Quantum Key Leasing for PKE and FHE with a Classical Lessor

Orestis Chardouvelis[1], Vipul Goyal[2], Aayush Jain[3], and Jiahui Liu[4]

[1,2,3]Carnegie Mellon University
[2]NTT Research
[4]Massachusetts Institute of Technology

today

In this work, we consider the problem of secure key leasing, also known as revocable cryptography (Agarwal et. al. Eurocrypt' 23, Ananth et. al. TCC' 23), as a strengthened security notion to its predecessor put forward in Ananth et. al. Eurocrypt' 21. This problem aims to leverage unclonable nature of quantum information to allow a lessor to lease a quantum key with reusability for evaluating a classical functionality. Later, the lessor can request the lessee to provably delete the key and then the lessee will be completely deprived of the capability to evaluate. In this work, we construct a secure key leasing scheme to lease a decryption key of a (classical) public-key, homomorphic encryption scheme from standard lattice assumptions. Our encryption scheme is exactly identical to the (primal) version of Gentry-Sahai-Waters homomorphic encryption scheme with a carefully chosen public key matrix. We achieve strong form of security where:

- The entire protocol (including key generation and verification of deletion) uses merely classical communication between a *classical lessor (client)* and a quantum lessee (server).
- Assuming standard assumptions, our security definition ensures that every computationally bounded quantum adversary could only simultaneously provide a valid classical deletion certificate and yet distinguish ciphertexts with at most some negligible probability.

Our security relies on subexponential time hardness of learning with errors assumption. Our scheme is the first scheme to be based on a standard assumption and satisfying the two properties mentioned above.

The main technical novelty in our work is the design of an FHE scheme that enables us to apply elegant analyses done in the context of classical verification of quantumness from LWE (Brakerski et. al.(FOCS'18, JACM'21) and its parallel amplified version in Radian et. al.(AFT'21)) to the setting of secure leasing. This connection to classical verification of quantumness leads to a modular construction and arguably simpler proofs than previously known. An important technical component we prove along the way is an amplified quantum search-to-decision reduction: we design an extractor that uses a quantum distinguisher (who has an internal quantum state) for decisional LWE, to extract secrets with success probability amplified to almost one. This technique might be of independent interest.

1 Introduction

Quantum information, through principles such as the *no-cloning theorem*, offers excititng possibilities for which there are no classical counterparts. An active area of research are primitives such as copy protection [Aar09] and quantum money [Wie83], which simply can't be realized classically.

Copy protection, a task that captures a wide variety of unlconability related tasks, is aimed at leveraging the power of unclonable quantum information to prevent piracy of software functionalities. However, constructions of quantum copy protection have turned out to be notoriously hard – in fact, it is shown to be impossible in general [AP21] and only exist relative to structured oracles [ALL$^+$20] if we aim at a generic scheme secure for all copy-protectable functions. Only a handful of functions are known to be copy-protectable without using oracles in the construction, but either they are built from from very strong, less-standard cryptographic assumptions (e.g. post-quantum iO) [CLLZ21, LLQZ22] or the functionalities are very limited (point functions) [CMP20, BJL$^+$21].

Meaningful relaxations to copy protection come in the form of primitives that allow revocation of the quantum software via proofs of "destruction" of the quantum functionality after use. Thus, if the revocation or proof verifies, a computationally bounded user is deprived of any ability to evaluate the function in any meaningful ways.

In this work, we consider one such recent notion of secure key leasing [AKN$^+$23], also called key-revocable cryptography in the literature [APV23]. This notion is inspired by secure software leasing in [AP21], but possesses significantly stronger security guarantees [1].

In secure key leasing, a lessor (also called the client) leases a quantum state to a lessee (also called the server). The quantum state might allow the server to obtain a capability to compute an advanced classical function, such as decryption (in case of public-key encryption with secure key leasing), or PRF evaluation, for polynomially many times.

At a later point in time, the lessor may ask the lessee to return or destroy the key for reasons such as lessee's subscription expiring or the lessor(client) deciding to use another server. The notion of secure key leasing allows the lessee to "delete" the quantum key and create a deletion certificate which can be verified by the lessor. Once the deletion certificate is sent to the lessor and verified, the security guarantee says that the lessee provably loses the associated capability. This must hold even if the lessee is fully malicious and may make arbitrary attempts to copy the quantum state or learn the underlying classical functionality before the deletion. A number of recent papers have studied the notion of secure key leasing for primitives like PKE, FE, FHE and PRF [APV23, AKN$^+$23], in the context of a quantum lessor and quantum lessee with quantum communication. While the notion of secure software leasing is interesting on its own, remarkly it has also found an application in the seemingly unrelated context of *leakage or intrusion detection* [ÇGLR23] where an adversary breaks into a machine to steal a secret key and one would like to detect if such an attack happened.

Secure key leasing is a uniquely quantum phonomenon which is enabled by the no-cloning theorem, clearly impossible in the classical setting. Therefore, some quantum capabilities are necessary to enable it. However, a natural question is: *to what extent are quantum capabilities necessary? Do both the client and the server need quantum capabilities? Do we require quantum communication in*

[1] Our security notion is *stronger* and has more applicability than the previous, similar primitive called secure software leasing (SSL) [AP21]. In SSL the adversarial lessee is only semi-malicious, while in [AKN$^+$23, APV23] and our work it is fully malicious. See section 1.2 for detailed discussions.

addition to quantum computation? As discussed in [Shm22, CHV23], even in the future where full scale quantum computer come into the household and every user can maintain quantum memory, setting up large-scale end-to-end quantum communication is still expensive and lossy. Moreover, having a client operating only on classical resources and delegating the quantum work to a server is desirable in almost all sorts of circumstances.

A recent beautiful work of Ananth, Poremba, and Vaikuntanathan [APV23] used the dual Regev cryptosystem to build primitives such as public-key encryption (PKE), fully-homomorphic encryption (FHE), and pseudo random functions (PRFs) with key revocation. While their basic constructions require quantum communication as well as a quantum certificate of deletion, they can extend their constructions to achieve classical deletion certificates [APV23] as well. However a key limitation is their work is that the security only holds either based on an unproven conjecture, or only against an adversarial server (leasee) which outputs a valid deletion certificate with probability negligibly close to 1. Proving security against general adversaries based on standard assumptions was left as an open problem. In addition, in both works [AKN$^+$23, APV23], the client and the server must have both quantum capabilities and quantum communication is also necessary in the key generation process. This leads to the following natural open problem:

Can we build secure leasing for unlearnable functions e.g. PKE decryption, FHE decryption, and other cryptographic functionalities with a completely classical lessor/client? and more desirably, from standard assumptions?

If so, this would lead to constructions which require only bare minimum quantum infrastructure needed in our setting.

1.1 Our Results

We answer all the above questions affirmatively thus settling the open problem from [APV23]. We obtain the following results:

Theorem 1.1. *(Informal) Assuming the post-quantum subexponental hardness of Learning-with-Errors, there exists a secure key leasing scheme for PKE and FHE with a completely classical lessor/client.*

More specifically, from LWE, we can achieve secure leasing for FHE decryption circuit with following properties:

1. The key generation is a one-message protocol with a single classical message from the lessor to lessee.
2. Upon being asked to delete the key, the lessee will produce a classical certificate, and send it to the client. The client will perform a classical verification algorithm on the certificate.
3. As natural in the definition for secure key leasing of classical PKE, the encryption is a public, classical procedure. The decryption can be done quantumly (on the lessee/server's side) or classically (on the lessor/client's side)

A key contribution of our work is to connect our problem to classically verifiable proof of quantumness [BCM$^+$21]. The latter is a well developed research direction with beautiful ideas. This connection allows us to design modular constructions and write proofs of security that are arguably simpler, thanks to a number of technical ideas that we could rely on the literature in classical verification of quantum computation.

To show the security of our construction, we prove along the way a quantum search-to-decision reduction for LWE that extracts the secret *with the success probability boosted to almost 1*.

Theorem 1.2 (Informal)**.** *Given a quantum distinguisher with an auxiliary quantum state, that distinguishes LWE samples from random with inverse polynomial probability for some arbitrary secret, there exists an extractor running in time polynomial (in the the secret size, the distinguisher's time and inverse its advantage) that produces the secret with probability negligibly close to 1.*

To the best of our knowledge, this is the first search-to-decision reduction with an almost-perfect search when using quantum algorithms with auxiliary quantum states. To prove the above theorem, we rely on a measurement implementation on the distinguisher's quantum state [Zha20, ALL$^+$20] and make new observations about this measurement procedure, by aligning its properties with the features required by the classical LWE search-to-decision reduction by Regev [Reg05a].

We believe this technique and the new observations made maybe of independent interest.

1.2 Comparison with Related Works

Secure Key Leasing/Revocable PKE: Comparison with Existing Works Previously, secure circuit leasing for PKE was achieved in [AKN$^+$23]. [AKN$^+$23] achieved it by a clean compiler from any PKE scheme. But they need quantum communication and quantum computation on the lessor's side throughout the protocol, and there is no easy, generic way to turn their scheme into one with classical leaser with classical communication.

A similar result on revocable PKE, FHE and PRF to [APV23] was shown by the concurrent work from LWE with interesting ideas from quantum money combined with dual Regev PKE. However, they need new conjectures for enabling the security proof where adversaries only have inverse polynomial probability of revocation, otherwise they can only handle a weaker security where the adversary has to pass the revocation with probability almost 1 [APV23]. Whereas we achieve the stronger definition, i.e. adversaries only need to pass the revocation test with noticeable probability, from LWE alone. Moreover, they also need the lessor to prepare the quantum key and send the key via a quantum channel.

Our techniques also differ greatly from the above works. Our main insight is to manipulate the primitives used in the classical verification of quantum computation protocol and turning it into an FHE scheme from LWE that supports leasing.

Secure Software Leasing: SSL, short for Secure Software Leasing in [AP21] is another relaxation of quantum copy protection. It is a weaker security notion than the notion studied in this work, even though bearing a similar name. In SSL, after revocation/deletion, one tests whether the leftover state of the adversary function correctly, using the leaser's circuit. In other words, in SSL, the pirate can be arbitrarily malicious in copying the function, but the free-loaders of the pirated programs are semi-honest by following the authentic software's instructions. In our work, we allow the adversarial state to be run by any adversarial circuit; as long as it gives the correct output, we count it as a successful attack., i.e. both the pirate and free-loaders can have arbitrary malicious behaviors.

We defer more related works and discussions on SSL to Section 2.2.

Post-quantum LWE Search-to-Decision Reduction with Quantum Auxiliary Input [BBK22] showed a general method to lift classical reductions to post-quantum reductions for a class of primitives satisfying certain constraints. Even though the classical search-to-decision reduction for LWE fit into the framework required in the above lifting theorem, our proof roadmap requires a stronger statement than the one shown in [BBK22].To the best of our knowledge, there is no generic or black-box method to amplify the reduction in [BBK22] for our use. Besides, our techniques differ from [BBK22] in that we do not perform the more involved "state repair" procedure used by [BBK22], which was inspired by [CMSZ22]. Therefore, we believe our techniques may be of independent interest and can shed light on new observations in quantum reductions.

Another work [STHY23] presents a quantum LWE search to decision reduction (with the possibility of amplification) but their quantum distinguisher is simply a unitary that does not have an auxiliary state. Therefore, it is incomparable to our techniques. Our reduction is more general and more challenging to achieve due to the potentially destructible auxiliary quantum state.

