbf{x}\in\mathbb{Z}_q^m \\ \mathbf{A}\mathbf{x}=\mathbf{y} \bmod q}} \rho_\sigma(\mathbf{x}) \, |\mathbf{x}\rangle_X \otimes |\mathbf{y}\rangle_Y .$$

4 Output the state $|\psi_{\mathbf{y}}\rangle$ in system X and the outcome $\mathbf{y} \in \mathbb{Z}_q^n$ in system Y.

Theorem 2.10 (Gaussian-collapsing property, [Por22], Theorem 4)**.** *Let $n \in \mathbb{N}$ and q be a prime with $m \geq 2n\log q$, each parameterized by $\lambda \in \mathbb{N}$. Let $\sqrt{8m} < \sigma < q/\sqrt{8m}$. Then, the following samples are computationally indistinguishable assuming the quantum hardness of decisional $\mathsf{LWE}^m_{n,q,\alpha q}$, for any noise ratio $\alpha \in (0,1)$ with relative noise magnitude $1/\alpha = \sigma \cdot 2^{o(n)}$:*

$$\left(\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m},\ |\psi_{\mathbf{y}}\rangle = \sum_{\substack{\mathbf{x}\in\mathbb{Z}_q^m \\ \mathbf{A}\mathbf{x}=\mathbf{y}}} \rho_\sigma(\mathbf{x}) \ |\mathbf{x}\rangle\,,\ \mathbf{y}\in\mathbb{Z}_q^n\right) \approx_c \left(\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m},\ |\mathbf{x}_0\rangle\,,\ \mathbf{A}\cdot\mathbf{x}_0 \in \mathbb{Z}_q^n\right)$$

where $(|\psi_{\mathbf{y}}\rangle, \mathbf{y}) \leftarrow \mathsf{GenGauss}(\mathbf{A}, \sigma)$ and where $\mathbf{x}_0 \sim D_{\mathbb{Z}_q^m, \frac{\sigma}{\sqrt{2}}}$ is a discrete Gaussian error.

The algorithm $\mathsf{GenGauss}$ can generate a Gaussian coset state $|\psi_{\mathbf{y}}\rangle$ for random $\mathbf{y}$. To generate a Gaussian coset state $|\psi_{\mathbf{y}}\rangle$ for a specific $\mathbf{y}$, we need a trapdoor $\mathsf{td}_{\mathbf{A}}$ for matrix $\mathbf{A}$.

Theorem 2.11 (Quantum Discrete Gaussian Sampler, [APV23], Theorem 3.3)**.** *Let $n \in \mathbb{N}$, q be a prime with $m \geq 2n\log q$ and $\sqrt{8m} < \sigma < q/\sqrt{8m}$. Let $(\mathbf{A}, \mathsf{td}_{\mathbf{A}}) \leftarrow \mathsf{GenTrap}(1^n, 1^m, q)$ be sampled as in Theorem 2.9 and let $\mathbf{y} \in \mathbb{Z}_q^n$ be arbitrary. Then, there exists a QPT algorithm $\mathsf{QSampGauss}(\mathbf{A}, \mathsf{td}_{\mathbf{A}}, \mathbf{y}, \sigma)$ that given $\mathbf{A}, \mathbf{y}$ and a trapdoor of $\mathbf{A}$ outputs a state which is within negligible trace distance of the (normalized variant of the) state*

$$|\psi_{\mathbf{y}}\rangle = \sum_{\substack{\mathbf{x}\in\mathbb{Z}_q^m \\ \mathbf{A}\mathbf{x}=\mathbf{y} \pmod q}} \rho_\sigma(\mathbf{x}) \, |\mathbf{x}\rangle$$

with overwhelming probability.

2.4 Threshold Implementation and its Approximate Version

In the subsection, we review some techniques called *Threshold Implementation* [ALL+21], which is a simple extension of Projective Implementation [Zha20]. It allows us to test whether the success probability of a *quantum algorithm* exceeds some threshold.

Theorem 2.12 (Threshold implementation, [ALL+21])**.** *Let $\gamma \in (0,1)$ be a parameter and let $\mathcal{P} = (P, Q)$ be a two-outcome POVM, where P has an eigenbasis $\{|\psi_i\rangle\}$ with associated eigenvalues $\{\lambda_i\}$. Then, there exists a projective threshold implementation $(\mathsf{TI}_\gamma(\mathcal{P}), \mathrm{I} - \mathsf{TI}_\gamma(\mathcal{P}))$ such that*

- *$\mathsf{TI}_\gamma(\mathcal{P})$ projects a quantum state into the subspace spanned by $\{|\psi_i\rangle\}$ whose eigenvalues λ_i satisfy the property $\lambda_i \leq \gamma$.*
- *$\mathrm{I} - \mathsf{TI}_\gamma(\mathcal{P})$ projects a quantum state into the subspace spanned by $\{|\psi_i\rangle\}$ whose eigenvalues λ_i satisfy the property $\lambda_i > \gamma$.*

Unfortunately, the threshold implementation can, in general, not be efficiently computable. However, inspired by the work of Marriott and Watrous [MW05], Zhandry [Zha20] showed that the approximate version of the threshold implementation can be implemented efficiently as long as the POVM is a mixture of projective measurements. We first review the definition of mixture of projective measurements.

Definition 2.13 (Mixture of projective measurements)**.** *Let $\mathcal{P} = \{\mathcal{P}_i\}_{i \in \mathcal{I}}$ be a collection of binary outcome projective measurements $\mathcal{P}_i = (P_i, Q_i)$ over the same Hilbert space $\mathcal{H}$, and suppose that P_i corresponds to outcome 1 and Q_i corresponds to outcome 0. Let D be a distribution over the the index set $\mathcal{I}$. Then, $\mathcal{P}_D = (P_D, Q_D)$ is the following mixture of projective measurements:*

$$P_D = \sum_{i \in \mathcal{I}} \Pr[i \leftarrow D]\, P_i \qquad \text{and} \qquad Q_D = \sum_{i \in \mathcal{I}} \Pr[i \leftarrow D]\, Q_i.$$

In other words, $\mathcal{P}_D$ is the same as first sampling i according to the distribution D, and then applying the projective measurements $\mathcal{P}_i$.

For any mixture of projective measurements $\mathcal{P}_D$, the approximate threshold implementation satisfies the following properties.

Lemma 2.14 (Approximate threshold implementation, Theorem 6.2 in [Zha20] and Corollary 1 in [ALL+21])**.** *Let $\mathcal{P}_D = (P_D, Q_D)$ be a binary outcome POVM over Hilbert space $\mathcal{H}$ that is a mixture of projective measurements over some distribution D. Let $\varepsilon, \delta, \gamma \in (0,1)$. Then, there exists an efficient binary-outcome quantum algorithm $\mathsf{ATI}^{\varepsilon,\delta}_{\mathcal{P},D,\gamma}$, interpreted as the POVM element corresponding to outcome 1, such that the following holds:*

- *For all quantum states ρ, $\mathrm{Tr}[\mathsf{ATI}^{\varepsilon,\delta}_{\mathcal{P},D,\gamma-\varepsilon}\, \rho] \geq \mathrm{Tr}[\mathsf{TI}_\gamma(\mathcal{P}_D)\, \rho] - \delta$.*
- *For all quantum states ρ, it holds that $\mathrm{Tr}[\mathsf{TI}_{\gamma-2\varepsilon}(\mathcal{P}_D)\, \rho'] \geq 1 - 2\delta$, where ρ' is the post-measurement state which results from applying the measurement $\mathsf{ATI}^{\varepsilon,\delta}_{\mathcal{P},D,\gamma}$ to ρ and obtaining outcome 1.*
- *The expected running time to implement $\mathsf{ATI}^{\varepsilon,\delta}_{\mathcal{P},D,\gamma}$ is proportional to $\mathrm{poly}(1/\varepsilon, \log(1/\delta))$, the time it takes to implement P_D, and the time it takes to sample from D.*

3 Definition: Key-Revocable Public-Key Encryption

A key-revocable public-key encryption is a type of public-key encryption. Consider the case where the secret key holder wishes to temporarily give the secret key to an third party and later wants to take it back while maintaining the security i.e. the third party upon taken its key away, can't decrypt any message later. This is impossible in the classical case since the third party can always copy the secret key locally. But we may achieve this functionality by representing the secret key as a quantum state.

Definition 3.1 (Key-Revocable Public-Key Encryption [APV23])**.** *A key-revocable public-key encryption scheme consists of efficient algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Revoke})$*, where* Enc *is a* PPT *algorithm and* $\mathsf{KeyGen}, \mathsf{Dec}, \mathsf{Revoke}$ *are* QPT *algorithms defined as follows:*

- $\mathsf{KeyGen}(1^\lambda)$*: given as input a security parameter* λ*, output a public key* PK*, a master secret key* MSK *and a quantum decryption key* ρ_{SK}*.*
- $\mathsf{Enc}(\mathsf{PK}, \mu)$*: given a public key* PK *and plaintext* $\mu \in \{0,1\}$*, output a ciphertext* CT*.*
- $\mathsf{Dec}(\rho_{\mathsf{SK}}, \mathsf{CT})$*: given a decryption key* ρ_{SK} *and ciphertext* CT*, output a message* y*.*
- $\mathsf{Revoke}(\mathsf{PK}, \mathsf{MSK}, \sigma)$*: given as input a master secret key* MSK*, a public key* PK *and quantum state* σ*, output* Valid *or* $\mathsf{Invalid}$*.*

Correctness of Decryption. For $\mu \in \{0,1\}$, the following holds:

$$\Pr\left[\mu \leftarrow \mathsf{Dec}(\rho_{\mathsf{SK}}, \mathsf{CT}) : \begin{smallmatrix}(\mathsf{PK},\mathsf{MSK},\rho_{\mathsf{SK}})\leftarrow\mathsf{KeyGen}(1^\lambda)\\ \mathsf{CT}\leftarrow\mathsf{Enc}(\mathsf{PK},\mu)\end{smallmatrix}\right] \geq 1 - \mathsf{negl}.$$

Correctness of Revocation. The following holds:

$$\Pr\left[\mathsf{Valid} \leftarrow \mathsf{Revoke}(\mathsf{PK}, \mathsf{MSK}, \rho_{\mathsf{SK}}) : (\mathsf{PK}, \mathsf{MSK}, \rho_{\mathsf{SK}}) \leftarrow \mathsf{KeyGen}(1^\lambda)\right] \geq 1 - \mathsf{negl}.$$

3.1 Security Definition

The security captures the case where the adversary is given the key and later taken back. After that, if the key passes the revocation check the adversary is asked to play a CPA like game that it is given either the ciphertext of a chosen message or a random message. The adversary wins if it can distinguish between these two cases.

Definition 3.2. *A key-revocable public-key encryption scheme* $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Revoke})$ *is* (ϵ, δ)*-secure if, for every* QPT *adversary* $\mathcal{A}$ *with*

$$\Pr\left[\mathsf{Invalid} \leftarrow \mathsf{Expt}^{\Sigma,\mathcal{A}}(1^\lambda, b)\right] \leq \delta(\lambda)$$

for $b \in \{0,1\}$*, it holds that*

$$\left|\Pr\left[1 \leftarrow \mathsf{Expt}^{\Sigma,\mathcal{A}}(1^\lambda, 0)\right] - \Pr\left[1 \leftarrow \mathsf{Expt}^{\Sigma,\mathcal{A}}(1^\lambda, 1)\right]\right| \leq \epsilon(\lambda),$$

where $\mathsf{Expt}^{\Sigma,\mathcal{A}}(1^\lambda, b)$ *is defined as Figure 1. If* $\delta(\lambda) = 1 - \frac{1}{\mathrm{poly}(\lambda)}$ *and* $\epsilon(\lambda) = \mathsf{negl}(\lambda)$*, we simply say the key-revocable encryption scheme is secure.*

$\underline{\mathsf{Expt}^{\Sigma,\mathcal{A}}(1^\lambda, b)}$:

Initialization Phase:

- The challenger runs $(\mathsf{PK}, \mathsf{MSK}, \rho_{\mathsf{SK}}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and sends $(\mathsf{PK}, \rho_{\mathsf{SK}})$ to $\mathcal{A}$.

Revocation Phase:

- The challenger sends the message `REVOKE` to $\mathcal{A}$.
- The adversary $\mathcal{A}$ returns a state σ.
- The challenger aborts if $\mathsf{Revoke}(\mathsf{MSK}, \mathsf{PK}, \sigma)$ outputs $\mathsf{Invalid}$.

Guessing Phase:

- $\mathcal{A}$ submits a plaintext $\mu \in \{0,1\}$ to the challenger.
- If $b = 0$: The challenger sends $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{PK}, \mu)$ to $\mathcal{A}$. Else, if $b = 1$, the challenger sends $\mathsf{CT} \leftarrow \mathcal{C}$, where $\mathcal{C}$ is the ciphertext space of ℓ bit messages.
- Output $b_{\mathcal{A}}$ if the output of $\mathcal{A}$ is $b_{\mathcal{A}}$.