2 Technical Overview

We start by recalling the definition of Public-Key Encryption with Secure Key Leasing (also referred to as PKESKL from here on). (Levelled) homomorphic encryption with secure key leasing is defined analogously. In a PKESKL scheme with classical leaser, we have algorithms (Setup,KeyGen, Enc, Dec, Del, VerDel). The Setup algorithm (to be run by the lessor) takes in a security parameter and outputs a classical master public-key mpk and a classical trapdoor td. The KeyGen algorithm (to be run by the lessee) takes input mpk and outputs a classical public-key pk and a quantum secret key ρ_{sk}. Enc is defined in a usual way, it is a classical procedure that takes as input the key pk and a bit μ and produces a ciphertext ct. The quantum procedure Dec takes as input a ciphertext ct and the state ρ_{sk} and should produce the message that was encrypted. Moreover, the decryption should keep the state ρ_{sk} statistically close to the initial state as long as the decrypted ciphertext were not malformed. Then, the deletion algorithm Delete should take in the state ρ_{sk} and produce a classical certificate cert. Finally, the certificate can be verified by the classical algorithm VerDel using pk, and the trapdoor td as other inputs. Conclusively, our scheme will allow the generation of key ρ_{sk} by a protocol involving merely classical effort on the part of a client (lessor) and quantum effort from the server (lessee).

Our security requirement is also fairly intuitive. In the security game, the challenger generates a classical mpk along with a trapdoor td and gives mpk to a quantum polynomial time attacker $\mathcal{A}$. $\mathcal{A}$ generates a quantum decryption key ρ_{sk} and its associated public key pk. $\mathcal{A}$ publishes its pk. The challenger then asks $\mathcal{A}$ to provide a certificate cert. The game aborts if cert does not verify. However, if the certificate verifies, we require that $\mathcal{A}$ should not be able to distinguish an encryption of 0 from an encryption of 1 with a non-negligible probability. We formalize this by requiring that after the step of successful verification of certificate, an adversary outputs a quantum state ρ_{Delete} as a adversarial quantum decyrptor.

Checking a quantum state 's success probability can be difficult due to its destructible and unclonable nature. To check if this state is a good decryptor, we define a special binary-outcome projective measurement, called Threshold Implementation, $\mathsf{TI}_{\frac{1}{2}+\gamma}$ (from [Zha20, ALL^{+}20]). This measurement projects the state ρ_{Delete} onto the subspace of all possible states that are good at distinguishing ciphertexts with probability at least $\frac{1}{2}+\gamma$. If $\mathsf{TI}_{\frac{1}{2}+\gamma}$ outputs 1 on ρ_{Delete} with some noticeable probability, it implies that a noticeable "fraction" of the quantum states can distinguish ciphertexts with probability at least $1/2 + \gamma$.

Passing the $\mathsf{TI}_{1/2+\gamma}$ test combined with handing in a valid certificate, we consider such an adversary to be successful.

The definitions are inspired from the framework introduced by Zhandry [Zha20](see Section 8.2,Section 5 for details) and will imply the past PKE-SKL definition used in [AKN$^+$23, APV23]. As we will discuss later, defining the security via Threshold Implementation also assists us in our security proof.

We then start by describing the ideas to build a PKE scheme first. These ideas can be lifted to build a (levelled) FHE scheme. Below, we show a series of ideas leading up to our final scheme.

Starting Point: Regev PKE. Our starting point is a public-key encryption due to Regev [Reg05a], which is also the basis of many FHE schemes such as the BV scheme [BV11], and the GSW scheme [GSW13]. In Regev PKE, the public key consists of a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ where $m = \Omega(n \cdot \log q)$, along with m LWE samples $\mathbf{b} = \mathbf{s}\mathbf{A} + \mathbf{e} \mod q$ where $\mathbf{s} \leftarrow \mathbb{Z}_q^{1 \times n}$ is a randomly chosen and $\mathbf{e}$ is a discrete Gaussian distributed error with some parameter $\sigma \ll q$. The classical secret key is simply the secret vector $\mathbf{s}$ for the LWE sample.

To encrypt a message $\mu \in \{0,1\}$, one samples a binary random vector $\mathbf{r} \in \mathbb{Z}_q^{m \times 1}$. Encryptor then computes $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$ where $\mathsf{ct}_1 = (\mathbf{A} \cdot \mathbf{r}, \mathsf{ct}_2 = \mathbf{b} \cdot \mathbf{r} + \mu \cdot \lceil \frac{q}{2} \rceil)$. Observe that the correctness follows from the fact that $\mathbf{b} \cdot \mathbf{r} \approx \mathbf{s} \cdot \mathbf{A} \cdot \mathbf{r}$, where $\approx$ means that they are close to each other in the ℓ_2 norm, since $\mathbf{r}$ and $\mathbf{e}$ are both low-norm. The security follows from the fact that due to LWE, $\mathbf{b}$ is pseudorandom. Therefore, one can apply leftover hash lemma argument to that μ is hidden. We want to turn this into a scheme that supports leasing. Namely, instead of having the secret key $\mathbf{s}$, we would like to have it encoded as a quantum state that somehow encodes s and (say) another hidden value (for the deletion certificate) so that it's computationally hard for an attacker to produce both? Unfortunately, it's not so clear how to make things work if the scheme has exactly one key s.

Two Key Regev PKE. Our next idea is to modify the Regev style PKE scheme to work with two secret keys $\mathbf{s}_0, \mathbf{s}_1$ instead. The public-key would now consist of a matrix $\mathbf{A}$ along with two LWE samples $\mathbf{b}_0 = \mathbf{s}_0\mathbf{A} + \mathbf{e}_0$ and $\mathbf{b}_1 = \mathbf{s}_1\mathbf{A} + \mathbf{e}_1$. To generate an encryption of $\mu \in \{0,1\}$ one outputs $\mathsf{ct} = (\mathsf{ct}', \mathsf{ct}_0, \mathsf{ct}_1)$ where $\mathsf{ct}' = \mathbf{A}\mathbf{r}$, $\mathsf{ct}_i = \mathbf{b}_i\mathbf{r} + \mu \cdot \lceil \frac{q}{2} \rceil$ for $i \in \{0,1\}$. Observe that now the ciphertext ct could be decrypted using either of the two secret vectors $\mathbf{s}_0$ and $\mathbf{s}_1$. This gives us the following idea: perhaps one could encode the secret key as a superposition of the two keys, $\rho_{\mathsf{sk}} = |0, \mathbf{s}_0\rangle + |1, \mathbf{s}_1\rangle$ (ignoring normalization). Observe that using such a state one could decrypt any honestly generated ciphertext without disturbing the state.

What is a good candidate for a deletion certificate? Naturally, in order to delete the secret information in the computational basis, a deletion certificate could be a measurement in the Hadamard basis, giving a string $\mathbf{d}$ and a value $\alpha = \langle \mathbf{d}, \mathbf{s}_0 \oplus \mathbf{s}_1 \rangle$. We could perhaps hope that if an algorithm supplies the measurement outcome in the Hadamard basis correctly with overwhelming probability, then it would destroy information on the two secret keys.

How would one analyze such a scheme? The hope would be to arrive at a contradiction from such an adversary. Assuming that such an adversary outputs $(\alpha, \mathbf{d})$ with overwhelming probability and conditioned on that distinguishes encryptions of 0 from 1 with inverse polynomial probability, we would hope to extract one of the secrets $\mathbf{s}_0$ or $\mathbf{s}_1$ with inverse polynomial probability, from the adversary's internal quantum state. Simultaneously, we also have obtained the string $(\alpha, \mathbf{d})$ from the certificate handed in. We would like to argue that no efficient adversary is able to produce both a secret vector and a string $(\alpha, \mathbf{d})$. While this is often a reasonable property, subtleties can come in when combined with other structures of our encryption scheme. We instead turn to a very similar setting that has been extensively analyzed in prior works in context of classical proof of quantumness and certified randomness [BCM$^+$21]. This will save us the effort of reinventing the wheel.

Inspiration from Classical Verification of Quantum Computation. Let us now recall the very similar setting in [BCM$^+$21] which constructed proofs of quantumness from LWE. They build *Noisy Claw-free Trapdoor families (NTCF)* from LWE and show that a related task above is hard. Without going into the unnecessary semantics, we describe the setting relevant to us.

Consider the following game: The challenger samples $\mathbf{A}$ and an LWE sample $\mathbf{k} = \mathbf{s}\mathbf{A} + \mathbf{e} \mod q$ where $\mathbf{s}$ is (say) binary random secret $\mathbf{s}$ and $\mathbf{e}$ is chosen from a discrete Gaussian distribution with parameter σ. For any such sample, there exists a BQP algorithm that could sample a vector $\mathbf{y} = \mathbf{x}_0\mathbf{A} + \mathbf{e}'$ where $\mathbf{e}'$ chosen from discrete Gaussian with parameter $\sigma' \gg \sigma$ (by a superpolynomial factor), along with a special state ρ_{sk}. The state is of the form $\rho_{\mathsf{sk}} = |(0, \mathbf{x}_0)\rangle + |(1, \mathbf{x}_1)\rangle$ where $\mathbf{x}_1 = \mathbf{s} - \mathbf{x}_0$. Measuring this state yields either $\mathbf{x}_0$ so that $\mathbf{x}_0\mathbf{A} \approx \mathbf{y}$ or $\mathbf{x}_1$ so that $\mathbf{x}_1\mathbf{A}$ is close to $\mathbf{y} - \mathbf{k}$. On the other hand if we measured in the Hadamard basis we will again obtain $(\mathbf{d}, \alpha)$ for a random string $\mathbf{d}$ so that $\alpha = \langle \mathbf{d}, \mathbf{x}_0 \oplus \mathbf{x}_1 \rangle$ [2].

The test of quantumness asks an efficient quantum algorithm to produce a valid LWE sample $\mathbf{y} = \mathbf{x}_0\mathbf{A} + \mathbf{e}'$ of the kind above and based on a random challenge produce either one of $(\mathbf{x}_0, \mathbf{x}_1)$ for $b \in \{0, 1\}$ or a "valid" tuple $(\mathbf{d}, \alpha)$ as above.

This above protocol has a quantum correctness and a classical soundness, which we omit discussions due to irrelevance. However, to enable capabilities such as certified randomness/verification of quantum computation[Mah18], they show a more advanced soundness property against quantum provers. That is, even a BQP adversary cannot *simultaneously* produce these responses (the "computational response" for $b = 0$, and the "Hadamard response" for $b = 1$), assuming quantum security of LWE.

In short, we conclude the properties we need from the NTCF-based protocol: any QPT algorithm, given $\mathbf{A}$ and $\mathbf{k} = \mathbf{s}\mathbf{A} + \mathbf{e}$, can produce a valid LWE sample $\mathbf{x}_0\mathbf{A} + \mathbf{e}' \approx \mathbf{x}_0\mathbf{A} + \mathbf{e}' + \mathbf{k} = \mathbf{y}$. If it is asked to output *either*, (1) one of $\mathbf{x}_0, \mathbf{x}_1$, *or*, (2) a non-trivial pair $(\mathbf{d}, \alpha)$ such that $\alpha = \langle \mathbf{d}, \mathcal{J}(\mathbf{x}_0) \oplus \mathcal{J}(\mathbf{x}_1) \rangle$, it can do so with probability 1. However, if it is asked to output *both* (1) and (2) at once for the same $\mathbf{y}$, it cannot achieve so with probability noticeably higher than trivial guessing for one of them.

From Noisy Trapdoor Claw-free Families to Our Scheme. Now we gradually approach a scheme similar to the two-key Regev PKE scheme which can benefit from the NTCF security properties described above. The idea is that we have a public key $\mathbf{A}$, an LWE sample $\mathbf{k} = \mathbf{s}\mathbf{A} + \mathbf{e}$ where $\mathbf{s}$ is a random binary secret and $\mathbf{e}$ is Gaussian distributed exactly like the distribution used for proof of quantumness. Along with $\mathbf{k}$, the public key also consists of an LWE sample $\mathbf{y} = \mathbf{x}_0\mathbf{A} + \mathbf{e}'$ chosen as per the distribution in the proof of quantumness again. The client manintains a trapdoor td: td

[2] In the actual protocol, α is not equal to $\langle \mathbf{d}, \mathbf{s} \rangle$ where $\mathbf{s} = \mathbf{x}_0 \oplus \mathbf{x}_1$. In fact, we first apply a binary decomposition to $\mathbf{x}_0, \mathbf{x}_1$ and then measure in Hadamard basis. i.e. we have $\alpha = \langle \mathbf{d}, \mathcal{J}(\mathbf{x}_0) \oplus \mathcal{J}(\mathbf{x}_1) \rangle \mod 2$, where $\mathcal{J}$ is the invertible binary decomposition function. Such a binary decomposition is needed for the security proof of the NTCF to go through in [BCM$^+$21]

consists of a trapdoor matrix $\mathbf{T}$ of $\mathbf{A}$ that can be generated at the same time as sampling $\mathbf{A}$ (recall that td will be used to verify the deletion certificate later). The leased decryption state ρ_{sk} is the superposition $|0, x_0\rangle + |1, x_1\rangle$ where $\mathbf{x}_1 = \mathbf{x}_0 - \mathbf{s}$.

To encrypt a bit μ, one samples a random binary string $\mathbf{r}$ and computes a ciphertext $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3)$ where $\mathsf{ct}_1 = \mathbf{Ar}$, $\mathsf{ct}_2 = \mathbf{kr}$ and $\mathsf{ct}_3 = \mathbf{yr} + \mu\lceil\frac{q}{2}\rceil$. Observe that the ciphertext can be decrypted by both $\mathbf{x}_0$ or $\mathbf{x}_1$. This is because $\mathsf{ct}_3 - \mathbf{x}_0\mathbf{A}$ is close to $\mu\lceil\frac{q}{2}\rceil$ and similarly, $-\mathsf{ct}_2 + \mathsf{ct}_3 + \mathbf{x}_1\mathbf{A}$ is also close to $\mu\lceil\frac{q}{2}\rceil$. Thus, there is a way to decrypt a ciphertext cohenrently without disturbing the state ρ_{sk}, by the gentle measurement lemma [Aar04]. Accordingly, we set the deletion certificate to be the string $(\alpha, \mathbf{d})$ to use the property of NTCF.

Security: First Attempt We first discuss a very weak definition of security which the above simple scheme (almost) satisfies already. In this definition, a successful BQP adversary provides a deletion certificate that's valid with probability $1 - \mathsf{negl}$ for some negligible negl and then conditioned on that distinguishes ciphertexts with a noticable probability. One observation is that given a decryptor that's capable of distinguishing encryptions of zero from encryptions of one, must also enable distinguishing ciphertexts of zero of the form $(\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3)$ where $\mathsf{ct}_1 = \mathbf{A} \cdot \mathbf{r}$, $\mathsf{ct}_2 = \mathbf{k} \cdot \mathbf{r}$ and $\mathsf{ct}_3 = \mathbf{y} \cdot \mathbf{r}$ from truly random strings with a noticable probabilty. This means that this distinguisher distinguishes samples of the form $(\mathbf{Ar}, \mathbf{kr}, \mathbf{yr})$ from random. Since $\mathbf{k}$ is pseudorandom due to LWE it must also distinguish $(\mathbf{Ar}, \mathbf{kr}, \mathbf{yr})$ from random where $\mathbf{k}$ is chosen at random. Observe that $\mathbf{y} = \mathbf{x}_0\mathbf{A} + \mathbf{e}'$, $\mathbf{yr} = \mathbf{x}_0\mathbf{Ar} + \mathbf{e}'\mathbf{r}$. Given that $\mathbf{e}'\mathbf{r}$ losses information on $\mathbf{r}$, and $\mathbf{A}, \mathbf{k}$ are now chosen at random one can appeal to LHL to show that such a distinguisher should distinguish between $(\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3)$ generated as $(\mathbf{a}, u, \langle \mathbf{x}_0, \mathbf{a}\rangle + \mathbf{e}'\mathbf{r})$ from random, where $\mathbf{a} \in \mathbb{Z}_q^{n\times 1}$ and $u \in \mathbb{Z}_q$ are random.

Together $(\mathsf{ct}_1, \mathsf{ct}_3)$ are now almost distributed as an LWE sample in $\mathbf{x}_0$. If we have a quantum search to decision reduction for LWE using the quantum distinguisher (with an internal state), we should be able to extract $\mathbf{x}_0$ efficiently giving rise to a contradiction. That is, this reduction would have produced a valid $(\alpha, \mathbf{d})$ with overwhelming probability and conditioned on that $\mathbf{x}_0$ with inverse polynomial probability (something ruled out by the NTCF in [BCM+21]).

In this work, we construct a post-quantum (quantum) reduction that runs in time polynomial in $(B, n, \frac{1}{\epsilon}, \log q, T_{\mathcal{A}})$ where ϵ is the distinguishing advantage of an adversary that distinguishes samples LWE in a fixed secret from random samples and recovers that secret with probability polynomial in ϵ[3]. Above B is the bound on the coordinates of $\mathbf{x}_0$ which we will set to be slightly superpolynomial and $T_{\mathcal{A}}$ is the running time of the adversary.

There is a minor flaw in the above argument that can be fixed. The distribution $(\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3)$ formed as $(\mathbf{Ar}, \mathbf{kr}, \mathbf{yr})$ does not behave statistically close to $(\mathbf{a}, u, \langle \mathbf{a}, \mathbf{x}_0\rangle + e)$ for truly random $\mathbf{a}, u$ and an error e sampled according to LWE distribution. The reason for that is that if we consider $\mathbf{yr} = \mathbf{x}_0\mathbf{Ar} + \mathbf{e}'\mathbf{r}$, the error $e = \mathbf{e}'\mathbf{r}$ might not behave as an statistically independent discrete Gaussian. To fix this issue, we modify our encryption algorithm to have a smudging noise $\mathbf{e}''$ with superpolynomially larger parameter $\sigma'' \gg \sigma$ and construct $(\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3)$ as $(\mathbf{Ar}, \mathbf{kr}, \mathbf{yr} + \mathbf{e}'' + \mu\lceil\frac{q}{2}\rceil)$. With this smudging $\mathbf{e}'\mathbf{r}$ can now be drowned by $\mathbf{e}''$ effectively now behaving as a fresh LWE sample in $\mathbf{x}_0$ with slightly larger noise. Accordingly, we design our search-to-decision reduction to also work for samples statistically-close to LWE versus samples close to random, as opposed to the simple case of real LWE versus truly random.

[3]While we can also obtain such a reduction directly from plugging in Theorem 7.1 in [BBK22], as we will discuss later, their reduction's success probability is not sufficient in our proof of the standard security.

Stronger Security from Parallel Repetition While the scheme above gives rise to a PKE scheme satisfying a weak but a non-trivial security guarantee, such a definition is not really enough. It might be possible that an adversary simply guesses the correct certificate $(\alpha, \mathbf{d})$ by picking $\mathbf{d}$ randomly and choosing α at random as well. With $\frac{1}{2}$ probability, this adversary might be right in producing a valid certificate. Since its state is still preserved, it could continue to successfully decrypt all ciphertexts (moreover, it could simply measure the state in the standard basis and keep a classical key). We'd like to achieve a stronger definition where even if the certificate passes with a noticable probability, upon successful passing, the advantage should not be non-negligible.

Our main idea there is that one could create λ independent instances of the scheme where one encrypts secret sharing of the bit μ where λ is the security parameter. The idea is that now the public key would consist of λ indepdent matrices $\mathbf{A}_i$, λ indepdent binary secret LWE samples $\mathbf{k}_i = \mathbf{s}_i\mathbf{A}_i + \mathbf{e}_i$ along with $\mathbf{y}_i \approx \mathbf{x}_{i,0}\mathbf{A}_i + \mathbf{e}'_i$ for $i \in [\lambda]$. The deletion certificate would consist of λ vectors $(\alpha_i, \mathbf{d}_i)$ and each of them are indepdently verified. The hope again is that if one is able to distinguish encryptions of 0 from encryptions of 1, then such an adversary should essentially be able to distinguish for each component $i \in [\lambda]$. We could use this to exctract all $\{\mathbf{x}_{i,b_i}\}_{i\in[\lambda]}(b_i = 0$ or 1). Then, we can appeal to the results from the parallel repetition variant of the game used for proofs of quantumness. This variant has been studied in [RS19]. The soundness property ensures that it is computationally hard to come up with noticable probability valid responses $\{(\alpha_i, \mathbf{d}_i)\}_{i\in[\lambda]}$ and $\{\mathbf{x}_{i,b_i}\}_{i\in[\lambda]}(b_i = 0$ or 1) for all indices.

Lifting the Scheme to Support Homomorphism The above idea could work however, we would have to examine a lot of care to extract the secrets $\{\mathbf{x}_{b_i,i}\}_{i\in[\lambda]}$ as a quantum state can evolve over time. At this point, we move to directly construct a (levelled) FHE scheme. The structural properties of our levelled FHE scheme would solve two problems in one shot. It will not only yield as an FHE scheme, but will also simplify our analysis.

To lift to FHE, we take inspiration for the GSW scheme [GSW13]. In the GSW scheme, the public key consists of a pseudorandom LWE matrix $\mathbf{B} \in \mathbb{Z}_q^{N\times M}$ where $M = \Omega(N \log q)$ such that there exists one secret vector $\mathbf{s} \in \mathbb{Z}_q^{1\times N}$ so that $\mathbf{sB}$ is small norm. Such matrices can be generated by sampling first $N - 1$ rows at random and the last row generated as an LWE sample with the coefficient being the first $N-1$ rows. The encryption of a bit μ is of the form $\mathbf{BR} + \mu\mathbf{G}$ where $\mathbf{R}$ is a small norm random binary matrix and $\mathbf{G}$ is a special gadget matrix [MP12]. The consequence of this is that $\mathbf{BR}$ behaves essentially like a random LWE sample with the same secret vector as $\mathbf{B}$ and one could argue security by appealing to LHL. We omit a description of why the ciphertext could be homomorphically computed on (one could refer to either [GSW13] or our technical sections).

To lift to such an FHE scheme, we work with a specially chosen $\mathbf{B}$. We choose it as:

$$\mathbf{B} = \begin{bmatrix} \mathbf{k}_1 = \mathbf{s}_1\mathbf{A}_1 + \mathbf{e}_1 \\ \mathbf{A}_1 \\ \cdots \\ \cdots \\ \mathbf{k}_\lambda = \mathbf{s}_\lambda\mathbf{A}_\lambda + \mathbf{e}_\lambda \\ \mathbf{A}_\lambda \\ \sum_{i\in[\lambda]} \mathbf{y}_i = \sum_{i\in[\lambda]} \mathbf{x}_{i,0}\mathbf{A}_i + \mathbf{e}'_i \end{bmatrix}$$

Observe that there are many vectors $\mathbf{v}$ so that $\mathbf{vB}$ is small norm. This is crucial because we would like the ciphertexts to be decryptable using a secret key $\rho_{\mathsf{sk}} = \otimes_i^\lambda \rho_{\mathsf{sk},i}$ where $\rho_{\mathsf{sk},i} =$

$|(0, \mathbf{x}_{i,0})\rangle + |(1, \mathbf{x}_{1,i})\rangle$. The idea is that $\mathbf{x}_{0,i}\mathbf{A}_i$ is close to $\mathbf{y}_i$ and similarly $\mathbf{k}_i - \mathbf{x}_{1,i}\mathbf{A}_i$ is also close to $\mathbf{y}_i$. Thus, $\mathbf{B}$ can be decrypted by viewing ρ_{sk} as a vector we can perform a GSW-like decryption in superposition, with a gentle measurement on the final output.

For security, we consider the structure of the ciphertext $\mathsf{ct} = \mathbf{BR} + \mu\mathbf{G}$, the term $\mathbf{BR}$ is of the form:

$$\mathbf{BR} = \begin{bmatrix} \mathbf{s}_1\mathbf{A}_1\mathbf{R} + \mathbf{e}_1\mathbf{R} \\ \mathbf{A}_1\mathbf{R} \\ \cdots \\ \cdots \\ \mathbf{s}_\lambda\mathbf{A}_\lambda\mathbf{R} + \mathbf{e}_\lambda\mathbf{R} \\ \mathbf{A}_\lambda\mathbf{R} \\ \sum_{i\in[\lambda]} \mathbf{y}_i\mathbf{R} = \sum_{i\in[\lambda]} \mathbf{x}_{i,0}\mathbf{A}_i\mathbf{R} + \mathbf{e}'_i\mathbf{R} \end{bmatrix}$$

Our intuition to extract $\mathbf{x}_{i,0}$(w.l.o.g. or $\mathbf{x}_{i,1}$) for all $i \in [\lambda]$ is as follows. First we observe that it will suffice to devise an extractor that will succeed with high probability in the case when $\mathbf{k}_i$ is chosen to be random as opposed to be pseudorandom due to LWE. If we are able to do that, we should be able to extract $\mathbf{x}_{i,b_i}$ with similar probability in the world where the $\mathbf{k}_i$'s are pseudorandom due to LWE security.