Figure 1: Security Experiment

4 Construction: Key Revocable Dual-Regev Encryption

The construction is exactly the same as the construction in [APV23]. We include the construction here for completeness.

Construction 4.1 (Key Revocable Dual-Regev Encryption [APV23])**.** *Let $n \in \mathbb{N}$ be the security parameter and $m \in \mathbb{N}$. Let $q \geq 2$ be a prime and let $\alpha, \beta, \sigma > 0$ be parameters. The key-revocable public key scheme* $\mathsf{RevDual} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Revoke})$ *consists of the following* QPT *algorithms:*

- $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{PK}, \rho_{\mathsf{SK}}, \mathsf{MSK})$*: Sample* $(\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \mathsf{td}_{\mathbf{A}}) \leftarrow \mathsf{GenTrap}(1^n, 1^m, q)$ *where* $\mathsf{GenTrap}$ *is the algorithm that generates the LWE matrix with its trapdoor. Then generate a Gaussian superposition* $(|\psi_{\mathbf{y}}\rangle, \mathbf{y}) \leftarrow \mathsf{GenGauss}(\mathbf{A}, \sigma)$ *for some* $\mathbf{y} \in \mathbb{Z}_q^n$*. Output* $\mathsf{PK} = (\mathbf{A}, \mathbf{y}), \rho_{\mathsf{SK}} = |\psi_{\mathbf{y}}\rangle$ *and* $\mathsf{MSK} = \mathsf{td}_{\mathbf{A}}$*.*
- $\mathsf{Enc}(\mathsf{PK}, \mu) \to \mathsf{CT}$*: to encrypt a bit* $\mu \in \{0,1\}$*, sample a random vector* $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ *and errors* $\mathbf{e} \sim D_{\mathbb{Z}^m, \alpha q}$ *and* $e' \sim D_{\mathbb{Z}, \beta q}$ *and output the ciphertext pair*
$$\mathsf{CT} = \left(\mathbf{s}^\intercal \mathbf{A} + \mathbf{e}^\intercal \pmod q, \mathbf{s}^\intercal \mathbf{y} + e' + \mu \cdot \lfloor \frac{q}{2} \rfloor \pmod q\right) \in \mathbb{Z}_q^m \times \mathbb{Z}_q.$$
- $\mathsf{Dec}(\rho_{\mathsf{SK}}, \mathsf{CT}) \to \{0,1\}$ *: to decrypt* CT*, apply the unitary* $U : |\mathbf{x}\rangle |0\rangle \to |\mathbf{x}\rangle |\mathsf{CT} \cdot (-\mathbf{x}, 1)^\intercal\rangle$ *on input* $|\psi_{\mathbf{y}}\rangle |0\rangle$*, where* $\rho_{\mathsf{SK}} = |\psi_{\mathbf{y}}\rangle$*, and measure the second register in the computational basis. Output* 0*, if the measurement outcome is closer to* 0 *than to* $\lfloor \frac{q}{2} \rfloor$*, and output* 1*, otherwise.*

- $\mathsf{Revoke}(\mathsf{MSK}, \mathsf{PK}, \rho) \to \{\top, \bot\}$: *on input* $\mathsf{td}_{\mathbf{A}} \leftarrow \mathsf{MSK}$ *and* $(\mathbf{A}, \mathbf{y}) \leftarrow \mathsf{PK}$*, apply the measurement* $\{|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}|, I - |\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}|\}$ *onto the state* ρ *using the procedure* $\mathsf{QSampGauss}(\mathbf{A}, \mathsf{td}_{\mathbf{A}}, \mathbf{y}, \sigma)$*. Output* $\top$ *if the measurement is successful, and* $\bot$ *otherwise.*

From [APV23], this construction satisfies the correctness of decryption and the correctness of revocation. In this work, we will focus on showing the construction is in fact secure.

Theorem 4.2. *Let* $n \in \mathbb{N}$ *and* q *be a prime modulus with* $q = 2^{o(n)}$ *and* $m \geq 2n \log q$*, each parameterized by security parameter* $\lambda \in \mathbb{N}$*. Let* $\sqrt{8m} < \sigma < q/\sqrt{8m}$ *and let* $\alpha, \beta \in (0, 1)$ *be noise ratios chosen such that* $\beta/\alpha = 2^{o(n)}$ *and* $1/\alpha = 2^{o(n)} \cdot \sigma$*. Then, assuming the polynomial hardness of* $\mathsf{LWE}^m_{n,q,\alpha q}$ *with sub-exponential modulus, the scheme* $\mathsf{RevDual} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Revoke})$ *in Construction 4.1 is a secure key-revocable public-key encryption scheme according to Definition 3.2.*

We organize the proof of Theorem 4.2 in the following way:

- In Section 5, we prove an important property for approximate threshold implementation, which allows us to do hybrid arguments between approximate threshold implementation on computationally indistinguishable distributions.
- In Section 6, we present our construction for the almost perfect preimage extractor that lies in the heart of our result.
- In Section 7, we complete our proof of the above theorem.

5 Indistinguishability on Approximate Threshold Implementation

Zhandry [Zha20] analyzed the relationship between the output distribution of $\mathsf{TI}_{\gamma_0}(\mathcal{P}_{D_0})$ and $\mathsf{TI}_{\gamma_1}(\mathcal{P}_{D_1})$ (and $\mathsf{ATI}_{\mathcal{P},D_0,\gamma_0}$ and $\mathsf{ATI}_{\mathcal{P},D_1,\gamma_1}$) for some thresholds γ_0 and γ_1 on the same state for two computationally indistinguishable distributions D_0 and D_1. However, in our work, we also care about the residual state after applying the procedures. So we give a more precise analysis below.

In this section, we show how to leverage a (possibly not efficiently constructible) quantum state ρ on which $\mathsf{ATI}_{\mathcal{P},D_0,\gamma}$ and $\mathsf{ATI}_{\mathcal{P},D_1,\gamma}$ behave differently to construct a quantum distinguishing algorithm (with auxiliary state ρ) for D_0 and D_1. This can be viewed as an extension of Theorem 6.5 and Corollary 6.9 in [Zha20].

This result allows us to do hybrid arguments between $\mathsf{ATI}_{\mathcal{P},D_0,\gamma}$ and $\mathsf{ATI}_{\mathcal{P},D_1,\gamma}$ with exactly the same threshold parameter γ for computationally indistinguishable distributions D_0 and D_1 even when there are some efficient quantum procedure on the residual state after applying ATI. Notably, it applies even when we need some classical advice to sample from D_0 and D_1, in which case, our quantum distinguishing algorithm additionally takes the same classical advice and distinguishes D_0 and D_1.

Lemma 5.1. *Let* $\mathcal{P}$ *be a collection of projective measurements indexed by some set* $\mathcal{I}$*. Suppose* $\mathcal{P}$ *can be implemented by a quantum circuit of size* $|\mathcal{P}|$*. Let* D_0, D_1 *be two efficiently sampleable distributions over* $\mathcal{I}$*. For any state* $\rho \in \mathcal{D}(\mathcal{H})$*, denote* $(b, \rho') \leftarrow \mathsf{ATI}^{\varepsilon,\delta}_{\mathcal{P},D,\gamma}(\rho)$ *be the procedure that runs* $\mathsf{ATI}^{\varepsilon,\delta}_{\mathcal{P},D,\gamma}$ *on state* ρ*, and gets an output* b *and the post-measurement state* ρ'*. For any*

polynomial μ, any quantum state ρ and any (possibly quantum) predicate $h : \{0,1\} \times \mathcal{D}(\mathcal{H}) \to \{0,1\}$ with circuit size $|h|$, if

$$\left|\Pr\left[h(b,\rho')=1 \middle| (b,\rho') \leftarrow \mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_0,\gamma}(\rho)\right] - \Pr\left[h(b,\rho')=1 \middle| (b,\rho') \leftarrow \mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_1,\gamma}(\rho)\right]\right| \geq \frac{1}{\mu(\lambda)}.$$

Then there exists a quantum circuit $\mathcal{C}$ of size $\mathrm{poly}(\lambda, 1/\epsilon, \log(1/\delta), \mu, |\mathcal{P}|, |h|)$ (which only use the quantum circuits to implement $\mathcal{P}$, h and to sample D_0, D_1 as a black box)

$$\left|\Pr\left[\mathcal{C}(\rho,x)=b : b \xleftarrow[x\sim D_b]{\$} \{0,1\}\right] - \frac{1}{2}\right| \geq \frac{1}{(\mu(\lambda))^3 \cdot \mathrm{poly}(\lambda, 1/\epsilon, \log(1/\delta))}$$

which is an inverse polynomial if μ is a polynomial.

Proof. The proof follows the same idea as the proof for Theorem 6.5 in [Zha20]. Roughly speaking, the output of $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$ can be approximated up to inverse polynomial additive error given only polynomial samples from D. We refer the reader to Appendix A for the full proof. □

ATI may change the input state in an essential way even when it outputs 1 with overwhelming probability because ATI is not a projector. For example, let a pure quantum state ρ be a superposition of eigenvectors (of $\mathcal{P}_D$) $|\psi_i\rangle$ whose eigenvalues λ_i satisfy the property $\lambda_i \geq \gamma + 10\epsilon$. If we apply $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$ on ρ, we will get outcome 1 with almost certainty, but the residual state ρ' may lose coherence and become closer to a mixture of $|\psi_i\rangle$.

When we know the ATI outputs 0 or 1 with overwhelming probability, it is a good idea to *minimize the disturbance* by purifying ATI and performing uncompututation, just like the famous gentle measurements. To be more precise, we consider the projective version of $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$. Formally, $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$ can be written as introducing $\mathrm{poly}(1/\epsilon, \log(1/\delta))$ ancillas initialized as $|0\rangle$, applying a unitary U on the state, and then applying a projective measurement $(|0\rangle\langle 0|, |1\rangle\langle 1|)$ on the output register of state to get the output. We will denote the binary-outcome projective measurement $(U^\dagger|0\rangle\langle 0|U, U^\dagger|1\rangle\langle 1|U)$ as $\overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$, the projective version of $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$, which also has size $\mathrm{poly}(1/\epsilon, \log(1/\delta))$. By definition, for any quantum state ρ, the output distribution of running $\overline{\mathsf{ATI}}$ on ρ along with enough fresh ancillas is the same as the output distribution of running ATI on ρ (but the residual states are different).

Roughly speaking, $\overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$ does the same thing as $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$ except that it uncomputes intermediate results. Notice that a quantum query to function f is implemented as $U_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$, whose inverse is exactly U_f. We can use the same proof technique in Lemma 5.1 to show that $\overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$ can also be approximated by polynomial classical samples from D up to inverse polynomial precision and thus we can also apply hybrid arguments between $\overline{\mathsf{ATI}}_{\mathcal{P},D_0,\gamma}$ and $\overline{\mathsf{ATI}}_{\mathcal{P},D_1,\gamma}$ for computationally indistinguishable distributions D_0 and D_1. Formally,

Lemma 5.2. *Let $\mathcal{P}$ be a collection of projective measurements indexed by some set $\mathcal{I}$. Suppose $\mathcal{P}$ can be implemented by a quantum circuit of size $|\mathcal{P}|$. Let D_0, D_1 be two efficiently sampleable distributions over $\mathcal{I}$. For any state $\rho \in \mathcal{D}(\mathcal{H}_1)$, denote $(b,\rho') \leftarrow \overline{\mathsf{ATI}}^{\varepsilon,\delta}_{\mathcal{P},D,\gamma}(\rho)$ be the procedure that runs $\overline{\mathsf{ATI}}^{\varepsilon,\delta}_{\mathcal{P},D,\gamma}$ on state ρ along with enough fresh ancillas initialized to $|0\rangle$, and gets an output b*

and the post-measurement state ρ'. For any polynomial μ, any quantum state ρ and any (possibly quantum) predicate $h : \{0,1\} \times \mathcal{D}(\mathcal{H}_2) \to \{0,1\}$ with circuit size $|h|$, if

$$\left|\mathsf{Pr}\left[h(b,\rho')=1 \middle| (b,\rho') \leftarrow \overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P},D_0,\gamma}(\rho)\right] - \mathsf{Pr}\left[h(b,\rho')=1 \middle| (b,\rho') \leftarrow \overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P},D_1,\gamma}(\rho)\right]\right| \geq \frac{1}{\mu(\lambda)}.$$

Then there exists a quantum circuit $\mathcal{C}$ of size $\mathrm{poly}(\lambda, 1/\epsilon, \log(1/\delta), \mu, |\mathcal{P}|, |h|)$ (which only use the quantum circuits to implement $\mathcal{P}$, h and to sample D_0, D_1 as a black box)

$$\left|\mathsf{Pr}\left[\mathcal{C}(\rho,x)=b : \substack{b \xleftarrow{\$} \{0,1\} \\ x \sim D_b}\right] - \frac{1}{2}\right| \geq \frac{1}{(\mu(\lambda))^3 \cdot \mathrm{poly}(\lambda, 1/\epsilon, \log(1/\delta))}$$

which is an inverse polynomial if μ is a polynomial.

Proof. As the proof is almost the same as the proof of Lemma 5.1, we omit the proof. □

6 Almost Perfect Extraction of Preimages

In this section, we show how to extract a short preimage of $\mathbf{y}$ with overwhelming probability, given a good (quantum) distinguisher between the distribution of a ciphertext of message μ and a uniform distribution. Our main contribution is an extraction algorithm that is guaranteed to work with *overwhelming probability*, in contrast to the extraction algorithm in [APV23] that only works with probability inversely proportional to the field size.

Since a general quantum distinguisher can be a superposition of a good distinguisher and a useless distinguisher, we use (Approximate) Threshold Implementation to (approximately) test whether a given quantum distinguisher is good before we apply the extraction algorithm. We need the following notations before we formally define what is a good quantum distinguisher.

Threshold Implementation on a Quantum Distinguisher For a quantum algorithm $\mathcal{A}$ with auxiliary quantum states ρ, let two-outcome projective measurements $\{\mathcal{P}^{\mathcal{A}}_x = (P^{\mathcal{A}}_x, Q^{\mathcal{A}}_x)\}$ correspond to running $\mathcal{A}$ on x and the auxiliary state ρ. Suppose that $P^{\mathcal{A}}_x$ corresponds to outcome 1 and $Q^{\mathcal{A}}_x$ corresponds to outcome 0.

For two distributions D_0 and D_1, denote (D_0, D_1) to be the distribution of (b, x) where $b \xleftarrow{\$} \{0,1\}$ and $x \sim D_b$. We say that $(\mathcal{A}, \rho)$ is a *γ-good quantum distinguisher* for distributions D_0 and D_1 with support $\mathcal{X}$ if and only if ρ passes the projector $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}^{\mathcal{A}}_{(D_0,D_1)})$. Here, we abuse the notation to define the POVM $\mathcal{P}^{\mathcal{A}}_{(D_0,D_1)} = (P^{\mathcal{A}}_{(D_0,D_1)}, Q^{\mathcal{A}}_{(D_0,D_1)})$[5] such that

$$P^{\mathcal{A}}_{(D_0,D_1)} = \frac{P^{\mathcal{A}}_{D_1} + Q^{\mathcal{A}}_{D_0}}{2} = \frac{\sum_{x\in\mathcal{X}} \mathsf{Pr}[x \leftarrow D_1]\, P^{\mathcal{A}}_x + \sum_{x\in\mathcal{X}} \mathsf{Pr}[x \leftarrow D_0]\, Q^{\mathcal{A}}_x}{2},$$

$$Q^{\mathcal{A}}_{(D_0,D_1)} = I - P^{\mathcal{A}}_{(D_0,D_1)}.$$

In other words, $\mathcal{P}^{\mathcal{A}}_{(D_0,D_1)} = (P^{\mathcal{A}}_{(D_0,D_1)}, Q^{\mathcal{A}}_{(D_0,D_1)})$ is the POVM measurement (where $P^{\mathcal{A}}_{(D_0,D_1)}$ corresponds to output 1 and $Q^{\mathcal{A}}_{(D_0,D_1)}$ corresponds to output 0) that on any input quantum state ρ,

[5] $\mathcal{P}^{\mathcal{A}}_{(D_0,D_1)}$ is actually a mixture of projective measurements for the distribution (D_0, D_1) and a collection of binary outcome projective measurements $\mathcal{P}_{b,x} = (Q^{\mathcal{A}}_x, P^{\mathcal{A}}_x)$ if $b = 0$ and $\mathcal{P}_{b,x} = (P^{\mathcal{A}}_x, Q^{\mathcal{A}}_x)$ if $b = 1$.

- Sample $(b, x) \sim (D_0, D_1)$.
- Feed x and the input quantum state ρ into the algorithm $\mathcal{A}$, which outputs a guess b'.
- Output 1 if $b' = b$; 0 otherwise.

We denote the approximate version of $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}^{\mathcal{A}}_{(D_0,D_1)})$ as $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{A}},(D_0,D_1),1/2+\gamma}$. Roughly speaking, $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{A}},(D_0,D_1),1/2+\gamma}$ can efficiently estimate whether the algorithm $\mathcal{A}$, along with the input quantum state as auxiliary, can distinguish D_0 and D_1 with advantage at least γ. We denote the projective version of $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{A}},(D_0,D_1),1/2+\gamma}$ as $\overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{A}},(D_0,D_1),1/2+\gamma}$.

Some Important Distributions The threshold implementation will be used to test whether a quantum distinguisher works well on the following distributions. The prime modulus q, the noise ratios $\alpha, \beta \in (0,1)$ and $n, m \in N$ are all fixed parameters that will be soon clear from the context. For matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and vectors $\mathbf{y} \in \mathbb{Z}_q^n, \mathbf{x} \in \mathbb{Z}_q^m$,

- Denote $D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}}$ to be the distribution of $(\mathbf{A}, \mathbf{y}, \mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal, \mathbf{s}^\intercal\mathbf{y} + e')$ where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \sim D_{\mathbb{Z}_q^m,\alpha q}$ and $e' \sim D_{\mathbb{Z}_q,\beta q}$.
- Denote $D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}$ to be the distribution of $(\mathbf{A}, \mathbf{y}, \mathbf{u}^\intercal, u')$ where $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ and $u' \xleftarrow{\$} \mathbb{Z}_q$.
- Denote $\bar{D}^{\mathbf{A},\mathbf{x}}_{\mathsf{lwe}}$ to be the distribution of $(\mathbf{A}, \mathbf{A}\mathbf{x}, \mathbf{u}^\intercal, \mathbf{u}^\intercal\mathbf{x} + e')$ where $e' \sim D_{\mathbb{Z}_q,\beta q}$ and $\mathbf{u}^\intercal = \mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal$ for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \sim D_{\mathbb{Z}_q^m,\alpha q}$.
- Denote $D^{\mathbf{A},\mathbf{x}}_{\mathsf{gl}}$ to be the distribution of $(\mathbf{A}, \mathbf{A}\mathbf{x}, \mathbf{u}^\intercal, \mathbf{u}^\intercal\mathbf{x} + e')$ where $e' \sim D_{\mathbb{Z}_q,\beta q}$ and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$.

For each of the above distribution D, we denote $D(i, g_i)$ to be the distribution of $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 + c \cdot \hat{\mathbf{i}}^\intercal, v_4 + c \cdot g_i)$ where $\hat{\mathbf{i}}$ is the unit vector with its i^{th} coordinate being 1, c is sampled uniformly at random from $\mathbb{Z}_q$, and $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, v_4)$ is sampled from D. It is easy to generate a sample from $D(i, g_i)$ given i, g_i and a sample from D. Thus if we can efficiently distinguish between $D_0(i, g_i)$ and $D_1(i, g_i)$, then on input (i, g_i), we can efficiently distinguish between D_0 and D_1.

We show the following result.

Theorem 6.1 (Almost Optimal Search-to-Decision Reduction with Quantum Auxiliary Input)**.** *Let $n \in \mathbb{N}$ and q be a prime modulus with $q = 2^{o(n)}$ and let $m \geq 2n \log q$, each parameterized by the security parameter $\lambda \in \mathbb{N}$ such that $m \leq \mathrm{poly}(\lambda)$. Let $\sqrt{8m} < \sigma < q/\sqrt{8m}$ and let $\alpha, \beta \in (0,1)$ be noise ratios with $\beta/\alpha = 2^{o(n)}$, $2^{-o(n)} \leq \alpha\sigma \leq \mathsf{negl}(\lambda)$ and $\alpha\sigma/\beta \leq \mathsf{negl}(\lambda)$. Let $\mathcal{A} = \{(\mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}, \nu_\lambda)\}_{\lambda\in\mathbb{N}}$ be any non-uniform quantum algorithm consisting of a family of polynomial-sized quantum circuits and polynomial-sized advice states $\nu_\lambda \in \mathcal{D}(\mathcal{H}_{B_\lambda})$ which are independent of $\mathbf{A}$.*

Assume the decisional $\mathsf{LWE}^m_{n,q,\alpha q}$ cannot be solved by a quantum algorithm running in time $\mathrm{poly}(\lambda, \sigma)$ with distinguishing advantage $1/\mathrm{poly}(\lambda, \sigma)$. If there exist functions $\varepsilon(\lambda) = 1/\mathrm{poly}(\lambda)$, $\gamma(\lambda) = 1/\mathrm{poly}(\lambda)$, $\delta(\lambda) = 2^{-\Theta(\lambda)}$ and a QPT distinguisher $\mathcal{D}$ such that

$$\mathsf{Pr}\left[1 \leftarrow \mathsf{SearchToDecisionExpt}^{\mathcal{A},\mathcal{D}}(1^\lambda)\right] = \varepsilon(\lambda).$$

$\underline{\mathsf{SearchToDecisionExpt}^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)}$:

1. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
2. Generate $(|\psi_{\mathbf{y}}\rangle, \mathbf{y}) \leftarrow \mathsf{GenGauss}(\mathbf{A}, \sigma)$.
3. Generate $\rho_{R,\,\mathrm{AUX}} \leftarrow \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}| \otimes \nu_\lambda)$.
4. Let two-outcome projective measurements $\{\mathcal{P}_x^{\mathcal{D}}\}$ correspond to running $\mathcal{D}$ on samples x and the auxiliary state in AUX. We approximately test whether $(\mathcal{D}, \rho_{\mathrm{AUX}})$ is a γ-good quantum distinguisher between the distributions $D_{\mathsf{lwe}}^{\mathbf{A},\mathbf{y}}$ and $D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}$ by running $\mathsf{ATI}^{\gamma/6,\delta}_{\mathcal{P}^{\mathcal{D}},(D_{\mathsf{lwe}}^{\mathbf{A},\mathbf{y}},D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}),1/2+\gamma}$ on ρ_{AUX} and output the result.

Figure 2: The experiment $\mathsf{SearchToDecisionExpt}^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)$.

Then, there exists a quantum extractor $\mathcal{E}$ that takes as input $\mathbf{A}$, $\mathbf{y}$ and system AUX *of the state $\rho_{\mathrm{R,AUX}}$ and outputs a short vector in the coset $\Lambda_q^{\mathbf{y}}(\mathbf{A})$ in time $\mathrm{poly}(\lambda, \sigma, 1/\gamma)$ such that*

$$\Pr\left[\mathbf{x} \in \Lambda_q^{\mathbf{y}}(\mathbf{A}) \cap \mathcal{B}^m(\mathbf{0}, \sigma\sqrt{m/2}) \;:\; \begin{array}{c} \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m} \\ (|\psi_{\mathbf{y}}\rangle, \mathbf{y}) \leftarrow \mathsf{GenGauss}(\mathbf{A}, \sigma) \\ \rho_{\mathrm{R,AUX}} \leftarrow \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}| \otimes \nu_\lambda) \\ 1 \leftarrow \mathsf{ATI}^{\gamma/6,\delta}_{\mathcal{P}^{\mathcal{D}},(D_{\mathsf{lwe}}^{\mathbf{A},\mathbf{y}},D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}),1/2+\gamma}(\rho_{\mathrm{AUX}}) \\ \mathbf{x} \leftarrow \mathcal{E}(\mathbf{A}, \mathbf{y}, \mathrm{AUX}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

In other words,

$$\Pr\left[b = 1 \wedge \mathbf{x} \notin \Lambda_q^{\mathbf{y}}(\mathbf{A}) \cap \mathcal{B}^m(\mathbf{0}, \sigma\sqrt{m/2}) \;:\; \begin{array}{c} \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m} \\ (|\psi_{\mathbf{y}}\rangle, \mathbf{y}) \leftarrow \mathsf{GenGauss}(\mathbf{A}, \sigma) \\ \rho_{\mathrm{R,AUX}} \leftarrow \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}| \otimes \nu_\lambda) \\ b \leftarrow \mathsf{ATI}^{\gamma/6,\delta}_{\mathcal{P}^{\mathcal{D}},(D_{\mathsf{lwe}}^{\mathbf{A},\mathbf{y}},D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}),1/2+\gamma}(\rho_{\mathrm{AUX}}) \\ \mathbf{x} \leftarrow \mathcal{E}(\mathbf{A}, \mathbf{y}, \mathrm{AUX}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

6.1 Construction of the Extractor

In the subsection, we formally define our quantum extractor $\mathcal{E}$. $\mathcal{E}$ takes $\mathbf{A}, \mathbf{y}$ and the quantum state in AUX as input, and does the following:

$\underline{\mathcal{E}(\mathbf{A}, \mathbf{y}, \text{AUX})}$:

1. Set $\mathbf{x} = \mathbf{0}$, $\epsilon = \gamma/6$, $\gamma' = \gamma - 3\epsilon = \gamma/2$.
2. For each $i = 1, 2, \cdots, m$:

   For each $g_i \in [-\sigma\sqrt{m/2}, \cdots, \sigma\sqrt{m/2}]$:

   i. Let AUX store the current state of the quantum distinguisher.

   ii. Run $\overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{D}},\left(D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}}(i,g_i), D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}\right),1/2+\gamma'}$ on the residual state in register AUX along with enough fresh ancillas initialized to $|0\rangle$.

   iii. If it outputs 1, set $\mathbf{x}_i = g_i$, and move on to the next guess.

   iv. If it outputs 0, move on to the next guess.
3. Output $\mathbf{x}$.