To realize our relaxed goal, we observe that the last row $\sum_{i\in[\lambda]} \mathbf{y}_i\mathbf{R}$ is close to a linear equation of the form:

$$\sum_{i\in[\lambda]} \mathbf{y}_i\mathbf{R} \approx \sum_{i\in[\lambda]} \mathbf{x}_{i,0}\mathbf{V}_i$$

where $\mathbf{V}_i = \mathbf{A}_i\mathbf{R}$. Thus, we modify our encryption algorithm to have smudging noise.

After we modify the encryption algorithm to compute $\mathbf{BR} + \mathbf{E} + \mu\mathbf{G}$ where $\mathbf{E}$ is zero everywhere else except the last row containing discrete Gaussian with parameter σ'' superpolnomially more than the norm of $\mathbf{e}'_i\mathbf{R}$, we make a nice observation: the last row of $\mathbf{BR} + \mathbf{E}$ would now be distributed as an LWE sample: in terms of the secret $(\mathbf{x}_{1,b_1}, \mathbf{x}_{2,b_2}, \ldots, \mathbf{x}_{\lambda,b_\lambda})(b_i = 0 \text{ or } 1)$ with the coefficient vector $[\mathbf{V}_1^\top, \ldots, \mathbf{V}_\lambda^\top]^\top$. Now we could appeal the LHL to replace $\{(\mathbf{A}_i\mathbf{R}, \mathbf{V}_i\mathbf{R}, \mathbf{k}_i\mathbf{R})\}_{i\in[\lambda]}$ to completely random and the last row by a fresh sample with error according to discrete Gaussian in parameter σ'', with the long secret $((\mathbf{x}_{1,b_1}, \mathbf{x}_{2,b_2}, \ldots, \mathbf{x}_{\lambda,b_\lambda})$ and random coefficient vector $[\mathbf{V}_1^\top, \ldots, \mathbf{V}_\lambda^\top]^\top$. If an adversary now distinguish such ciphertexts from random, we should be able to extract the entire long secret vector in one shot $(\mathbf{x}_{1,b_1}, \mathbf{x}_{2,b_2}, \ldots, \mathbf{x}_{\lambda,b_\lambda})$ using our proposed quantum search-to-decision reduction.

Completely Classical Communication and Classical Lessor Our protocol comes with classical lessor and classical communication for free. We observe that from the property of the underlying NTCF-based protocol, the lessor only has to run Setup and sends the classical $\mathsf{mpk} = \{\mathbf{A}, \mathbf{sA} + \mathbf{e}\}$(naturally extended to parallel-repeated case) to the lessee. The lessee can prepare its own quantum key given mpk by preparing a superposition of $\sum_{b,\mathbf{x},\mathbf{e}'} |b, \mathbf{x}_b\rangle |\mathbf{y} = \mathbf{x}_b\mathbf{A} + \mathbf{e}' + b \cdot \mathbf{sA}\rangle$ by sampling $\mathbf{e}'$ on its own. It then measures the $\mathbf{y}$-register to obtain a public key $\mathbf{y}$ and a quantum decryption key of the form $|0, \mathbf{x}_0\rangle + |1, \mathbf{x}_1\rangle$. Working with the properties of NTCF (shown in [BCM+21]), the security of our scheme will be guaranteed even for maliciously generated $\mathbf{y}$.

2.1 Detailed Overview on Security Proof: Reduction to NTCF and the Use of Search-to-Decision Reduction for LWE

Proof Outline We now go into slight more details about our high level proof strategy. Recall that in the security game, a successful attacker would first need to output a valid deletion certifiacte (which we denote by the event $\mathsf{CertPass}$) along with that it must produce a state ρ_{delete} for which our test of good decryptor $\mathsf{TI}_{\frac{1}{2}+\gamma}$ for some noticable γ passes with invere-polynomial probability. Namely $\mathsf{TI}_{\frac{1}{2}+\gamma}(\rho_{\mathsf{delete}}) = 1$ with inverse polynomial probability. We call this event $\mathsf{GoodDecryptor}$. It must hold that $\Pr[\mathsf{GoodDecryptor} \wedge \mathsf{CertPass}]$ is noticable for such successful attacker. Since we are guaranteed that there is a noticable chance of overlap of the two events, our hope is that our extraction procedure would extract $\{\mathbf{x}_{0,i}\}$ just with enough probability to induce a noticable probability overlap between $\mathsf{CertPass}$ and successful extraction causing our reduction to win the parallel repetion game of Radian et. al. We call the event when the extraction holds as $\mathsf{ExtractionOccurs}$.

At this point it is tempting to think of a search-to-decision classical reduction for LWE and compile it to a quantum-reduction via known menthods such as the one by Bitansky et. al. [BBK22]. Unfortunately, we don't have a citable theorem from this work to use directly[4]. We need precise bounds on the probability of success, and the running time for the extraction.

Let us briefly explain the story: for the extraction to occur we would need to move to a world where $\mathbf{k}_i$'s are switched with random. While this won't change the probability of $\mathsf{GoodDecryptor}$ by a non-negligible amount due to LWE security, in this hybrid deletion certificates will no longer exist. We could infer from there using reductions compiled using [BBK22]) that $\mathsf{ExtractionOccurs}$ with probability $\frac{1}{\mathsf{poly}(\lambda)}$ for some polynomial poly. This probability of extraction will still be the same (up to a negligible loss) if we went back to the world where $\{\mathbf{k}_i\}$ are again from the LWE distribution. However, we can't infer from this that the event of $\mathsf{ExtractionOccurs}$ overlaps with $\mathsf{CertPass}$. A similar issue was encountered by Ananth et. al. [APV23]. To address this issue the [APV23] had to rely on new conjectures.

To address this issue, we devise a high probability search-to-decision reduction that extracts (in the world where $\mathbf{k}_i$'s are random) with probability $1 - \mathsf{negl}(\lambda)$ whenever $\mathsf{GoodDecryptor}$ holds. Thus when we switch $\mathbf{k}_i$'s with LWE, due to LWE security, $\mathsf{ExtractionOccurs}$ also succeeds with all but negligible security whenever $\mathsf{GoodDecryptor}$ holds. Since we are guaranteed that there is a noticeable overlap between $\mathsf{GoodDecryptor}$ and $\mathsf{CertPass}$, this implies a noticeable overlap between $\mathsf{CertPass}$ and $\mathsf{ExtractionOccurs}$.

Threshold Implementation Before going into more technical details, we briefly describe the properties of the measurement we perform on the adversarial decryptor state to test if it succeeds on distinguishing ciphertexts with high enough probability. We will leverage the properties of this measurement in our security proofs.

Threshold Implementation (and its related efficient measurement implementations in Section 5) is a powerful technique by Zhandry (which is further inspired by Mariott-Watrous's work on QMA amplification [MW05]) used in a number of recent works [CLLZ21, ALL+20, CMSZ22, LLQZ22].

[4] As discussed in Section 1.2, in fact all existing works on post-quantum LWE search-to-decision reductions are not directly applicable to our setting.

The Threshold Implementation $\mathsf{TI}_{\gamma+1/2}$ has the following properties and relations to our security:

1. We will call ρ a good decryptor if we $\mathsf{TI}_{\gamma+1/2}$ applied on ρ outputs 1.
2. For a successful adversary in our game, it must produce a good decryptor ρ with probability p for some noticeable p (apart from giving a valid certificate).
3. For the remaining state ρ' after performing the above $\mathsf{TI}_{\gamma+1/2}$, given the outcome being 1, applying the same $\mathsf{TI}_{\gamma+1/2}$ on ρ' will yield outcome 1 with probability 1.

The above statement basically says that the measurement $\mathsf{TI}_{\gamma+1/2}$ is projective and it "preserves" the advantage of the quantum distinguisher's state.

Ideas from Classical Search-to-Decision Reduction Our quantum search-to-decision reduction the quanis inspired by Regev's search to decision reduction [Reg05b]. We now describe the setup of our reduction. We consider a simpler setup than in our technical section (which is a bit specific to our setting).

The adversary gets as input an LWE sample of the form $(\mathbf{A}, \mathbf{x}\mathbf{A} + \mathbf{e} \mod p)$ where $\mathbf{x}$ is arbitrary secret with bounded norm B, $\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times m}$ for large enough $m = \Omega(n \log p)$, $\mathbf{e}$ is Gaussian distributed with parameter σ. The reduction has a quantum state ρ that has an inverse polynomial weight on vectors that enables distinguishing samples $(\mathbf{A}', \mathbf{x}\mathbf{A}' + \mathbf{e}' \mod p)$ for randomly chosen $\mathbf{A}' \in \mathbb{Z}_p^{n\times m}$ and error $\mathbf{e}'$ sampled according to discrete Gaussian with parameter σ' superpolynomially larger than σ from truly random with probability $\frac{1}{2} + \gamma$.

We first describe the classical intuition and then describe how to lift that intution to the quantum setting. Classically, we can consider recovering x coordinate by coordinate. Say the first coordinate $x_1 \in [-B, B]$, we could choose a total of $2B$ guesses. For each guess g, we could consider the process of generating samples as:

- Sample a matrix $\mathbf{C} \in \mathbb{Z}_p^{n\times m}$ so that it is random subject to the bottom $n-1$ rows are 0.
- Sample $\mathbf{R}$ to be a random binary matrix $\{0,1\}^{m\times m}$.
- Set $\mathbf{A}' = \mathbf{A}\mathbf{R} + \mathbf{C}$ and $\mathbf{b}' = (\mathbf{x}\mathbf{A} + \mathbf{e})\mathbf{R} + \mathbf{v} + \mathbf{e}'$ where $\mathbf{v}$ is the guess g times the first (and the only non-zero) row of $\mathbf{C}$. Here $\mathbf{e}'$ is generated from the discrete Gaussian vector with parameter σ'.

If our guess was correct, the sample that we end up producing $(\mathbf{A}', \mathbf{b}')$ is distributed statistically close to the distribution for the distinguishing problem (LWE with secret $\mathbf{x}$). This is because $\mathbf{e}' + \mathbf{e}\mathbf{R}$ due to noise flodding is within $\mathsf{poly}(m) \cdot \frac{\sigma}{\sigma'}$ statistical distance. Similarly, $\mathbf{A}'$ due to LHL is within q^{-n} statistical distance from uniform if m is sufficiently large. This means that the distribution is within $\eta = \mathsf{poly}(m)\frac{\sigma}{\sigma'}$ statistical distance from the relevant distinguishing problem.

On the other hand, if the guess is incorrect, then one can observe that the distribution produces samples that are within similar statistical distance $\eta = \mathsf{poly}(m) \cdot \frac{\sigma}{\sigma'}$ from truly random distribution.

Thus, if the guess is correct, a classical adversary can distinguish the above distribution from random with probability at least $\frac{1}{2} + \gamma - O(\eta)$, and likewise if the guess is incorrect the maximum distinguishing probability is $\frac{1}{2} + O(\eta)$. We could therefore test the adversary by making roughly $\mathsf{poly}(\frac{1}{\gamma}, \log \delta)$ calls to the distinguisher to guarantee the guess is correct with probability $1 - \delta$. Setting $\delta = p^{-n}$, the reduction will extract $\mathbf{x}$ in time that's polynomial in $B, n, m, \log p, \frac{1}{\gamma}$ (bound on $\mathbf{x}$'s norm).

Our Quantum Search to Decision Reduction Moving to the quantum setting, we have to address a number of challenges. If we use our state to distinguish and LWE sample from random in the way above, the state could get destructed preventing us from doing repetitions. In particular, the classical reduction needs to run the distinguisher many times over randomized inputs and "measure" its outcome to obtain useful information, which seems implausible when using a quantum distinguisher.

To overcome these issue, we will leverage the properties of Threshold Implementation. We make a key observation that the procedure where we repeatedly create samples according to our guess g and check on the distinguisher's output distribution to get an estimate on whether g is correct, can happen "inside" the measurement procedure TI.

Suppose we have efficient projective measurements $\mathsf{TI}_{g,1/2+\gamma} = (\mathsf{TI}_{g,1/2+\gamma}, \mathbf{I} - \mathsf{TI}_{g,1/2+\gamma})$ for various guesses g. $\mathsf{TI}_{g,1/2+\gamma}$ projects the state ρ onto vectors that distinguish the above distribution made from the guess g above from truly random with probability $1/2 + \gamma$. For simplicity, we call it TI_g. from now on.

We define two other projections $\mathsf{TI}_{\mathsf{LWE}}$ and $\mathsf{TI}_{\mathsf{unif}}$:

- $\mathsf{TI}_{\mathsf{LWE}}$ projects onto distinguishers good at distinguishing the ciphertexts in our scheme from uniform random samples with probability at least $1/2+\gamma$. (Intuitively, these are distinguishers for "noisy" LWE instances versus uniform random)
- $\mathsf{TI}_{\mathsf{unif}}$ projects onto distinguishers good at distinguishing uniform radnom samples from uniform random samples with probability at least $1/2 + \gamma$.

Our goal is to show that $\Pr[\mathsf{ExtractionOccurs}] \geq \Pr[\mathsf{TI}_{\mathsf{LWE}}(\rho) = 1] - \mathsf{negl}(\lambda)$. Thus it suffices to consider a world where $\mathsf{TI}_{\mathsf{LWE}}(\rho) = 1$ already happens and show that $\Pr[\mathsf{ExtractionOccurs}] \geq 1 - \mathsf{negl}(\lambda)$ in this world. ρ' is the post measurement state.

We now make the following observations, by recalling the properties of TI:

- Let ρ' be the state we get post measurement of $\mathsf{TI}_{g_i,1/2+\gamma}(\rho) \to 1$.
- We start working with the state ρ' at the beginning of our extraction algorithm.
- Recall that by the projective property of TI, given that outcome is 1, we have $\mathsf{Pr}[\mathsf{TI}_{\mathsf{LWE}},\ (\rho')] = 1$.
- We start with the first entry of $\mathbf{x}$ and pick the smallest possible value as our guess g for this entry.
- As we have discussed in the classical setting, when the guess g is correct, we get to create samples statistically close to LWE samples; when the guess g is incorrect, we get to create samples close to uniform random. Combining these with properties of TI_g for distributionally close measurements, we can show that:
  - **When the guess is correct:** If we apply projection TI_g on it for a correct guess g, we will have $\Pr[\mathsf{TI}_g(\rho') = 1]$ overwhelmingly close to 1. In this setting, we are statistically close to measuring the original $\mathsf{TI}_{\mathsf{LWE}}$. That is, if a distinguisher can distinguish between (noisy) LWE versus real random. We will therefore get output 1 with overwhelming probability as a consequence of TI being a projection.
    We can then assign g to the entry of $\mathbf{x}$ we are guessing for, and move on to the next entry.
  - **When the guess is incorrect:** If we apply projection TI_g on it for an incorrect guess g, we will have $\Pr[(\mathbf{I} - \mathsf{TI}_g)(\rho') = 1] = \Pr[\mathsf{TI}_g(\rho') = 0]$ overwhelmingly close to 1. In this case,

we are statistically close to measuring $\mathsf{TI}_{\mathsf{unif}}$, i.e. asking the distinguisher to distinguish between uniform vs. uniform. Clearly, no distinguisher can distinguish with noticeable advantage γ. Therefore $\mathsf{TI}_{\mathsf{unif}}$ will output 0 for all possible states.
We then move on to perform TI_g with the next possible value of g.

A key observation is that, in both cases above, because we obtain a certain measurement outcome with overwhelming probability, we can apply the gentle measurement lemma [Aar04], we can recover back to a state very close to ρ'. Thus, we could find out $\mathbf{x}$ entry by entry while keeping our quantum state almost intact.

However, TI is not an efficient measurement and we need to make the above procedure efficient. Fortunately, the work of Zhandry [Zha20], showed how to replace TI by an efficient, approximate projection. These approximate projections induce an error of their own, but are very easy to account for. Moreover, the fact that TI implicitly performs amplification that can be analyzed by concentration bounds analogous to the repetitions used in the classical reduction, is even easier to observe when we look into its efficient implementation algorithm (see Section 5.0.1).

In the end, we further materialize our approach with some proper choice of parameters so that the errors will not accumulate on distinguisher's state to a degree that affects the algorithm's performance. Conclusively, the exact reduction in our scheme works if the number of guesses are superpolynomially lesser than $\frac{1}{\eta}$. See details in Section 11. On the other hand, if we consider a clean setting where our input distributions are real LWE versus truly uniform, our reduction works for any subexponential number of guesses (Appendix E).

2.2 More on Related Works

We addtionally discuss some other related works.

Quantum Copy Protection [Aar09] first built copy-protection for all unlearnable functions based on a quantum oracle. [ALL$^+$20] showed a construction for all unlearnable functions based on a classical oracle. [CLLZ21, LLQZ22] showed copy protection for signatures, decryption and PRF evaluation in the plain model. [CMP20, AKL$^+$22] constructed copy-protection for point functions and compute-and-compare functions in QROM, the latter improving the security of the former.

Regarding the negative results: [AP21] demonstrated that it is impossible to have a copy-protection scheme for all unlearnable circuits in the plain model, assuming LWE and quantum FHE.

More on Secure Software Leasing [ALL$^+$20] observed that under a definition essentially equivalent to infinite-term SSL, namely copy-detection, one could obtain a black-box construction for infinite-term SSL from watermarking and public-key quantum money. [KNY21] constructed finite-term SSL for PRFs and compute-and-compare functions from (subexponential) LWE, with similar observations.

[BJL$^+$21, CMP20] constructed secure software leasing for point functions and compute-and-compare functions; [BJL$^+$21] is information-theoretically secure and [CMP20] is secure under QROM. They both used a stronger version of finite-term SSL security: while the vendor will honestly check the returned state from the adversary, the adversary can execute the leftover half of its bipartite state maliciously, i.e., not following the instructions in the evaluation algorithm. SSL security of this stronger finite-term variant is only known for point/compute-and-compare functions up till now.

Unclonable Encryption and Certified Deletion Unclonable encryption is studied in [GC01, BL19, AK21, AKL$^+$22]. It is an encryption scheme where the ciphertext is encoded in quantum information, so that the adversary cannot split it into two copies that can both decrypt when given a (classical) decryption key.

Certified deletion of ciphertext is studied in various works [BI20, Por22, HMNY21, BK23, BGG$^+$23], where the ciphertext is also encoded in a quantum state and given to a server. When asked by the client to delete, the server must provide a proof of deletion of the ciphertext. This bears some similarity to our setting. But note the major difference is that in the certified deletion of ciphertext setting, we need to show that the server no longer has information about the encrypted message(a bit string), while in our setting we need to show the server is deprived of a functionality. Therefore, the results and techniques in secure key leasing and certified deletion of ciphertext are most of the time incomparable.

3 Acknowledgement

Orestis Chardouvelis and Aayush Jain were supported by gifts from CyLab of CMU and Google. Jiahui Liu was supported by DARPA under Agreement No. HR00112020023, NSF CNS-2154149 and a Simons Investigator award.

We thank Qipeng Liu, Mark Zhandry and the authors of [APV23] for helpful discussions.

4 Preliminaries

4.1 Quantum Information an Computation

We refer the reader to [NC02] for a reference of basic quantum information and computation concepts.

A quantum system Q is defined over a finite set B of classical states. In this work we will consider $B = \{0,1\}^n$. A **pure state** over Q is a unit vector in $\mathbb{C}^{|B|}$, which assigns a complex number to each element in B. In other words, let $|\phi\rangle$ be a pure state in Q, we can write $|\phi\rangle$ as:

$$|\phi\rangle = \sum_{x \in B} \alpha_x |x\rangle$$

where $\sum_{x \in B} |\alpha_x|^2 = 1$ and $\{|x\rangle\}_{x \in B}$ is called the "**computational basis**" of $\mathbb{C}^{|B|}$. The computational basis forms an orthonormal basis of $\mathbb{C}^{|B|}$.

Given two quantum systems R_1 over B_1 and R_2 over B_2, we can define a **product** quantum system $R_1 \otimes R_2$ over the set $B_1 \times B_2$. Given $|\phi_1\rangle \in R_1$ and $|\phi_2\rangle \in R_2$, we can define the product state $|\phi_1\rangle \otimes |\phi_2\rangle \in R_1 \otimes R_2$.

We assume a quantum computer can implement any unitary transformation (by using these basic gates, Hadamard, phase, CNOT and $\frac{\pi}{8}$ gates), especially the following two unitary transformations:

- **Classical Computation:** Given a function $f : X \to Y$, one can implement a unitary U_f over $\mathbb{C}^{|X|\cdot|Y|} \to \mathbb{C}^{|X|\cdot|Y|}$ such that for any $|\phi\rangle = \sum_{x\in X,y\in Y} \alpha_{x,y}|x,y\rangle$,

$$U_f|\phi\rangle = \sum_{x\in X,y\in Y} \alpha_{x,y}|x, y \oplus f(x)\rangle$$

  Here, $\oplus$ is a commutative group operation defined over Y.
- **Quantum Fourier Transform:** Let $N = 2^n$. Given a quantum state $|\phi\rangle = \sum_{i=0}^{2^n-1} x_i|i\rangle$, by applying only $O(n^2)$ basic gates, one can compute $|\psi\rangle = \sum_{i=0}^{2^n-1} y_i|i\rangle$ where the sequence $\{y_i\}_{i=0}^{2^n-1}$ is the sequence achieved by applying the classical Fourier transform QFT_N to the sequence $\{x_i\}_{i=0}^{2^n-1}$:

$$y_k = \frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} x_i \omega_n^{ik}$$

  where $\omega_n = e^{2\pi i/N}$, i is the imaginary unit.
  One property of QFT is that by preparing $|0^n\rangle$ and applying QFT_2 to each qubit, $(\mathsf{QFT}_2|0\rangle)^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$ which is a uniform superposition over all possible $x \in \{0, 1\}^n$.

Lemma 4.1 (Almost As Good As New(Gentle Measurement) Lemma [Aar04])**.** *Suppose a binary-outcome POVM measurement $(\mathcal{P}, \mathcal{Q})$ on a mixed state ρ yields a particular outcome with probability $1-\epsilon$. Then after the measurement, one can recover a state $\tilde{\rho}$ such that $\|\tilde{\rho} - \rho\|_{\mathrm{tr}} \leq \sqrt{\epsilon}$. Here $\|\cdot\|_{\mathrm{tr}}$ is the trace distance.*

5 Preliminaries: Testing Quantum Adversaries

In this section, we include several definitions about measurements, which are relevant to testing whether quantum adversaries are successful in the security games of Section 8. Part of this section is taken verbatim from [ALL+20]. As this section only pertains directly to our security definitions for secure key leasing schemes, the reader can skip ahead, and return to this section when reading Section 8.

In classical cryptographic security games, the challenger typically gets some information from the adversary and checks if this information satisfies certain properties. However, in a setting where the adversary is required to return *quantum* information to the challenger, classical definitions of "testing" whether a quantum state returned by the adversary satisfies certain properties may result in various failures as discussed in [Zha20], as this state may be in a superposition of "successful" and "unsuccessful" adversaries, most likely unclonable and destructible by the classical way of "testing" its success.

In short, we need a way to test the success probability of an adversarial quantum state analogous to what happens classically, where the test does not completely destruct the adversary's state. Instead, the state after the test has its success probability preserved in some sense. Such a procedure allows us to do multiple measurements on the state without rendering it useless, and thus facilitates quantum reductions.