Figure 3: The quantum extractor $\mathcal{E}(\mathbf{A}, \mathbf{y}, \text{AUX})$.

By construction, the extractor runs in time $\text{poly}(\lambda, \sigma, m, 1/\epsilon, \log\frac{1}{\delta}) = \text{poly}(\lambda, \sigma, 1/\gamma)$.

6.2 Analysis of the Extractor

Before we analyze the success probability of our extractor, we make crucial observations on the distributions $D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}}$, $D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}$, $\bar{D}^{\mathbf{A},\mathbf{x}}_{\mathsf{lwe}}$ and $D^{\mathbf{A},\mathbf{x}}_{\mathsf{gl}}$.

Lemma 6.2. *For any $\mathbf{x} \in \mathcal{B}^m(\mathbf{0}, \sigma\sqrt{m/2})$, the statistical distance between $\bar{D}^{\mathbf{A},\mathbf{x}}_{\mathsf{lwe}}$ and $D^{\mathbf{A},\mathbf{A}\mathbf{x}}_{\mathsf{lwe}}$ is at most $\mathsf{negl}(\lambda)$.*

Proof. For any $\mathbf{x} \in \mathcal{B}^m(\mathbf{0}, \sigma\sqrt{m/2})$, by noise smudging (Lemma 2.5), the statistical distance between the distribution $D^{\mathbf{A},\mathbf{A}\mathbf{x}}_{\mathsf{lwe}}$ and the distribution of $(\mathbf{A}, \mathbf{A}\mathbf{x}, \mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal, \mathbf{s}^\intercal\mathbf{A}\mathbf{x} + \mathbf{e}^\intercal\mathbf{x} + e')$ where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \sim D_{\mathbb{Z}_q^m,\alpha q}$ and $e' \sim D_{\mathbb{Z}_q,\beta q}$ is at most $\alpha\sigma m/\beta + 2^{-\Omega(\lambda)}$ because $|\mathbf{e}^\intercal\mathbf{x}| \geq \alpha q\sigma m$ with at most $2^{-\Omega(\lambda)}$ probability over the choice of $\mathbf{e}$ (from Lemma 2.3).

By our choice of parameters, $\alpha\sigma m/\beta + 2^{-\Omega(\lambda)} \leq \mathsf{negl}(\lambda)$. □

Lemma 6.3. *For integer $i \in [m]$ and $g_i = \mathbf{x}_i$, $D^{\mathbf{A},\mathbf{x}}_{\mathsf{gl}}(i, g_i) = D^{\mathbf{A},\mathbf{x}}_{\mathsf{gl}}$.*

For integer $i \in [m]$ and $g_i \neq \mathbf{x}_i$, $D^{\mathbf{A},\mathbf{x}}_{\mathsf{gl}}(i, g_i) = D^{\mathbf{A},\mathbf{A}\mathbf{x}}_{\mathsf{unif}}$.

Proof. This follows directly from definition and the fact that the distribution of $D^{\mathbf{A},\mathbf{x}}_{\mathsf{gl}}(i, g_i)$ is the distribution of

$$(\mathbf{A}, \mathbf{A}\mathbf{x}, \mathbf{u}^\intercal + c \cdot \hat{\mathbf{i}}^\intercal, \mathbf{u}^\intercal\mathbf{x} + e' + c \cdot g_i) = (\mathbf{A}, \mathbf{A}\mathbf{x}, \mathbf{u}^\intercal + c \cdot \hat{\mathbf{i}}^\intercal, (\mathbf{u}^\intercal + c \cdot \hat{\mathbf{i}}^\intercal)\mathbf{x} + e' + c \cdot (g_i - \mathbf{x}_i))$$

where $\hat{\mathbf{i}}$ is the unit vector with its i^{th} coordinate being 1, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$, $e' \sim D_{\mathbb{Z}_q,\beta q}$ and c is sampled uniformly at random from $\mathbb{Z}_q$.

Recall that q is a prime modulus, when $g_i \neq \mathbf{x}_i$, $c \cdot (g_i - \mathbf{x}_i)$ is a uniformly random element in $\mathbb{Z}_q$ when $c \xleftarrow{\$} \mathbb{Z}_q$. □

Now we are ready to prove Theorem 6.1.

Proof. To prove Theorem 6.1, it suffices to prove that

$$\mathsf{Pr}\left[1 \leftarrow \mathsf{Game}_0^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)\right] \leq \mathsf{negl}(\lambda)$$

where $\mathsf{Game}_0^{\mathcal{A},\mathcal{D}}$ is shown in Figure 4.

$\underline{\mathsf{Game}_0^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)}$:

1. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
2. Generate $(|\psi_{\mathbf{y}}\rangle, \mathbf{y}) \leftarrow \mathsf{GenGauss}(\mathbf{A}, \sigma)$.
3. Generate $\rho_{R,\,\mathrm{AUX}} \leftarrow \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}| \otimes \nu_\lambda)$.
4. Compute $b \leftarrow \mathsf{ATI}^{\gamma/6,\delta}_{\mathcal{P}^{\mathcal{D}},(D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}),1/2+\gamma}(\rho_{\mathrm{AUX}})$. Abort if $b = 0$.
5. Compute $\mathbf{x} \leftarrow \mathcal{E}(\mathbf{A}, \mathbf{y}, \mathrm{AUX})$.
6. Output 1 if $\mathbf{x} \notin \Lambda_q^{\mathbf{y}}(\mathbf{A}) \cap \mathcal{B}^m(\mathbf{0}, \sigma\sqrt{m/2})$; Otherwise, output 0.

Figure 4: The game $\mathsf{Game}_0^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)$.

Let's consider the following sequence of hybrid distributions.

H_0: This is the same as the game $\mathsf{Game}_0^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)$ defined in Figure 4.

H_1: This is the following distribution:

1. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
2. Sample a Gaussian vector $\mathbf{x}' \sim D_{\mathbb{Z}_q^m,\sigma/\sqrt{2}}$ and let $\mathbf{y} = \mathbf{A} \cdot \mathbf{x}' \bmod q$.
3. Generate $\rho_{R,\,\mathrm{AUX}} \leftarrow \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\mathbf{x}'\rangle\langle\mathbf{x}'| \otimes \nu_\lambda)$.
4. Compute $b \leftarrow \mathsf{ATI}^{\gamma/6,\delta}_{\mathcal{P}^{\mathcal{D}},(D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}),1/2+\gamma}(\rho_{\mathrm{AUX}})$. Abort if $b = 0$.
5. Compute $\mathbf{x} \leftarrow \mathcal{E}(\mathbf{A}, \mathbf{y}, \mathrm{AUX})$.
6. Output 1 if $\mathbf{x} \notin \Lambda_q^{\mathbf{y}}(\mathbf{A}) \cap \mathcal{B}^m(\mathbf{0}, \sigma\sqrt{m/2})$; Otherwise, output 0.

H_2: This is the following distribution:

1. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
2. Sample a Gaussian vector $\mathbf{x}' \sim D_{\mathbb{Z}_q^m,\sigma/\sqrt{2}}$ and let $\mathbf{y} = \mathbf{A}\cdot\mathbf{x}' \bmod q$.
3. Generate $\rho_{R,\,\text{Aux}} \leftarrow \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\mathbf{x}'\rangle\langle\mathbf{x}'| \otimes \nu_\lambda)$.
4. Compute $b \leftarrow \mathsf{ATI}^{\gamma/6,\delta}_{\mathcal{P}^{\mathcal{D}},(D_{\mathsf{gl}}^{\mathbf{A},\mathbf{x}'},D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}),1/2+\gamma}(\rho_{\text{Aux}})$. Abort if $b=0$.
5. Compute $\mathbf{x} \leftarrow \mathcal{E}(\mathbf{A},\mathbf{y},\text{Aux})$.
6. Output 1 if $\mathbf{x} \notin \Lambda_q^{\mathbf{y}}(\mathbf{A}) \cap \mathcal{B}^m(\mathbf{0},\sigma\sqrt{m/2})$; Otherwise, output 0.

$\mathsf{H}_{3,k}$: This is the following distribution (recall the description of $\mathcal{E}$ defined in Figure 3). It is replacing $\overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{D}},(D_{\mathsf{lwe}}^{\mathbf{A},\mathbf{y}}(i,g_i),D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}),1/2+\gamma'}$ in $\mathcal{E}$ with $\overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{D}},\left(D_{\mathsf{gl}}^{\mathbf{A},\mathbf{x}'}(i,g_i),D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}\right),1/2+\gamma'}$ one by one.

1. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
2. Sample a Gaussian vector $\mathbf{x}' \sim D_{\mathbb{Z}_q^m,\sigma/\sqrt{2}}$ and let $\mathbf{y} = \mathbf{A}\cdot\mathbf{x}' \bmod q$.
3. Generate $\rho_{R,\,\text{Aux}} \leftarrow \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\mathbf{x}'\rangle\langle\mathbf{x}'| \otimes \nu_\lambda)$.
4. Compute $b \leftarrow \mathsf{ATI}^{\gamma/6,\delta}_{\mathcal{P}^{\mathcal{D}},(D_{\mathsf{gl}}^{\mathbf{A},\mathbf{x}'},D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}),1/2+\gamma}(\rho_{\text{Aux}})$. Abort if $b=0$.
5. Set $\mathbf{x}=\mathbf{0}$, $\epsilon=\gamma/6$, $\gamma'=\gamma-3\epsilon=\gamma/2$, $t=0$.
6. For each $i=1,2,\cdots,m$:

   For each $g_i \in [-\sigma\sqrt{m/2},\cdots,\sigma\sqrt{m/2}]$:
   i. Let Aux store the current state of the quantum distinguisher. $t \leftarrow t+1$.
   ii. If $t \le k$, run $\overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{D}},\left(D_{\mathsf{gl}}^{\mathbf{A},\mathbf{x}'}(i,g_i),D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}\right),1/2+\gamma'}$ on the residual state in register Aux along with enough fresh ancillas initialized to $|0\rangle$. Otherwise, run $\overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{D}},\left(D_{\mathsf{lwe}}^{\mathbf{A},\mathbf{y}}(i,g_i),D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}\right),1/2+\gamma'}$ on the residual state in register Aux along with enough fresh ancillas initialized to $|0\rangle$.
   iii. If it outputs 1, set $\mathbf{x}_i = g_i$, and move on to the next guess.
   iv. If it outputs 0, move on to the next guess.
7. Output 1 if $\mathbf{x} \notin \Lambda_q^{\mathbf{y}}(\mathbf{A}) \cap \mathcal{B}^m(\mathbf{0},\sigma\sqrt{m/2})$; Otherwise, output 0.

We now show the following:

Lemma 6.4. *Assuming the quantum hardness of* $\mathsf{LWE}^m_{n,q,\alpha q}$*, the hybrids* H_0 *and* H_1 *are computationally indistinguishable,*

$$\mathsf{H}_0 \approx_c \mathsf{H}_1.$$

Proof. This follows directly from the Gaussian-collapsing property (Theorem 2.10).

By the Gaussian-collapsing property, assume the quantum hardness of $\mathsf{LWE}^m_{n,q,\alpha q}$,

$$\left(\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m},\ |\psi_{\mathbf{y}}\rangle = \sum_{\substack{\mathbf{x}\in\mathbb{Z}_q^m\\ \mathbf{A}\mathbf{x}=\mathbf{y}}} \rho_\sigma(\mathbf{x})\ |\mathbf{x}\rangle\,,\ \mathbf{y}\in\mathbb{Z}_q^n\right) \approx_c \left(\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m},\ |\mathbf{x}'\rangle\,,\ \mathbf{A}\cdot\mathbf{x}' \in \mathbb{Z}_q^n\right)$$

where $(|\psi_{\mathbf{y}}\rangle\,,\mathbf{y}) \leftarrow \mathsf{GenGauss}(\mathbf{A},\sigma)$ and $\mathbf{x}' \sim D_{\mathbb{Z}_q^m,\sigma/\sqrt{2}}$.

Thus for the non-uniform quantum algorithm $\mathcal{A}$,

$$\mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}|\otimes\nu_\lambda) \approx_c \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\mathbf{x}'\rangle\langle\mathbf{x}'|\otimes\nu_\lambda)$$

□

Lemma 6.5. *Assuming the quantum hardness of* $\mathsf{LWE}^m_{n,q,\alpha q}$*, the hybrids* H_1 *and* H_2 *are computationally indistinguishable,*

$$\mathsf{H}_1 \approx_c \mathsf{H}_2.$$

Proof. We prove the claim by contradiction.