Projective Implementation Motivated by the discussion above, [Zha20] (inspired by [MW05]) formalizes a new measurement procedure for testing a state received by an adversary. We will be adopting this procedure when defining security of secure key leasing schemes in Section Section 8. Consider the following procedure as a binary POVM $\mathcal{P}$ acting on an alleged-copy-protected program ρ: sample a uniformly random input x, evaluates the copy-protected program on x, and checks if the output is correct. In a nutshell, the new procedure consists of applying an appropriate projective measurement which *measures* the success probability of the tested state ρ under $\mathcal{P}$, and to output "accept" if the success probability is high enough. Of course, such measurement will not be able extract the exact success probability of ρ, as this is impossible from we have argued in the discussion above. Rather, the measurement will output a success probability from a finite set, such that the expected value of the output matches the true success probability of ρ. We will now describe this procedure in more detail.

The starting point is that a POVM specifies exactly the probability distribution over outcomes $\{0,1\}$ ("success" or "failure") on any copy-protected program, but it does not uniquely determine the post-measurement state. Zhandry shows that, for any binary POVM $\mathcal{P} = (P, I - P)$, there exists a particularly nice implementation of $\mathcal{P}$ which is projective, and such that the post-measurement state is an eigenvector of P. In particular, Zhandry observes that there exists a projective measurement $\mathcal{E}$ which *measures* the success probability of a state with respect to $\mathcal{P}$. More precisely,

- $\mathcal{E}$ outputs a *distribution* D of the form $(p, 1-p)$ from a finite set of distribution over outcomes $\{0,1\}$. (we stress that $\mathcal{E}$ actually outputs a distribution).
- The post-measurement state upon obtaining outcome $(p, 1-p)$ is an *eigenvector* (or a mixture of eigenvectors) of P with eigenvalue p.

A measurement $\mathcal{E}$ which satisfies these properties is the measurement in the common eigenbasis of P and $I - P$ (such common eigenbasis exists since P and $I - P$ commute).

Note that since $\mathcal{E}$ is projective, we are guaranteed that applying the same measurement twice will yield the same outcome. Thus, what we obtain from applying $\mathcal{E}$ is a state with a "well-defined" success probability with respect to $\mathcal{P}$: we know exactly how good the leftover program is with respect to the initial testing procedure $\mathcal{P}$.

Formally, to complete the implementation of $\mathcal{P}$, after having applied $\mathcal{E}$, one outputs the bit 1 with probability p, and the bit 0 with probability $1 - p$. This is summarized in the following definition.

Definition 5.1 (Projective Implementation of a POVM)**.** *Let $\mathcal{P} = (P, Q)$ be a binary outcome POVM. Let $\mathcal{D}$ be a finite set of distributions $(p, 1-p)$ over outcomes $\{0,1\}$. Let $\mathcal{E} = \{E_p\}_{(p,1-p)\in\mathcal{D}}$ be a projective measurement with index set $\mathcal{D}$. Consider the following measurement procedure:*

(i) Apply the projective measurement $\mathcal{E}$ and obtain as outcome a distribution $(p, 1-p)$ over $\{0,1\}$;
(ii) Output a bit according to this distribution, i.e. output 1 w.p p and output 0 w.p $1-p$.

We say the above measurement procedure is a projective implementation of $\mathcal{P}$, which we denote by $\mathsf{ProjImp}(\mathcal{P})$*, if it is equivalent to $\mathcal{P}$ (i.e. it produces the same probability distribution over outcomes).*

Zhandry shows that any binary POVM has a projective implementation, as in the previous definition.

Lemma 5.2 (Adapted from Lemma 1 in [Zha20])**.** *Any binary outcome POVM $\mathcal{P} = (P, Q)$ has a projective implementation* $\mathsf{ProjImp}(\mathcal{P})$*.*

Moreover, if the outcome is a distribution $(p, 1-p)$ when measuring under $\mathcal{E}$, the collapsed state ρ' is a mixture of eigenvectors of P with eigenvalue p, and it is also a mixture of eigenvectors of Q with eigenvalue $1-p$.

As anticipated, the procedure that we will eventually use to test a state received from the adversary will be to:

(i) *Measure* the success probability of the state,
(ii) Accept if the outcome is large enough.

As you may guess at this point, we will employ the projective measurement $\mathcal{E}$ defined previously for step (i). We call this variant of the projective implementation a *threshold implementation*.

Threshold Implementation The concept of threshold implementation of a POVM was proposed by Zhandry, and formalized by Aaronson, Liu, Liu, Zhandry and Zhang [ALL+20]. The following is a formal definition.

Definition 5.3 (Threshold Implementation)**.** *Let $\mathcal{P} = (P, Q)$ be a binary POVM. Let $\mathsf{ProjImp}(\mathcal{P})$ be a projective implementation of $\mathcal{P}$, and let $\mathcal{E}$ be the projective measurement in the first step of $\mathsf{ProjImp}(\mathcal{P})$ (using the same notation as in Definition 5.1). Let $\gamma > 0$. We refer to the following measurement procedure as a* threshold implementation *of $\mathcal{P}$ with parameter γ, and we denote is as $\mathsf{TI}_\gamma(\mathcal{P})$.*

- *Apply the projective measurement $\mathcal{E}$, and obtain as outcome a vector $(p, 1-p)$;*
- *Output a bit according to the distribution $(p, 1-p)$: output 1 if $p \geq \gamma$, and 0 otherwise.*

For simplicity, for any quantum state ρ, we denote by $\mathrm{Tr}[\mathsf{TI}_\gamma(\mathcal{P})\,\rho]$ the probability that the threshold implementation applied to ρ **outputs 1**. Thus, whenever $\mathsf{TI}_\gamma(\mathcal{P})$ appears inside a trace Tr, we treat $\mathsf{TI}_\gamma(\mathcal{P})$ as a projection onto the 1 outcome (i.e. the space spanned by eigenvectors of P with eigenvalue at least γ).

Similarly to Lemma 5.2, we have the following lemma.

Lemma 5.4. *Any binary outcome POVM $\mathcal{P} = (P, Q)$ has a threshold implementation $\mathsf{TI}_\gamma(\mathcal{P})$ for any γ.*

In this work, we are interested in threshold implementations of POVMs with a particular structure. These POVMs represent a challenger's test of a quantum state received from an adversary in a security game (like the POVM described earlier for testing whether a program evaluates correctly on a uniformly random input). These POVMs have the following structure:

- Sample a projective measurement from a set of projective measurements $\mathcal{I}$, according to some distribution D over $\mathcal{I}$.
- Apply this projective measurement.

We refer to POVMs of this form as *mixtures of projective measurements*. The following is a formal definition.

Definition 5.5 (Collection and Mixture of Projective Measurements)**.** *Let $\mathcal{R}, \mathcal{I}$ be sets. Let $\{(P_i, Q_i)\}_{i\in I}$ be a collection of binary projective measurements $(P_i, Q_i))$ over the same Hilbert space $\mathcal{H}$ where $\mathcal{P}_i$ corresponds to output 0, and $\mathcal{Q}_i$ corresponds to output 1. We will assume we can efficiently measure the $\mathcal{P}_i$ for superpositions of i, meaning we can efficiently perform the following projective measurement over $\mathcal{I} \otimes \mathcal{H}$:*

$$(\sum_i |i\rangle\langle i| \otimes P_i, \sum_i |i\rangle\langle i| \otimes Q_i) \tag{1}$$

Let $\mathcal{D} : \mathcal{R} \to \mathcal{I}$ be some distribution. The mixture of projective measurements *associated to $\mathcal{R}, \mathcal{I}, \mathcal{D}$ and $\{(P_i, Q_i)\}_{i\in I}$ is the binary POVM $\mathcal{P}_{\mathcal{D}} = (P_{\mathcal{D}}, Q_{\mathcal{D}})$ defined as follows:*

$$P_D = \sum_{i\in\mathcal{I}} \Pr[i \leftarrow \mathcal{D}(R)]\, P_i, \qquad Q_{\mathcal{D}} = \sum_{i\in\mathcal{I}} \Pr[i \leftarrow \mathcal{D}(R)]\, Q_i, \tag{2}$$

In other words, $\mathcal{P}_D$ is implemented in the following way: sample randomness $r \leftarrow \mathcal{R}$, compute the index $i = D(r)$, and apply the projective measurement (P_i, Q_i). Thus, for any quantum state ρ, $\mathrm{Tr}[P_D\rho]$ is the probability that a projective measurement (P_i, Q_i), sampled according to the distribution induced by D, applied to ρ outputs 1.

Example To further explain the above definition, we consider a concrete example: when the input quantum state ρ is supposedly a decryptor for an encryptions with respect to public key pk, and our goal is to measure if ρ can distinguish between encryption of message 0 and message 1.

The distribution $\mathcal{D}$ is the distribution over all randomness used to encrypt a message and the coin flip b to decide which ciphertext to feed to the adversary. For a ciphertext $\mathsf{ct}_i \leftarrow \mathcal{D}$, $\mathcal{P}_{\mathsf{ct}_i}$ is the measurement that runs the adversary ρ on ct_i and tests if the outcome $b' = b$.

Purifying Mixtures of Projection We next define the purified version of the above operations, called Controlled Projections:

Definition 5.6 (Controlled Projective Measurements)**.** *Let $\mathcal{P} = \{\mathcal{P}_i = (P_i, Q_i)\}, i \in \mathcal{I}$ be a collection of projective measurements over $\mathcal{H}$. Let $\mathcal{D}$ a distribution with random coin set $\mathcal{R}$. We will abuse notation and let $\mathcal{R}$ also denote the $|\mathcal{R}|$-dimensional Hilbert space. The controlled projection is the measurement: $\mathsf{CProj}_{\mathcal{P},\mathcal{D}} = (\mathsf{CProj}^0_{\mathcal{P},\mathcal{D}}, \mathsf{CProj}^1_{\mathcal{P},\mathcal{D}})$ where:*

$$\mathsf{CProj}^0_{\mathcal{P},\mathcal{D}} = \sum_{r\in\mathcal{R}} |r\rangle\langle r| \otimes P_{\mathcal{D}(r)}, \mathsf{CProj}^1_{\mathcal{P},\mathcal{D}} = \sum_{r\in\mathcal{R}} |r\rangle\langle r| \otimes Q_{\mathcal{D}(r)}$$

$\mathsf{CProj}_{\mathcal{P},\mathcal{D}}$ can be implemented using the measurement $(\sum_i |i\rangle\langle i| \otimes P_i, \sum_i |i\rangle\langle i| \otimes Q_i)$. First, initialize control registers $\mathcal{I}$ to 0. Then perform the map $|r\rangle|i\rangle \to |r\rangle|i \oplus \mathcal{D}(r)\rangle$ to the $\mathcal{R} \otimes \mathcal{I}$ registers. Next, apply the mixture of projective measurements assumed in Equation (1). Finally, perform the map $|r\rangle|i\rangle \to |r\rangle|i \oplus \mathcal{D}(r)\rangle$ again to uncompute the control registers, and discard the control registers.

5.0.1 Approximating Threshold Implementation

Projective and *threshold* implementations of POVMs are unfortunately not efficiently computable in general.

However, they can be approximated if the POVM is a mixture of projective measurements, as shown by Zhandry [Zha20], using a technique first introduced by Marriott and Watrous [MW05] in the context of error reduction for quantum Arthur-Merlin games.

The Uniform Test Before we describe the ATI algorithm, we define the projection on register $\mathcal{R}$ that tests if it is a superposition with each $r \in \mathcal{R}$ weighted according to the distribution $\mathcal{D}$ [5]: $\mathsf{IsUniform} = (|\mathbf{1}_\mathcal{R}\rangle\langle\mathbf{1}_\mathcal{R}|, \mathcal{I} - |\mathbf{1}_\mathcal{R}\rangle\langle\mathbf{1}_\mathcal{R}|)$ where:

$$|\mathbf{1}_\mathcal{R}\rangle = \frac{1}{\sqrt{|\mathcal{R}|}}\sum_{r\in\mathcal{R}}|r\rangle$$

The Algorithm ATI We present the algorithm $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D},\gamma}$ using the syntax from [Zha20]:

Our algorithm is parameterized by a distribution $\mathcal{D}$, collection of projective measurements $\mathcal{P}$, and real values $0 < \epsilon, \delta, \gamma \leq 1$, and is denoted as $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D},\gamma}$.