Suppose H_1 and H_2 can be distinguished by a QPT algorithm $\mathcal{B}$ with advantage at least $1/\lambda^c$ for a constant $c>0$ and infinitely many λ. Fix one such λ.

By standard averaging argument, for at least $\frac{1}{2\lambda^c}$ fraction of $(\mathbf{A},\mathbf{x}')$ sampled according to $\mathbf{A}\xleftarrow{\$}\mathbb{Z}_q^{n\times m}$ and $\mathbf{x}'\sim D_{\mathbb{Z}_q^m,\sigma/\sqrt{2}}$, $\mathcal{B}$ can distinguish the result of running step 3-6 of H_1 on $(\mathbf{A},\mathbf{x}')$, and the result of running step 3-6 of H_2 on $(\mathbf{A},\mathbf{x}')$ with advantage at least $\frac{1}{2\lambda^c}$. Let's call those $(\mathbf{A},\mathbf{x}')$ good. Then from Lemma 5.1, there exists a quantum circuit $\mathcal{C}$ of size $\mathrm{poly}(\lambda,1/\epsilon,\log(1/\delta))$ such that for each good $(\mathbf{A},\mathbf{x}')$, $\mathcal{C}(\rho_{\mathrm{AUX}},\mathbf{A},\mathbf{x}',\cdot)$ can distinguish samples from $\left(D^{\mathbf{A},\mathbf{x}'}_{\mathsf{gl}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}\right)$ and samples from $\left(D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}\right)$ with advantage at least $\frac{1}{\mathrm{poly}(\lambda,1/\epsilon,\log(1/\delta))}$.

As we can sample $D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}$ by ourselves and $D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}}\approx_s \bar{D}^{\mathbf{A},\mathbf{x}'}_{\mathsf{lwe}}$ (from Lemma 6.2 and the choice of parameters), there exists a polynomial size quantum circuit $\mathcal{C}'$ such that for each good $(\mathbf{A},\mathbf{x}')$, $\mathcal{C}'(\rho_{\mathrm{AUX}},\mathbf{A},\mathbf{x}',\cdot)$ can distinguish samples from $D^{\mathbf{A},\mathbf{x}'}_{\mathsf{gl}}$ and samples from $\bar{D}^{\mathbf{A},\mathbf{x}'}_{\mathsf{lwe}}$ with advantage at least $1/\lambda^d$ for some constant $d>0$.

Recall that the only difference in $D^{\mathbf{A},\mathbf{x}'}_{\mathsf{gl}}$ and $\bar{D}^{\mathbf{A},\mathbf{x}'}_{\mathsf{lwe}}$ is whether $\mathbf{u}$ is sampled according to LWE or sampled uniformly. Now let's show how to leverage the fact to break $\mathsf{LWE}^m_{n,q,\alpha q}$ using this $\mathcal{C}'$ (Algorithm 2). Notice that for all the good $(\mathbf{A},\mathbf{x}')$, line 3 passes with noticeable probability (by averaging arguments over the eigenspaces) and the residual state after running ATI and obtaining outcome 1 is still a good distinguisher (by Lemma 2.14). So Algorithm 2 breaks decisional $\mathsf{LWE}^m_{n,q,\alpha q}$ efficiently if Lemma 6.5 doesn't hold.

Algorithm 2: An algorithm to break decisional $\mathsf{LWE}^m_{n,q,\alpha q}$ if Lemma 6.5 doesn't hold

Input : Matrix $\mathbf{A}\in\mathbb{Z}_q^{n\times m}$ and vector $\mathbf{u}\in\mathbb{Z}_q^n$ (and quantum advice ν_λ).
Output: 0 or 1 (guess whether $\mathbf{u}$ is sampled from uniform or according to $\mathsf{LWE}^m_{n,q,\alpha q}$)

1 Sample a vector $\mathbf{x}'\sim D_{\mathbb{Z}_q^m,\sigma/\sqrt{2}}$ and let $\mathbf{y}=\mathbf{A}\cdot\mathbf{x}' \bmod q$. If $\|\mathbf{x}'\|\geq\sigma\sqrt{m/2}$, output 0 or 1 uniformly at random and abort.
2 Generate $\rho_{R,\,\mathrm{AUX}}\leftarrow\mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\mathbf{x}'\rangle\langle\mathbf{x}'|\otimes\nu_\lambda)$.
3 Test whether $\mathcal{C}'(\rho_{\mathrm{AUX}},\mathbf{A},\mathbf{x}',\cdot)$ can be used to distinguish samples from $D^{\mathbf{A},\mathbf{x}'}_{\mathsf{gl}}$ and samples from $\bar{D}^{\mathbf{A},\mathbf{x}'}_{\mathsf{lwe}}$ with advantage at least $1/\lambda^d$ by running $\mathsf{ATI}^{1/\lambda^{d+1},\delta}$ on it with threshold $1/2+\frac{1}{4\lambda^d}$. If the ATI outputs 0 (it is not a good distinguisher), output 0 or 1 uniformly at random and abort.
4 Denote the residual state (if not abort) in register AUX as ρ'_{AUX}.
5 Sample $e'\sim D_{\mathbb{Z}_q,\beta q}$.Let $\mathbf{v}:=(\mathbf{A},\mathbf{A}\mathbf{x}',\mathbf{u}^\intercal,\mathbf{u}^\intercal\mathbf{x}'+e')$.
6 Run $\mathcal{C}'(\rho'_{\mathrm{AUX}},\mathbf{A},\mathbf{x}',\mathbf{v})$ and output the result.

This ends our proof of the claim. □

Lemma 6.6. *Assume that the decisional* $\mathsf{LWE}^m_{n,q,\alpha q}$ *cannot be solved by a quantum algorithm running in time* $\mathrm{poly}(\lambda,\sigma)$ *with distinguishing advantage* $1/\mathrm{poly}(\lambda,\sigma)$.

The probability that hybrid $\mathsf{H}_{3,k}$ *outputs 1 and the probability that hybrid* $\mathsf{H}_{3,k+1}$ *outputs 1 are* $\mathsf{negl}(\lambda)/\sigma$ *close. Formally, for* $0 \le k \le \sqrt{2}\sigma m^{3/2} - 1$,

$$|\Pr[\mathsf{H}_{3,k+1} = 1] - \Pr[\mathsf{H}_{3,k} = 1]| \le \mathsf{negl}(\lambda)/\sigma$$

Proof. The proof is the same with the proof of Lemma 6.5 except that we apply Lemma 5.2 instead of Lemma 5.1 and that we set the parameter μ in Lemma 5.2 as $1/\mathrm{poly}(\lambda,\sigma)$ instead of $1/\mathrm{poly}(\lambda)$. We omit the proof details. □

Lemma 6.7. $\mathsf{H}_{3,\sqrt{2}\sigma m^{3/2}}$ *outputs 1 with negligible probability.*

Proof. We first define $\mathsf{Game}_1^{\mathcal{A},\mathcal{D}}(1^\lambda)$ in Figure 5. It is the same as $\mathsf{H}_{3,\sqrt{2}\sigma m^{3/2}}$ except that it will output 1 if $\mathbf{x} \neq \mathbf{x}'$ (which is implied by $\mathbf{x} \notin \Lambda_q^{\mathbf{y}}(\mathbf{A}) \cap \mathcal{B}^m(\mathbf{0}, \sigma\sqrt{m/2})$), so to prove Lemma 6.7, it suffices to prove that $\mathsf{Game}_1^{\mathcal{A},\mathcal{D}}(1^\lambda)$ outputs 1 with negligible probability.

Notice that in step 6, we just apply a sequence of projective measurements $\overline{\mathsf{ATI}}$ and set each coordinate of $\mathbf{x}'$ based on the measurement outcomes. By Quantum Union Bound (Lemma 2.2), $\Pr\left[1 \leftarrow \mathsf{Game}_1^{\mathcal{A},\mathcal{D}}(1^\lambda)\right]$ can be bounded by a union of events that for $\mathbf{x}'$ sampled according to $D_{\mathbb{Z}_q^m,\sigma/\sqrt{2}}$, $\mathsf{SubGame}^{\mathcal{A},\mathcal{D}}(1^\lambda, i, g_i, \mathbf{x}')$ outputs 1:

$$\begin{aligned}
&\Pr\left[1 \leftarrow \mathsf{Game}_1^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)\right] \\
\le& 4\sum_{i=1}^{m}\sum_{g_i=-\sigma\sqrt{m/2}}^{\sigma\sqrt{m/2}} \Pr\left[1 \leftarrow \mathsf{SubGame}^{\mathcal{A},\mathcal{D}}\left(1^\lambda, i, g_i, \mathbf{x}'\right) : \mathbf{x}' \sim D_{\mathbb{Z}_q^m,\sigma/\sqrt{2}}\right]
\end{aligned}$$

where $\mathsf{SubGame}^{\mathcal{A},\mathcal{D}}(1^\lambda, i, g_i, \mathbf{x}')$ is defined in Figure 6.

Now let's show for any fixed $i, g_i, \mathbf{x}'$,

$$\Pr\left[1 \leftarrow \mathsf{SubGame}^{\mathcal{A},\mathcal{D}}\left(1^\lambda, i, g_i, \mathbf{x}'\right)\right] \le \mathsf{negl}(\lambda)/\sigma \tag{1}$$

Case 1: $g_i = \mathbf{x}'_i$ Consider the residual state ρ'_{AUX} of running step 3 and obtaining $b = 1$. From Lemma 2.14, running $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{D}},\left(D^{\mathbf{A},\mathbf{x}'}_{\mathsf{gl}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}\right),1/2+\gamma'}$ on ρ'_{AUX}, we will obtain 1 with probability at least $1 - 3\delta$.

From Lemma 6.3, when $g_i = \mathbf{x}'_i$, $D^{\mathbf{A},\mathbf{x}'}_{\mathsf{gl}}(i, g_i) = D^{\mathbf{A},\mathbf{x}'}_{\mathsf{gl}}$. Thus the output distribution of running $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{D}},\left(D^{\mathbf{A},\mathbf{x}'}_{\mathsf{gl}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}\right),1/2+\gamma'}$ on ρ'_{AUX} is exactly the same as the output distribution of running $\overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{D}},\left(D^{\mathbf{A},\mathbf{x}'}_{\mathsf{gl}}(i,g_i),D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}\right),1/2+\gamma'}$ on ρ'_{AUX}. Therefore,

$$\Pr\left[1 \leftarrow \mathsf{SubGame}^{\mathcal{A},\mathcal{D}}\left(1^\lambda, i, g_i, \mathbf{x}'\right)\right] \le 3\delta \le \mathsf{negl}(\lambda)/\sigma.$$

$\underline{\mathsf{Game}_1^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)}$:

1. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
2. Sample a Gaussian vector $\mathbf{x}' \sim D_{\mathbb{Z}_q^m,\sigma/\sqrt{2}}$ and let $\mathbf{y} = \mathbf{A}\cdot\mathbf{x}' \bmod q$.
3. Generate $\rho_{R,\,\mathrm{AUX}} \leftarrow \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\mathbf{x}'\rangle\langle\mathbf{x}'| \otimes \nu_\lambda)$.
4. Compute $b \leftarrow \mathsf{ATI}^{\gamma/6,\delta}_{\mathcal{P}^{\mathcal{D}},(D_{\mathsf{gl}}^{\mathbf{A},\mathbf{x}'},D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}),1/2+\gamma}(\rho_{\mathrm{AUX}})$. Abort if $b=0$.
5. Set $\mathbf{x}=\mathbf{0}$, $\epsilon=\gamma/6$, $\gamma'=\gamma-3\epsilon=\gamma/2$.
6. For each $i=1,2,\cdots,m$:

   For each $g_i \in [-\sigma\sqrt{m/2},\cdots,\sigma\sqrt{m/2}]$:
   i. Let AUX store the current state of the quantum distinguisher.
   ii. Run $\overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{D}},\left(D_{\mathsf{gl}}^{\mathbf{A},\mathbf{x}'}(i,g_i),D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}\right),1/2+\gamma'}$ on the residual state in register AUX along with enough fresh ancillas initialized to $|0\rangle$.
   iii. If it outputs 1, set $\mathbf{x}_i = g_i$, and move on to the next guess.
   iv. If it outputs 0, move on to the next guess.
7. Output 1 if $\mathbf{x}\neq\mathbf{x}'$; Otherwise, output 0.

Figure 5: The game $\mathsf{Game}_1^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)$.

Case 2: $g_i \neq \mathbf{x}'_i$ Again consider the residual state ρ'_{AUX} of running step 3 and obtaining $b=1$. From Lemma 6.3, when $g_i \neq \mathbf{x}'_i$, $D_{\mathsf{gl}}^{\mathbf{A},\mathbf{x}'}(i,g_i) = D_{\mathsf{unif}}^{\mathbf{A},\mathbf{A}\mathbf{x}'} = D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}$. Thus when running $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{D}},(D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}},D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}),1/2+\gamma'}$ on ρ'_{AUX}, we will obtain 1 with probability exactly $\mathsf{Pr}\left[1 \leftarrow \mathsf{SubGame}^{\mathcal{A},\mathcal{D}}\left(1^\lambda,i,g_i,\mathbf{x}'\right)\right]$.

As $\mathcal{P}^{\mathcal{D}}_{\left(D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}},D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}\right)}$ only has eigenvalue $1/2 < 1/2+\gamma'-\epsilon$ (any distinguisher cannot do better than guessing a random bit when facing $D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}$ and $D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}$), running $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{D}},(D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}},D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}),1/2+\gamma'}$ on any state, we cannot get 1 with probability greater than δ, which implies that

$$\mathsf{Pr}\left[1 \leftarrow \mathsf{SubGame}^{\mathcal{A},\mathcal{D}}\left(1^\lambda,i,g_i,\mathbf{x}'\right)\right] \leq \delta \leq \mathsf{negl}(\lambda)/\sigma.$$

$\underline{\mathsf{SubGame}^{\mathcal{A},\mathcal{D}}\left(1^\lambda, i, g_i, \mathbf{x}'\right)}$:

1. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$. Let $\mathbf{y} = \mathbf{A}\cdot\mathbf{x}' \bmod q$.
2. Generate $\rho_{R,\,\mathrm{AUX}} \leftarrow \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\mathbf{x}'\rangle\langle\mathbf{x}'| \otimes \nu_\lambda)$.
3. Compute $b \leftarrow \mathsf{ATI}^{\gamma/6,\delta}_{\mathcal{P}^{\mathcal{D}},(D_{\mathsf{gl}}^{\mathbf{A},\mathbf{x}'},D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}),1/2+\gamma}(\rho_{\mathrm{AUX}})$. Abort if $b=0$.
4. Set $\mathbf{x}=\mathbf{0}$, $\epsilon=\gamma/6$, $\gamma'=\gamma-3\epsilon=\gamma/2$.
5. Run $\overline{\mathsf{ATI}}^{\epsilon,\delta}_{\mathcal{P}^{\mathcal{D}},\left(D_{\mathsf{gl}}^{\mathbf{A},\mathbf{x}'(i,g_i)},D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}\right),1/2+\gamma'}$ on the residual state in register AUX along with enough fresh ancillas initialized to $|0\rangle$.
6. Output 1 if $\left(g_i \neq \mathbf{x}'_i \text{ and } \overline{\mathsf{ATI}} \text{ outputs } 1\right)$ or $\left(g_i = \mathbf{x}'_i \text{ and } \overline{\mathsf{ATI}} \text{ outputs } 0\right)$; otherwise, output 0.

Figure 6: The game $\mathsf{SubGame}^{\mathcal{A},\mathcal{D}}\left(1^\lambda, i, g_i, \mathbf{x}'\right)$.

Summing up Equation (1) and averaging over $\mathbf{x}'$, we can get that

$$\begin{aligned}
&\mathsf{Pr}\left[1 \leftarrow \mathsf{Game}_1^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)\right]\\
\leq& 4\sum_{i=1}^{m}\sum_{g_i=-\sigma\sqrt{m/2}}^{\sigma\sqrt{m/2}} \mathsf{Pr}\left[1 \leftarrow \mathsf{SubGame}^{\mathcal{A},\mathcal{D}}\left(1^\lambda, i, g_i, \mathbf{x}'\right) : \mathbf{x}' \sim D_{\mathbb{Z}_q^m,\sigma/\sqrt{2}}\right]\\
\leq& \mathsf{negl}(\lambda),
\end{aligned}$$

which ends the proof.

□

Recall that H_0 is the same as the game $\mathsf{Game}_0^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)$, Theorem 6.1 follows directly from Lemma 6.4, Lemma 6.5, Lemma 6.6 over $\Theta(\sigma m^{3/2})$ pairs of consecutive hybrids, Lemma 6.7 and the observation that $\mathsf{H}_2 = \mathsf{H}_{3,0}$.

□

7 Proof of Theorem 4.2

We prove by contradiction. Let $\mathcal{A}$ be a QPT adversary and suppose that

$$\left|\mathsf{Pr}\left[1 \leftarrow \mathsf{Expt}^{\Sigma,\mathcal{A}}(1^\lambda, 0)\right] - \mathsf{Pr}\left[1 \leftarrow \mathsf{Expt}^{\Sigma,\mathcal{A}}(1^\lambda, 1)\right]\right| = \epsilon(\lambda),$$

where $\epsilon(\lambda)$ is inverse polynomial, $\mathsf{Expt}^{\Sigma,\mathcal{A}}(1^\lambda, b)$ is defined as Figure 1 and $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Revoke})$. Without loss of generality, we assume that

$$\Pr\left[1 \leftarrow \mathsf{Expt}^{\Sigma,\mathcal{A}}(1^\lambda, 1)\right] - \Pr\left[1 \leftarrow \mathsf{Expt}^{\Sigma,\mathcal{A}}(1^\lambda, 0)\right] = \epsilon(\lambda).$$

Without loss of generality, we assume that the adversary submits $\mu = 0$. We decompose the adversary into two QPT algorithms $\mathcal{A}, \mathcal{D}$ where $\mathcal{A}$ takes the Gaussian coset state $|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}|$ and generates the state $\rho_{\mathrm{R,AUX}}$. After returning system R to the challenger, $\mathcal{D}$ takes ρ_{AUX} and responds to the challenge. Then $\mathcal{A}, \mathcal{D}$ satisfies

$$\Pr\left[1 \leftarrow \mathsf{SecurityExpt}^{\mathcal{A},\mathcal{D}}(1^\lambda, 1)\right] - \Pr\left[1 \leftarrow \mathsf{SecurityExpt}^{\mathcal{A},\mathcal{D}}(1^\lambda, 0)\right] = \epsilon(\lambda)$$

where $\mathsf{SecurityExpt}^{\mathcal{A},\mathcal{D}}$ is the experiment shown in Figure 7, because the inefficient revocation implements $\mathsf{Revoke}(\mathsf{MSK}, \mathsf{PK}, \sigma)$.

Now we apply approximate threshold implementation on the residual state ρ_{AUX}.

$\underline{\mathsf{SecurityExpt}^{\mathcal{A},\mathcal{D}}\left(1^\lambda, b\right)}$:

1. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
2. Generate $(|\psi_{\mathbf{y}}\rangle, \mathbf{y}) \leftarrow \mathsf{GenGauss}(\mathbf{A}, \sigma)$.
3. Generate $\rho_{R,\,\mathrm{AUX}} \leftarrow \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}| \otimes \nu_\lambda)$.
4. Apply inefficient revocation on system R, if it fails, output $\mathsf{Invalid}$.
5. If $b = 0$, sample $(\mathbf{A}, \mathbf{y}, \mathbf{u}^\intercal, u') \sim D_{\mathsf{lwe}}^{\mathbf{A},\mathbf{y}}$. If $b = 1$, sample $(\mathbf{A}, \mathbf{y}, \mathbf{u}^\intercal, u') \sim D_{\mathsf{unif}}^{\mathbf{A},\mathbf{y}}$.
6. Run $b' \leftarrow \mathcal{D}(\mathbf{A}, \mathbf{y}, \mathbf{u}^\intercal, u', \mathrm{AUX})$ and output b'.

Figure 7: The experiment $\mathsf{SecurityExpt}^{\mathcal{A},\mathcal{D}}\left(1^\lambda, b\right)$.

Lemma 7.1. *For adversary $\mathcal{A}, \mathcal{D}$ that satisfies*

$$\Pr\left[1 \leftarrow \mathsf{SecurityExpt}^{\mathcal{A},\mathcal{D}}(1^\lambda, 1)\right] - \Pr\left[1 \leftarrow \mathsf{SecurityExpt}^{\mathcal{A},\mathcal{D}}(1^\lambda, 0)\right] = \epsilon(\lambda),$$

it also satisfies

$$\Pr\left[1 \leftarrow \mathsf{ATISecurityExpt}^{\mathcal{A},\mathcal{D},\gamma}(1^\lambda)\right] \geq \frac{\epsilon(\lambda)}{4} - \mathsf{negl}.$$

for $\gamma = \frac{3\epsilon}{14}$ where $\mathsf{ATISecurityExpt}^{\mathcal{A},\mathcal{D}}$ is shown in Figure 8.

$\underline{\mathsf{ATISecurityExpt}^{\mathcal{A},\mathcal{D},\gamma}\left(1^\lambda, b\right)}$:

1. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
2. Generate $(|\psi_{\mathbf{y}}\rangle, \mathbf{y}) \leftarrow \mathsf{GenGauss}(\mathbf{A}, \sigma)$.
3. Generate $\rho_{R,\,\mathrm{AUX}} \leftarrow \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}| \otimes \nu_\lambda)$.
4. Apply inefficient revocation on system R, if it fails, output $\mathsf{Invalid}$.
5. Run $\mathsf{ATI}^{\frac{\gamma}{6},\delta}_{\mathcal{P}^{\mathcal{D}},(D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}),\frac{1}{2}+\gamma}$ where $\delta = 2^{-\Theta(\lambda)}$ on system AUX and output the result.

Figure 8: The experiment $\mathsf{ATISecurityExpt}^{\mathcal{A},\mathcal{D},\varepsilon}\left(1^\lambda, b\right)$.

Proof. Suppose that revocation succeeds with probability p. The residual state ρ_{AUX} satisfies

$$\mathbb{E}\left[\mathrm{Tr}\left[\mathcal{P}^{\mathcal{D}}_{(D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}})}\rho_{\mathrm{AUX}}\right]\middle|\text{Revocation succeeds on R}\right] \geq \frac{1}{2} + \frac{\epsilon}{2p}.$$

By averaging argument and the definition of threshold implementation Theorem 2.12,

$$\mathbb{E}\left[\mathrm{Tr}\left[\mathsf{TI}_{\frac{1}{2}+\frac{\epsilon}{4}}\left(\mathcal{P}^{\mathcal{D}}_{(D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}})}\right)\rho_{\mathrm{AUX}}\right]\middle|\text{Revocation succeeds on R}\right] \geq \frac{\epsilon}{4p}.$$

By Lemma 2.14, if we set $\delta = 2^{-\Theta(\lambda)}$ we have,

$$\begin{aligned}
&\mathsf{Pr}\left[\mathsf{ATI}^{\frac{\gamma}{6},\delta}_{\mathcal{P}^{\mathcal{D}},(D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}),\frac{1}{2}+\gamma}(\rho_{\mathrm{AUX}}) = 1\middle|\text{Revocation succeeds on R}\right]\\
=&\,\mathbb{E}\left[\mathrm{Tr}\left[\mathsf{ATI}^{\frac{\gamma}{6},\delta}_{\mathcal{P}^{\mathcal{D}},(D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}),\frac{1}{2}+\gamma}\rho_{\mathrm{AUX}}\right]\middle|\text{Revocation succeeds on R}\right]\\
\geq&\,\mathbb{E}\left[\mathrm{Tr}\left[\mathsf{TI}_{\frac{1}{2}+\frac{\epsilon}{4}}\left(\mathcal{P}^{\mathcal{D}}_{(D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}})}\right)\rho_{\mathrm{AUX}}\right]\middle|\text{Revocation succeeds on R}\right] - \delta\\
\geq&\frac{\epsilon}{4p} - \mathsf{negl}.
\end{aligned}$$

□

Using the above lemma we can construct algorithm 3 for solving $\mathsf{SIS}^m_{n,q,\sigma\sqrt{2m}}$ problem using the adversary $\mathcal{A}, \mathcal{D}$. As for our choice of parameters, the hardness of $\mathsf{LWE}^m_{n,q,\alpha q}$ implies the hardness of $\mathsf{SIS}^m_{n,q,\sigma\sqrt{2m}}$, Theorem 4.2 follows directly from the correctness of algorithm 3, which we show in

the following claim.

Algorithm 3: $\mathsf{SIS\_Solver}(\mathbf{A})$

Input: Matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$.

Output: Vector $\mathbf{x} \in \mathbb{Z}^m$.

1 Generate a Gaussian state $(|\psi_{\mathbf{y}}\rangle, \mathbf{y}) \leftarrow \mathsf{GenGauss}(\mathbf{A}, \sigma)$ with

$$|\psi_{\mathbf{y}}\rangle \;=\; \sum_{\substack{\mathbf{x}\in\mathbb{Z}_q^m \\ \mathbf{A}\mathbf{x}=\mathbf{y} \pmod q}} \rho_\sigma(\mathbf{x})\;|\mathbf{x}\rangle$$

for some vector $\mathbf{y} \in \mathbb{Z}_q^n$.