On input a quantum state $|\psi\rangle$ over Hilbert space $\mathcal{H}$:

1. Initialize a state $|\mathbf{1}_\mathcal{R}\rangle|\psi\rangle$.
2. Initialize a classical list $L := (1)$.
3. Repeat the following loop a total of $T = \frac{\ln(4/\delta)}{\epsilon^2}$ times:
   (a) Apply $\mathsf{CProj}_{\mathcal{P},\mathcal{D}}$ to register $\mathcal{R} \otimes \mathcal{H}$. Let b_{2i-1} be the measurement outcome and set $L := (L, b_{2i-1})$.
   (b) Apply $\mathsf{IsUniform}$ to register $\mathcal{R}$. Let b_{2i} be the measurement outcome and set $L := (L, b_{2i})$.
4. Let t be the number of index i such that $b_{i-1} = b_i$ in the list $L = (0, b_1, ..., b_{2T})$, and $\tilde{p} := t/2T$.
5. If $b_{2T} = 0$, repeat the loop again until $b_{2i} = 1$.
6. Discard the $\mathcal{R}$ register.
7. Output 1 if $\tilde{p} \geq \gamma$ and 0 otherwise.

ATI Properties We will make use of the following lemma from a subsequent work of Aaronson et al. [ALL+20].

Lemma 5.7 (Corollary 1 in [ALL+20]). *For any $\epsilon, \delta, \gamma \in (0, 1)$, any collection of projective measurements $\mathcal{P} = \{(P_i, Q_i)\}_{i\in\mathcal{I}}$, where $\mathcal{I}$ is some index set, and any distribution D over $\mathcal{I}$, there exists a measurement procedure $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$ that satisfies the following:*

[5] The distribution $\mathcal{D}$ used in CProj can in fact be an arbitrary distribution, instead of uniform. Without loss of generality, we can use a uniform test IsUniform and map each r to an arbitrary distribution $\mathcal{D}(r)$ as specified in Definition 5.6.

1. *$\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$ implements a binary outcome measurement. For simplicity, we denote the probability of the measurement* ***outputting*** $\mathbf{1}$ *on ρ by $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}\,\rho]$.*
2. *For all quantum states ρ, $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma-\epsilon}\,\rho] \geq \mathrm{Tr}[\mathsf{TI}_{\gamma}(\mathcal{P}_D)\,\rho] - \delta$.*
3. *For all quantum states ρ, let ρ' be the post-measurement state after applying $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$ on ρ, and obtaining outcome 1. Then, $\mathrm{Tr}[\mathsf{TI}_{\gamma-2\epsilon}(\mathcal{P}_D)\,\rho'] \geq 1 - 2\delta$.*
4. *The expected running time is $T_{\mathcal{P},D} \cdot \mathsf{poly}(1/\epsilon, 1/(\log\delta))$, where $T_{\mathcal{P},D}$ is the combined running time of sampling according to D, of mapping i to (P_i, Q_i), and of implementing the projective measurement (P_i, Q_i).*

Intuitively the corollary says that if a quantum state ρ has weight p on eigenvectors with eigenvalues at least γ, then the measurement $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$ will produce with probability at least $p-\delta$ a post-measurement state which has weight $1-2\delta$ on eigenvectors with eigenvalues at least $\gamma-2\epsilon$. Moreover, the running time for implementing $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$ is proportional to $\mathsf{poly}(1/\epsilon, 1/(\log\delta))$, which is a polynomial in λ as long as ϵ is any inverse polynomial and δ is any inverse sub-exponential function.

TI and ATI For Computationally/Statistically Indistinguishable Distributions The following theorems will be used in the proof of security for our SKL encryption scheme in Section 10.

Informally, the lemma states the following. Let $\mathcal{P}_{D_0}$ and $\mathcal{P}_{D_1}$ be two mixtures of projective measurements, where D_0 and D_1 are two computationally indistinguishable distributions. Let $\gamma, \gamma' > 0$ be inverse-polynomially close. Then for any (efficiently constructible) state ρ, the probabilities of obtaining outcome 1 upon measuring $\mathsf{TI}_{\gamma}(\mathcal{P}_{D_0})$ and $\mathsf{TI}_{\gamma'}(\mathcal{P}_{D_1})$ respectively are negligibly close.

Theorem 5.8 (Theorem 6.5 in [Zha20])**.** *Let $\gamma > 0$. Let $\mathcal{P}$ be a collection of projective measurements indexed by some set $\mathcal{I}$. Let ρ be an efficiently constructible mixed state, and let D_0, D_1 be two efficiently sampleable and computationally indistinguishable distributions over $\mathcal{I}$. For any inverse polynomial ϵ, there exists a negligible function δ such that*

$$\mathrm{Tr}[\mathsf{TI}_{\gamma-\epsilon}(\mathcal{P}_{D_1})\rho] \geq \mathrm{Tr}[\mathsf{TI}_{\gamma}(\mathcal{P}_{D_0})\rho] - \delta\,,$$

where $\mathcal{P}_{D_i}$ is the mixture of projective measurements associated to $\mathcal{P}$ and D_i.

Corollary 5.9 (Corollary 6.9 in [Zha20])**.** *Let ρ be an efficiently constructible, potentially mixed state, and let $\mathcal{D}_0, \mathcal{D}_1$ be two computationally indistinguishable distributions. Then for any inverse polynomial ϵ and any function δ, there exists a negligible $\mathsf{negl}(\cdot)$ such that:*

$$\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{D}_1,\mathcal{P},\gamma-3\epsilon}(\rho)] \geq \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{D}_0,\mathcal{P},\gamma}(\rho)] - 2\delta - \mathsf{negl}(\lambda)$$

Unfortunately these above two theorems do not possess precise enough error parameters for our use in some settings. We additionally show a theorem with more precise parameters for two statistically close distributions.

Theorem 5.10. *let $\mathcal{D}_0, \mathcal{D}_1$ be two statistically-close distributions over $\mathcal{R}$, with distance η. Then for any inverse polynomial ϵ and any function δ, it holds that:*

$$\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{D}_1,\mathcal{P},\gamma-\epsilon}(\rho)] \geq \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{D}_0,\mathcal{P},\gamma}(\rho)] - O(\mathsf{poly}(1/\epsilon, 1/\log\delta) \cdot (\frac{1}{|\mathcal{R}|} + \eta))$$

Since we work with $|\mathcal{R}|$ with exponential size, we will assume $\frac{1}{|\mathcal{R}|}$ to be suppressed when we use the theoreom.

We refer the proof or the above theorem to Appendix C.2

6 Lattice Preliminaries

In this section, we recall some of the notations and concepts about lattice complexity problems and lattice-based cryptography that will be useful to our main result.

6.1 General definitions

A *lattice* $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^m$, or equivalently the set

$$\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \ : \ x_i \in \mathbb{Z} \right\}$$

of all integer combinations of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$. Such $\mathbf{b}_i$'s form a *basis* of $\mathcal{L}$.

The lattice $\mathcal{L}$ is said to be *full-rank* if $n = m$. We denote by $\lambda_1(\mathcal{L})$ the first minimum of $\mathcal{L}$, defined as the length of a shortest non-zero vector of $\mathcal{L}$.

Discrete Gaussian and Related Distributions For any $s > 0$, define

$$\rho_s(\mathbf{x}) = \exp(-\pi \|\mathbf{x}\|^2 / s^2)$$

for all $\mathbf{x} \in \mathbb{R}^n$. We write ρ for ρ_1. For a discrete set S, we extend ρ to sets by $\rho_s(S) = \sum_{\mathbf{x} \in S} \rho_s(\mathbf{x})$. Given a lattice $\mathcal{L}$, the *discrete Gaussian* $\mathcal{D}_{\mathcal{L},s}$ is the distribution over $\mathcal{L}$ such that the probability of a vector $\mathbf{y} \in \mathcal{L}$ is proportional to $\rho_s(\mathbf{y})$:

$$\Pr_{X \leftarrow \mathcal{D}_{\mathcal{L},s}} [X = \mathbf{y}] = \frac{\rho_s(\mathbf{y})}{\rho_s(\mathcal{L})}.$$

Using standard subgaussian tail-bounds, one can show we can show the following claim.

Claim 6.1. *Let $m \in \mathbb{N}$, $\sigma > 0$, then it holds that:*

$$\Pr_{\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}} [\|\mathbf{e}\| > m\sigma] < \exp(-\tilde{\Omega}(m)).$$

The following is Theorem 4.1 from [GPV07] that shows an efficient algorithm to sample from the discrete Gaussian.

Now we recall the Learning with Errors (LWE) distribution.

Definition 6.2. *Let $\sigma = \sigma(n) \in (0, 1)$. For $\mathbf{s} \in \mathbb{Z}_p^n$, the LWE distribution $A_{\mathbf{s},p,\sigma}$ over $\mathbb{Z}_p^n \times \mathbb{Z}_p$ is sampled by independently choosing $\mathbf{a}$ uniformly at random from $\mathbb{Z}_p^n$, and $e \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}$, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e \mod p)$. The LWE assumption $\mathsf{LWE}_{n,m,\sigma,p}$ states that m samples from $A_{\mathbf{s},p,\sigma}$ for a randomly chosen $\mathbf{s} \leftarrow \mathbb{Z}_p^n$ are indistinguishable from m random vectors $\mathbb{Z}_p^{n+1}$.*

6.2 Trapdoor Sampling for LWE

We will need the following definition of a lattice trapdoor [GPV07, MP13, Vai20]. For $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we define the rank m lattice

$$\mathcal{L}^{\perp}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m \ : \ \mathbf{A}\mathbf{z} = \mathbf{0} \pmod q\} \ .$$

A lattice trapdoor for $\mathbf{A}$ is a set of short linearly independent vectors in $\mathcal{L}^{\perp}(\mathbf{A})$.

Definition 6.3. *A matrix $\mathbf{T} \in \mathbb{Z}^{m\times m}$ is a β-good lattice trapdoor for a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ if*

1. $\mathbf{A}\mathbf{T} = \mathbf{0} \pmod q$.
2. *For each column vector $\mathbf{t}_i$ of $\mathbf{T}$, $\|\mathbf{t}_i\|_\infty \le \beta$.*
3. *$\mathbf{T}$ has rank m over $\mathbb{R}$.*

Theorem 6.4. *[GPV07, MP13] There is an efficient algorithm $\mathsf{GenTrap}$ that, on input $1^n, q, m = \Omega(n\log q)$, outputs a matrix $\mathbf{A}$ distributed statistically close to uniformly on $\mathbb{Z}_q^{n\times m}$, and a $O(m)$-good lattice trapdoor $\mathbf{T}$ for $\mathbf{A}$.*

Moreover, there is an efficient algorithm INVERT that, on input $(\mathbf{A}, \mathbf{T})$ and $\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top$ where $\|\mathbf{e}\| \le q/(C_T\sqrt{n\log q})$ and C_T is a universal constant, returns $\mathbf{s}$ and $\mathbf{e}$ with overwhelming probability over $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{GenTrap}(1^n, 1^m, q)$.

Lemma 6.5. *[Noise Smudging, [DGTK+10] Let $y, \sigma > 0$. Then, the statistical distance between the distribution $\mathcal{D}_{\mathbb{Z},\sigma}$ and $\mathcal{D}_{\mathbb{Z},\sigma+y}$ is at most y/σ.*

Lemma 6.6. *(Leftover Hash Lemma(SIS version))*

The leftover hash lemma says that if $m = \Omega(n\log q)$, then if you sample $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{x} \leftarrow \{0,1\}^m$ and $\mathbf{y} \leftarrow \mathbb{Z}_q^m$, we have:

$$(\mathbf{A}, \mathbf{A}\cdot\mathbf{x}) \approx_{q^{-n}} (\mathbf{A}, \mathbf{y}))$$

6.3 Structural Properties of GSW Homomorphic Encryption

Our scheme will build upon the GSW levelled FHE construction [GSW13]. We recall its structure here, for more details refer to [GSW13]. In GSW scheme, the public keys consists of matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ where $m = \Omega(n\log q)$. This matrix is pseudorandom due to LWE. The secret key is a vector $\mathbf{s} \in \mathbb{Z}_q^{1\times n}$ such that $\mathbf{s}\cdot\mathbf{B} = \mathbf{e}$ for a small norm error vector $\mathbf{e}$. Such matrices can be constructed easily using the LWE assumption.