2 Run $\mathcal{A}$ to generate a bipartite state $\rho_{R,\mathrm{AUX}}$ in systems $\mathcal{H}_\mathrm{R} \otimes \mathcal{H}_\mathrm{AUX}$ with $\mathcal{H}_\mathrm{R} = \mathcal{H}_q^m$.

3 Measure system R in the computational basis, and let $\mathbf{x}_0 \in \mathbb{Z}_q^n$ denote the outcome.

4 Run $\mathsf{ATI}^{\gamma/6,\delta}_{\mathcal{P}^\mathcal{D},(D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}),1/2+\gamma}$ on system AUX, abort if the output is 0.

5 Run the extractor $\mathcal{E}(\mathbf{A}, \mathbf{y}, \mathrm{AUX})$ from Theorem 6.1, and let $\mathbf{x}_1 \in \mathbb{Z}_q^n$ denote the outcome.

6 Output the vector $\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_0$.

Claim 7.2. *Algorithm 3 solves* $\mathsf{SIS}^m_{n,q,\sigma\sqrt{2m}}$ *with inverse polynomial probability when* $\mathcal{A}, \mathcal{D}$ *is a successful adversary.*

$\underline{\mathsf{SimultExtractionExpt}^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)}$:

1. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
2. Generate $(|\psi_{\mathbf{y}}\rangle, \mathbf{y}) \leftarrow \mathsf{GenGauss}(\mathbf{A}, \sigma)$.
3. Generate $\rho_{R,\,\mathrm{AUX}} \leftarrow \mathcal{A}_{\lambda,\mathbf{A},\mathbf{y}}(|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}| \otimes \nu_\lambda)$.
4. Apply inefficient revocation on system R, if it fails, output $\mathsf{Invalid}$.
5. Run $\mathsf{ATI}^{\gamma/6,\delta}_{\mathcal{P}^\mathcal{D},(D^{\mathbf{A},\mathbf{y}}_{\mathsf{lwe}},D^{\mathbf{A},\mathbf{y}}_{\mathsf{unif}}),1/2+\gamma}$ on system AUX, abort if the output is 0.
6. Run the extractor $\mathcal{E}(\mathbf{A}, \mathbf{y}, \mathrm{AUX})$ from Theorem 6.1, and let $\mathbf{x}_1 \in \mathbb{Z}_q^n$ denote the outcome.
7. Output 1 if $\mathbf{x}_1 \in \Lambda_q^{\mathbf{y}}(\mathbf{A}) \cap \mathcal{B}^m(\mathbf{0}, \sigma\sqrt{m/2})$; Otherwise, output 0.

Figure 9: The experiment $\mathsf{SimultExtractionExpt}^{\mathcal{A},\mathcal{D}}\left(1^\lambda\right)$.

Proof. Suppose $\mathcal{A}, \mathcal{D}$ is a successful adversary. To show that algorithm 3 can obtain an short

solution $\mathbf{x}$ we prove the following two statements:

- The probability that on system AUX the extractor $\mathcal{E}$ extracts a short preimage $\mathbf{x}_1$ of $\mathbf{y}$ and revocation succeeds on R is inverse polynomial

$$\Pr\left[\mathsf{SimultExtractionExpt}^{\mathcal{A},\mathcal{D}}(1^\lambda)=1\right]=\frac{1}{\mathrm{poly}(\lambda)}.$$

- Suppose that revocation succeeds with probability $\varepsilon(\lambda)$ conditioned on the extraction being successful. Then instead of running revocation on R, if we measure register R in computational basis and obtain result $\mathbf{x}_0$, the probability that $\mathbf{x}_0$ is a short preimage of $\mathbf{y}$ that is different from $\mathbf{x}_1$ is $\varepsilon(\lambda)-\mathsf{negl}(\lambda)$ conditioned on the extraction being successful.

If we have both statements to be true, by standard probability arguments we prove the claim.
The first statement follows from Lemma 7.1 and Theorem 6.1. Consider the experiment $\mathsf{SimulExtradctionExpt}$ shown in Figure 9. Let $\mathsf{GoodDecryptor}$ denote the event that we pass the ATI test on step 5. Let $\mathsf{RevocationSuc}$ denote the event that the inefficient revocation succeeds on system R on step 4. Let $\mathsf{ExtractionSuc}$ denote the event that $\mathbf{x}_1$ is a short preimage of $\mathbf{y}$ on step 7. By Lemma 7.1,

$$\Pr\left[\mathsf{RevocationSuc}\wedge\mathsf{GoodDecryptor}\right]=\frac{1}{\mathrm{poly}(\lambda)}.$$

By Theorem 6.1,

$$\Pr\left[\mathsf{ExtractionSuc}\,|\,\mathsf{GoodDecryptor}\right]\geq 1-\mathsf{negl}(\lambda).$$

By basic probability calculation,

$$\Pr\left[\mathsf{RevocationSuc}\wedge\mathsf{ExtractionSuc}\right]=\frac{1}{\mathrm{poly}(\lambda)}.$$

Now we prove the second statement. We show that given a specific short preimage $\mathbf{x}_1$ of $\mathbf{y}$ and a state ρ_{R} such that revocation succeeds on R with probability $\varepsilon(\lambda)$, if we measure R under computational basis, we obtain a short preimage $\mathbf{x}_0$ of $\mathbf{y}$ that is different from $\mathbf{x}_1$ with probability $\varepsilon(\lambda)-\mathsf{negl}(\lambda)$. Define the set of short preimages $\mathcal{S}=\left\{\mathbf{x}\middle|\mathbf{A}\mathbf{x}=\mathbf{y},\|\mathbf{x}\|\leq\sigma\sqrt{\frac{m}{2}}\right\}$ and

$$|\psi'_{\mathbf{y}}\rangle=\left(\sum_{\mathbf{x}\in\mathcal{S}}\rho_{\frac{\sigma}{\sqrt{2}}}(\mathbf{x})\right)^{-\frac{1}{2}}\sum_{\mathbf{x}\in\mathcal{S}}\rho_\sigma(\mathbf{x})\,|\mathbf{x}\rangle$$

be a 'truncated' Gaussian coset state. Consider the following projectors

- $\Pi_0=\sum_{\mathbf{x}\in\mathcal{S},\mathbf{x}\neq\mathbf{x}_1}|\mathbf{x}\rangle\langle\mathbf{x}|$ is a projector that projects onto all short preimages we want.
- $\Pi_1=|\psi'_{\mathbf{y}}\rangle\langle\psi'_{\mathbf{y}}|$ is the approximate revocation projector. The trace distance between Π_1 and the actual revocation projector $|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}|$ is negligible by Lemma 2.3.

Suppose that $\mathbf{A}$ is a full-rank matrix, if $\mathrm{Tr}[|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}|\,\rho_{\mathrm{R}}]=\varepsilon$ we have

$$\begin{aligned}\mathrm{Tr}[\Pi_0\rho_{\mathrm{R}}]\geq&\mathrm{Tr}[\Pi_1\Pi_0\rho_{\mathrm{R}}]\\ \geq&\mathrm{Tr}[\Pi_1\rho_{\mathrm{R}}]-\mathsf{negl}(\lambda)\\ \geq&\mathrm{Tr}[|\psi_{\mathbf{y}}\rangle\langle\psi_{\mathbf{y}}|\,\rho_{\mathrm{R}}]-\mathsf{negl}(\lambda)\\ =&\varepsilon-\mathsf{negl}(\lambda).\end{aligned}$$

where the second inequality follows from Lemma 2.6. Note that $\mathbf{A}$ is full-rank with $1-\mathsf{negl}(\lambda)$ probability. Combine all arguments above, we proof this claim. □

8 Applications

Combining our result with [APV23], we obtain constructions for

- Public-Key Encryption with Classical Key Revocation.
- Key-Revocable Fully Homomorphic Encryption.
- Revocable Pseudorandom Functions.

8.1 Public-Key Encryption with Classical Key Revocation

A public-key encryption with classical key-revocation is a public-key encryption such that whenever we want to perform key revocation:

- The lessee runs Delete on its quantum secret key ρ_{SK} and produce a classical certificate π.
- The lessor runs Revoke on input π and output Valid if it is a valid certificate.

The security of such scheme captures the idea that if an adversary produces a certificate π that passes the revocation then the remaining adversary cannot distinguish between a ciphertext of chosen message from a random ciphertext. In [APV23], they built a public-key encryption with classical key-revocation assuming the security of key-revocable Dual-Regev encryption. Combine with our result, we obtain the following theorem.

Theorem 8.1. *Assuming the polynomial hardness of* LWE *with sub-exponential modulus. The scheme* $\mathsf{CRevDual} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Delete}, \mathsf{Revoke})$ *(Construction 2,[APV23]) is a secure public-key encryption with classical key-revocation (Definition 7.1,7.2, [APV23]).*

8.2 Key-Revocable Fully Homomorphic Encryption

A key-revocable fully homomorphic encryption is a fully homomorphic encryption with quantum key revocation just like the key-revocable Dual-Regev Encryption. In [APV23], they built a key-revocable fully homomorphic encryption assuming the security of key-revocable Dual-Regev encryption. Meanwhile, this construction can be adapted to feature classical revocation via techniques used in public-key encryption with classical key-revocation mentioned above. Combine with our result, we obtain the following theorem.

Theorem 8.2. *Assuming the polynomial hardness of* LWE *and* SIS *with sub-exponential modulus. The scheme* $\mathsf{RevDualGSW} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}, \mathsf{Revoke})$ *(Construction 3,[APV23]) is a secure key-revocable fully homomorphic encryption (Definition 5.3, [APV23]). Meanwhile, this construction can be adapted to feature classical revocation via (Construction 2,[APV23]).*

8.3 Revocable Pseudorandom Functions

A key-revocable (or simply, called revocable) pseudorandom function is a weak pseudorandom function with its evaluation key revocable. The μ-security of such scheme captures the idea that if the revocation succeeds, the remaining adversary cannot distinguish between μ images $y_1 = \mathsf{PRF}(x_1), y_2 = \mathsf{PRF}(x_2), \cdots, y_\mu = \mathsf{PRF}(y_\mu)$ from μ random preimages $x_1, x_2, \cdots, x_\mu$ and uniform random values $y_1, y_2, \cdots, y_\mu$. Meanwhile, this construction can also be adapted to feature classical revocation. Combine with our result, we obtain the following theorem.

Theorem 8.3. *Assuming the polynomial hardness of* LWE *and* SIS *with sub-exponential modulus. The scheme* (Gen, PRF, Eval, Revoke) *(Construction 5,[APV23]) is a* poly*-secure revocable* PRF *scheme (Definition 9.2, 9.3, [APV23]). Meanwhile, this construction can be adapted to feature classical revocation via (Construction 2,[APV23]).*

Acknowledgements

We thank Jiahui Liu for patiently answering our questions about her work [CGJL23] and thank anonymous reviewers for their helpful comments.

References

[Aar09] Scott Aaronson. "Quantum copy-protection and quantum money". In: *2009 24th Annual IEEE Conference on Computational Complexity*. IEEE. 2009, pp. 229–242 (cit. on p. 1).

[Aar16] Scott Aaronson. *The Complexity of Quantum States and Transformations: From Quantum Money to Black Holes*. 2016. arXiv: 1607.05256 [quant-ph] (cit. on p. 8).

[AGKZ20] Ryan Amos, Marios Georgiou, Aggelos Kiayias, and Mark Zhandry. "One-shot signatures and applications to hybrid quantum/classical authentication". In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 2020, pp. 255–268 (cit. on p. 1).

[Ajt96] Miklós Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*. Ed. by Gary L. Miller. ACM, 1996, pp. 99–108. DOI: 10.1145/237814.237838. URL: https://doi.org/10.1145/237814.237838 (cit. on p. 10).

[AKN+23] Shweta Agrawal, Fuyuki Kitagawa, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. "Public Key Encryption with Secure Key Leasing". In: *arXiv preprint arXiv:2302.11663* (2023) (cit. on pp. 3, 4).

[ALL+21] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. "New approaches for quantum copy-protection". In: *Annual International Cryptology Conference*. Springer. 2021, pp. 526–555 (cit. on p. 13).

[APV23] Prabhanjan Ananth, Alexander Poremba, and Vinod Vaikuntanathan. "Revocable Cryptography from Learning with Errors". In: *Theory of Cryptography: 21st International Conference, TCC 2023, Taipei, Taiwan, November 29–December 2, 2023, Proceedings, Part IV*. Taipei, Taiwan: Springer-Verlag, 2023, pp. 93–122. ISBN: 978-3-031-48623-4. DOI: 10.1007/978-3-031-48624-1_4. URL: https://doi.org/10.1007/978-3-031-48624-1_4 (cit. on pp. 2–6, 9–12, 14–16, 18, 32, 33).

[Ban93] W. Banaszczyk. "New bounds in some transference theorems in the geometry of numbers." In: *Mathematische Annalen* 296.4 (1993), pp. 625–636. URL: http://eudml.org/doc/165105 (cit. on p. 9).

[BDGM20] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. "Factoring and pairings are not necessary for io: Circular-secure lwe suffices". In: *Cryptology ePrint Archive* (2020) (cit. on p. 2).

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. "Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits". In: *Advances in Cryptology–EUROCRYPT 2014: 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings 33*. Springer. 2014, pp. 533–556 (cit. on p. 2).

[BI20] Anne Broadbent and Rabib Islam. "Quantum encryption with certified deletion". In: *Theory of Cryptography Conference*. Springer. 2020, pp. 92–122 (cit. on pp. 1, 2).

[BK22] James Bartusek and Dakshita Khurana. *Cryptography with Certified Deletion*. 2022. DOI: 10.48550/ARXIV.2207.01754. URL: https://arxiv.org/abs/2207.01754 (cit. on p. 2).

[BL20] Anne Broadbent and Sébastien Lord. "Uncloneable Quantum Encryption via Oracles". In: *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*. Ed. by Steven T. Flammia. Vol. 158. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020, 4:1–4:22. DOI: 10.4230/LIPIcs.TQC.2020.4 (cit. on p. 1).

[CGJL23] Orestis Chardouvelis, Vipul Goyal, Aayush Jain, and Jiahui Liu. "Quantum key leasing for pke and fhe with a classical lessor". In: *arXiv preprint arXiv:2310.14328* (2023) (cit. on pp. 3, 4, 6, 33).

[DGT+10] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. "Public-Key Encryption Schemes with Auxiliary Inputs". In: *Theory of Cryptography*. Ed. by Daniele Micciancio. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 361–381. ISBN: 978-3-642-11799-2 (cit. on pp. 3, 10).

[Die82] DGBJ Dieks. "Communication by EPR devices". In: *Physics Letters A* 92.6 (1982), pp. 271–272 (cit. on p. 1).

[Gao15] Jingliang Gao. "Quantum union bounds for sequential projective measurements". In: *Physical Review A* 92.5 (2015), p. 052331 (cit. on p. 8).

[GPV07] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. *Trapdoors for Hard Lattices and New Cryptographic Constructions*. Cryptology ePrint Archive, Report 2007/432. https://eprint.iacr.org/2007/432. 2007 (cit. on p. 10).

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions". In: *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 2008, pp. 197–206 (cit. on p. 2).

[HMNY21] Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. *Quantum Encryption with Certified Deletion, Revisited: Public Key, Attribute-Based, and Classical Communication*. 2021. arXiv: 2105.05393 [quant-ph] (cit. on p. 2).

[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: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 611–638 (cit. on p. 6).

[Mah18] Urmila Mahadev. "Classical Homomorphic Encryption for Quantum Circuits". In: *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*. Ed. by Mikkel Thorup. IEEE Computer Society, 2018, pp. 332–338. DOI: 10.1109/FOCS.2018.00039. URL: https://doi.org/10.1109/FOCS.2018.00039 (cit. on p. 2).

[MP11] Daniele Micciancio and Chris Peikert. *Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller*. Cryptology ePrint Archive, Report 2011/501. https://eprint.iacr.org/2011/501. 2011 (cit. on p. 11).

[MR07] Daniele Micciancio and Oded Regev. "Worst-Case to Average-Case Reductions Based on Gaussian Measures". In: *SIAM J. Comput.* 37.1 (2007), pp. 267–302. DOI: 10.1137/S0097539705447360. URL: https://doi.org/10.1137/S0097539705447360 (cit. on p. 10).

[MW05] Chris Marriott and John Watrous. *Quantum Arthur-Merlin Games*. 2005. arXiv: cs/0506068 [cs.CC] (cit. on p. 13).

[Por22] Alexander Poremba. *Quantum Proofs of Deletion for Learning with Errors*. 2022. DOI: 10.48550/ARXIV.2203.01610. URL: https://arxiv.org/abs/2203.01610 (cit. on pp. 6, 12).

[Qua20] Willy Quach. "UC-secure OT from LWE, revisited". In: *Security and Cryptography for Networks: 12th International Conference, SCN 2020, Amalfi, Italy, September 14–16, 2020, Proceedings 12*. Springer. 2020, pp. 192–211 (cit. on p. 2).

[Reg05] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Journal of the ACM* 56.6 (2005), 34:1–34:40. ISSN: 0004-5411. DOI: 10.1145/1568318.1568324 (cit. on pp. 10, 11).

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. *Efficient Public Key Encryption Based on Ideal Lattices*. Cryptology ePrint Archive, Paper 2009/285. https://eprint.iacr.org/2009/285. 2009. URL: https://eprint.iacr.org/2009/285 (cit. on p. 11).

[WZ82] William K Wootters and Wojciech H Zurek. "A single quantum cannot be cloned". In: *Nature* 299.5886 (1982), pp. 802–803 (cit. on p. 1).

[Yue14] Henry Yuen. "A quantum lower bound for distinguishing random functions from random permutations". In: *Quantum Inf. Comput.* 14.13-14 (2014), pp. 1089–1097. ISSN: 1533-7146 (cit. on p. 36).

[Zha12a] Mark Zhandry. *How to Construct Quantum Random Functions*. Cryptology ePrint Archive, Paper 2012/182. https://eprint.iacr.org/2012/182. 2012. URL: https://eprint.iacr.org/2012/182 (cit. on pp. 36, 37).

[Zha12b] Mark Zhandry. *Secure Identity-Based Encryption in the Quantum Random Oracle Model*. Cryptology ePrint Archive, Paper 2012/076. https://eprint.iacr.org/2012/076. 2012. URL: https://eprint.iacr.org/2012/076 (cit. on p. 37).

[Zha15] Mark Zhandry. "A note on the quantum collision and set equality problems". In: *Quantum Inf. Comput.* 15.7-8 (2015), pp. 557–567. ISSN: 1533-7146 (cit. on p. 36).

[Zha20] Mark Zhandry. *Schrödinger's Pirate: How To Trace a Quantum Decoder*. Cryptology ePrint Archive, Paper 2020/1191. https://eprint.iacr.org/2020/1191. 2020. URL: https://eprint.iacr.org/2020/1191 (cit. on pp. 6, 7, 13, 16, 17).

A Proof of Lemma 5.1

Proof. Consider the following hybrid argument.

H_0. Consider the probability

$$p_0 = \mathsf{Pr}\left[h(b,\rho') = 1 \middle| (b,\rho') \leftarrow \mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_0,\gamma}(\rho)\right],$$

H_1. Since D_0 is efficiently sampleable, we abuse the notation and view it as a sampling procedure that takes a uniform random seed from $\mathcal{R}$ and outputs an index in $\mathcal{I}$. We may set the size of $\mathcal{R}$ to be $2^{\Omega(\lambda)}$ which is exponential. Then P_{D_0} can be rewritten in the following form,

$$P_{D_0} = \frac{1}{|R|}\sum_{r\in\mathcal{R}} P_{D_0(r)}$$

Let Π be the set of permutations mapping $\mathcal{R}$ to itself. Let $\pi \in \Pi$ be a random permutation. For any function f mapping $\mathcal{R}$ to itself, define $D_0^f(r) = D_0(f(r))$. Consider the probability

$$p_1 = \mathsf{Pr}\left[h(b,\rho') = 1 \middle| \begin{matrix}\pi \xleftarrow{\$} \Pi \\ (b,\rho')\leftarrow \mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_0^\pi,\gamma}(\rho)\end{matrix}\right],$$

we have $p_0 = p_1$ since D_0^π and $D_0(\pi(r))$ are identical distributions.

H_2. Now we change the random permutation $\pi \in \Pi$ to small-range functions $\sigma \in \Sigma$ of Zhandry [Zha12a]. Let G be the set of functions mapping $\mathcal{R}$ to $[s]$ and F be the set of functions mapping $[s]$ to $\mathcal{R}$ where s is a parameter to be set later (it is a polynomial if μ is a polynomial). Define

$$\Sigma = F \circ G = \{f \circ g | f \in F, g \in G\}$$

be the set of small-range functions. Consider the probability

$$p_2 = \mathsf{Pr}\left[h(b,\rho') = 1 \middle| \begin{matrix}\sigma \xleftarrow{\$} \Sigma \\ (b,\rho')\leftarrow \mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_0^\sigma,\gamma}(\rho)\end{matrix}\right],$$

Let Φ be the set of random functions mapping $\mathcal{R}$ to itself. Yuen and Zhandry show the following:

Theorem A.1 ([Yue14, Zha15])**.** *For any quantum algorithm B making Q quantum oracle queries to $\pi \in \Pi$ and $\phi \in \Phi$. We have*

$$\left|\mathsf{Pr}\left[B^\pi() = 1 \middle| \pi \xleftarrow{\$} \Pi\right] - \mathsf{Pr}\left[B^\phi() = 1 \middle| \phi \xleftarrow{\$} \Phi\right]\right| \leq O(Q^3/|\mathcal{R}|).$$

Theorem A.2 ([Zha12a]). *For any quantum algorithm B making Q quantum oracle queries to $\phi \in \Phi$ and $\sigma \in \Sigma$. We have*

$$\left|\mathsf{Pr}\left[B^{\phi}() = 1 \middle| \phi \xleftarrow{\$} \Phi\right] - \mathsf{Pr}\left[B^{\sigma}() = 1 \middle| \sigma \xleftarrow{\$} \Sigma\right]\right| \leq O(Q^3/s).$$

Theorem A.1 and Theorem A.2 imply that $|p_1 - p_2| \leq O(Q^3/s)$ where $Q = \mathrm{poly}\left(1/\varepsilon, \log(1/\delta)\right)$ is the number of oracle queries for $\mathsf{ATI}^{\epsilon,\delta}$.

H_3. Let E be a $2Q$-wise independent set of functions. Let $\Sigma' = F \circ E$, consider the probability

$$p_3 = \mathsf{Pr}\left[h(b,\rho') = 1 \middle|_{(b,\rho') \leftarrow \mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_0^{\sigma'},\gamma}(\rho)}^{\sigma' \xleftarrow{\$} \Sigma'}\right],$$

Since $\mathsf{ATI}^{\epsilon,\delta}$ only makes Q queries, the following theorem implies that $|p_2 - p_3| \leq O(Q^3/|R|)$

Theorem A.3 ([Zha12b]). *For any quantum algorithm B making Q quantum oracle queries to $g \in G$ and $e \in E$. We have*

$$\left|\mathsf{Pr}\left[B^{e}() = 1 \middle| e \xleftarrow{\$} E\right] - \mathsf{Pr}\left[B^{g}() = 1 \middle| g \xleftarrow{\$} G\right]\right| \leq O(Q^3/\mathcal{R}).$$

H_4. We change D_0 to D_1. Consider the probability

$$p_4 = \mathsf{Pr}\left[h(b,\rho') = 1 \middle|_{(b,\rho') \leftarrow \mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_1^{\sigma},\gamma}(\rho)}^{\sigma \xleftarrow{\$} \Sigma}\right],$$

First note that H_3 and H_4 are efficient hybrids since $D_0(f(i))$ for $i \in [s]$ over $f \xleftarrow{\$} F$ is just a 'list' of s independent samples.

H_4- H_7. For $i \in \{0,1,2,3\}$, H_{4+i} is just H_{3-i} except for replacing D_0 with D_1.

Thus, for any polynomial s, $|p_0 - p_3| \leq O(Q^3/s)$ and $|p_4 - p_7| \leq O(Q^3/s)$. By our assumption that $|p_0 - p_7| = 1/\mu(\lambda)$ we have $|p_3 - p_4| \geq 1/\mu(\lambda) - O(Q^3/s)$ for any polynomial s. Then we apply hybrid argument on the each element of the 'list' $f \in F$. Let $\mathsf{H}_{3,i}$ for $i = 0, 1, 2, \cdots, s$ be the hybrid that is the same as H_3 except that we change $D_0(f(e(r)))$ to

$$D'(r) = \begin{cases} D_0(f(e(r))) & (e(r) \leq i) \\ D_1(f(e(r))) & (e(r) > i) \end{cases}$$

Since H_3 is identical to $\mathsf{H}_{3,0}$ and H_4 is identical to $\mathsf{H}_{3,s}$, there must exists i such that the gap between $\mathsf{H}_{3,i}$ and $\mathsf{H}_{3,i+1}$ is at least $1/(\mu(\lambda)s) - O(Q^3/s^2)$. Using the distinguisher obtain by this two hybrids, we can set $s = \Theta(Q^4(\mu(\lambda))^2)$ and obtain a distinguisher that distinguish between D_0 and D_1 with probability $\Theta\left(\frac{1}{\mu(\lambda))^3 Q^4}\right)$. Moreover, this distinguisher has circuit size $\mathrm{poly}(\lambda, Q, s, 1/\epsilon, \log(1/\delta), |\mathcal{P}|, |h|) = \mathrm{poly}(\lambda, \mu, 1/\epsilon, \log(1/\delta), |\mathcal{P}|, |h|)$. □