For any such matrix $\mathbf{B}$, to encrypt a bit $\mu \in \{0,1\}$, the encryption algorithm produces a ciphertext $\mathsf{ct} \in \mathbb{Z}_q^{n\times\lceil n\log q\rceil}$ as follows. Sample a random matrix $\mathbf{R}$ to be a random small norm (where for instance each entry is chosen independently from $\{+1,-1\}$). Then we compute $\mathsf{ct} = \mathbf{B}\mathbf{R} + \mu\mathbf{G}$ where $\mathbf{G}$ is the Gadget matrix [MP13]: $\mathbf{G} = [\mathbf{I}\otimes[2^0, 2^1, \cdots, 2^{\lceil\log q\rceil-1}]\|\mathbf{0}^{m\times(m-n\lceil\log q\rceil)}] \in \mathbb{Z}^{n\times m}$ and $\mathbf{I}$ is the $n\times n$ identity. $\mathbf{G}$ converts a binary representation of a vector back to its original vector representation over the field $\mathbb{Z}_q$; the associated (non-linear) inverse operation $\mathbf{G}^{-1}$ converts vectors in Z_q to their binary representation.

Note that $\mathbf{B}\mathbf{R}$ is once again an LWE matrix in the same secret $\mathbf{s}$ and satisfies $\mathbf{s}\mathbf{B}\mathbf{R}$ is low norm therefore $\mathsf{s}\mathsf{ct} \approx \mathbf{s}\mu\mathbf{G}$ which could be used to learn μ.

One can compute NAND operation as follows. Say we have $\mathsf{ct}_1 = \mathbf{B}\mathbf{R}_1 + \mu_1\mathbf{G}$ and $\mathsf{ct}_2 = \mathbf{B}\mathbf{R}_2 + \mu_2\mathbf{G}$. One can compute the bit-decomposition of ct_2, $\mathbf{G}^{-1}(\mathsf{ct}_2)$ that simply expands out ct_2

component wise by replacing every coordinate of ct_2 by a its binary decomposition vector of size $\lceil \log_2 q \rceil$. Observe that $\mathbf{G}^{-1}(\mathsf{ct}_2)$ is a low norm matrix in $\mathbb{Z}_p^{\lceil n \log q \rceil \times \lceil n \log q \rceil}$ Then, one can compute $\mathsf{ct}_\times = \mathsf{ct}_1 \mathbf{G}^{-1}(\mathsf{ct}_2)$. This yields a ciphertext $\mathsf{ct}_\times = \mathbf{B}(\mathbf{R}_1\mathbf{G}^{-1}(\mathsf{ct}_2) + \mu_1\mathbf{R}_2) + \mu_1\mu_2\mathbf{G}$. Observe that the randomness $(\mathbf{R}_1\mathbf{G}^{-1}(\mathsf{ct}_2) + \mu_1\mathbf{R}_2)$ is small norm and the invariant is still maintained. Finally to compute NAND operation one simply outputs $\mathsf{ct}_{\mathsf{NAND}} = \mathbf{G} - \mathsf{ct}_\times$ yielding a ciphertext encryption $1 - \mu_1\mu_2$.

The security proof follows form the fact that by LHL, $\mathbf{BR}$ is random provided $\mathbf{B}$ is chosen at random. By the security of LWE, $\mathbf{B}$ is pseudorandom therefore $\mathbf{BR}$ is also pseudorandom.

7 Preliminaries: Noisy Claw Free Trapdoor Families

7.1 Noisy Trapdoor Claw-Free Families

The following definition of NTCF families is taken verbatim from [BCM+21, Definition 6]. For a more detailed exposition of the definition, we refer the readers to the prior work.

Definition 7.1 (NTCF family)**.** *Let λ be a security parameter. Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $\mathcal{K}_\mathcal{F}$ be a finite set of keys. A family of functions*

$$\mathcal{F} = \left\{ f_{k,b} : \mathcal{X} \to \mathcal{D}_\mathcal{Y} \right\}_{k \in \mathcal{K}_\mathcal{F}, b \in \{0,1\}}$$

is called a noisy trapdoor claw free (NTCF) family *if the following conditions hold:*

1. ***Efficient Function Generation.*** *There exists an efficient probabilistic algorithm $GEN_\mathcal{F}$ which generates a key $k \in \mathcal{K}_\mathcal{F}$ together with a trapdoor t_k:*

$$(k, t_k) \leftarrow GEN_\mathcal{F}(1^\lambda).$$

2. ***Trapdoor Injective Pair.*** *For all keys $k \in \mathcal{K}_\mathcal{F}$ the following conditions hold.*
   (a) Trapdoor: There exists an efficient deterministic algorithm $INV_\mathcal{F}$ such that for all $b \in \{0,1\}$, $x \in \mathcal{X}$ and $y \in \mathsf{SUPP}(f_{k,b}(x))$, $INV_\mathcal{F}(t_k, b, y) = x$. Note that this implies that for all $b \in \{0,1\}$ and $x \neq x' \in \mathcal{X}$, $\mathsf{SUPP}(f_{k,b}(x)) \cap \mathsf{SUPP}(f_{k,b}(x')) = \emptyset$.
   (b) Injective pair: There exists a perfect matching $\mathcal{R}_k \subseteq \mathcal{X} \times \mathcal{X}$ such that $f_{k,0}(x_0) = f_{k,1}(x_1)$ if and only if $(x_0, x_1) \in \mathcal{R}_k$.
3. ***Efficient Range Superposition.*** *For all keys $k \in \mathcal{K}_\mathcal{F}$ and $b \in \{0,1\}$ there exists a function $f'_{k,b} : \mathcal{X} \to \mathcal{D}_\mathcal{Y}$ such that the following hold.*
   (a) For all $(x_0, x_1) \in \mathcal{R}_k$ and $y \in \mathsf{SUPP}(f'_{k,b}(x_b))$, $INV_\mathcal{F}(t_k, b, y) = x_b$ and $INV_\mathcal{F}(t_k, b \oplus 1, y) = x_{b \oplus 1}$.
   (b) There exists an efficient deterministic procedure $CHK_\mathcal{F}$ that, on input k, $b \in \{0,1\}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, returns 1 if $y \in \mathsf{SUPP}(f'_{k,b}(x))$ and 0 otherwise. Note that $CHK_\mathcal{F}$ is not provided the trapdoor t_k.
   (c) For every k and $b \in \{0,1\}$,

$$\mathbb{E}_{x \leftarrow_U \mathcal{X}} \left[H^2(f_{k,b}(x), f'_{k,b}(x)) \right] \leq \mu(\lambda),$$

for some negligible function $\mu(\cdot)$. Here H^2 is the Hellinger distance. Moreover, there exists an efficient procedure SAMP$_{\mathcal{F}}$ that on input k and $b \in \{0,1\}$ prepares the state

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \sqrt{(f'_{k,b}(x))(y)}\, |x\rangle\, |y\rangle\,.$$

4. ***Adaptive Hardcore Bit.*** *For all keys $k \in \mathcal{K}_{\mathcal{F}}$ the following conditions hold, for some integer w that is a polynomially bounded function of λ.*
   (a) *For all $b \in \{0,1\}$ and $x \in \mathcal{X}$, there exists a set $G_{k,b,x} \subseteq \{0,1\}^w$ such that $\Pr_{d\leftarrow_U \{0,1\}^w}[d \notin G_{k,b,x}]$ is negligible, and moreover there exists an efficient algorithm that checks for membership in $G_{k,b,x}$ given k, b, x and the trapdoor t_k.*
   (b) *There is an efficiently computable injection $\mathcal{J} : \mathcal{X} \to \{0,1\}^w$, such that $\mathcal{J}$ can be inverted efficiently on its range, and such that the following holds. If*

$$H_k = \big\{(b, x_b, d, d \cdot (\mathcal{J}(x_0) \oplus \mathcal{J}(x_1))) \mid b \in \{0,1\}, (x_0, x_1) \in \mathcal{R}_k, d \in G_{k,0,x_0} \cap G_{k,1,x_1}\big\},$$
$$\overline{H}_k = \{(b, x_b, d, c) \mid (b, x, d, c \oplus 1) \in H_k\big\},$$

then for any quantum polynomial-time procedure $\mathcal{A}$ there exists a negligible function $\mu(\cdot)$ such that

$$\left| \Pr_{(k,t_k)\leftarrow GEN_{\mathcal{F}}(1^\lambda)}[\mathcal{A}(k) \in H_k] - \Pr_{(k,t_k)\leftarrow GEN_{\mathcal{F}}(1^\lambda)}[\mathcal{A}(k) \in \overline{H}_k] \right| \le \mu(\lambda).$$

7.2 (Extended) NTCF from LWE

Theorem 7.2 ([BCM$^+$21, Theorem 4.1] [Mah18, Theorem 9.2])**.** *Assuming the post-quantum hardness of LWE_{n,m,q,B_L}, (extended) NTCF families exist.*

The following construction description is mostly taken verbatim from [Mah18].

7.2.1 Parameter Choice

Let λ be the security parameter. All other parameters are functions of λ. Let $q \ge 2$ be a prime integer. Let $\ell, n, m, w \ge 1$ be polynomially bounded functions of λ and B_L, B_V, B_P be Gaussian parameters, B_X, B_S be norm bounds, such that the following conditions hold:

1. $n = \Omega(\ell \log q)$ and $m = \Omega(n \log q)$
2. $w = n\lceil \log q\rceil$.
3. $B_P = \frac{q}{2C_T\sqrt{mn\log q}}$ where C_T is the constant in theorem 6.4.
4. $2\sqrt{n} \le B_L \le B_V \le B_P \le B_X$
5. The ratios $\frac{B_V}{B_L}, \frac{B_P}{B_V}, \frac{B_{P'}}{B_P}, \frac{B_X}{B_S}$ are all super-polynomial in λ.
6. We denote $[B_X]$ as all integer taking values $[-B_X, -B_X + 1 \cdots, B_X - 1, B_X]$. Similarly for B_S. B_S can in fact be $\{0,1\}$.

7.2.2 Noisy Trapdoor claw-Free Families (2-to-1 Mode)

Let Let $\mathcal{X} = \mathbb{Z}_q^n$ and $\mathcal{Y} = \mathbb{Z}_q^m$. The key space is $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$. For $b \in \{0,1\}$, $\mathbf{x} \in \mathcal{X}$ and key $k = (\mathbf{A}, \mathbf{s}\mathbf{A} + \mathbf{e})$ where $(\mathbf{A}, \mathsf{td}_A) \leftarrow \mathsf{GenTrap}(1^n, 1^m, q)$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}_q^m, B_L}$, the density $f_{k,b}(x)$ is defined as:

$$\forall \mathbf{y} \in \mathcal{Y} : (f_{k,b}(\mathbf{x}))(\mathbf{y}) = \mathcal{D}_{\mathbb{Z}_q^m, B_P}(\mathbf{y} - \mathbf{x}^\top \mathbf{A} - b \cdot \mathbf{s}^\top \mathbf{A})$$

It follows that:

$$\text{SUPP}(f_{0,k}(\mathbf{x})) = \{\mathbf{y} = \mathbf{x}\mathbf{A} + \mathbf{e}' | \|\mathbf{e}'\| \leq B_P\sqrt{m}\}$$
$$\text{SUPP}(f_{1,k}(\mathbf{x})) = \{\mathbf{y} = \mathbf{x}\mathbf{A} + \mathbf{e}' + \mathbf{s}\mathbf{A} | \|\mathbf{e}'\| \leq B_P\sqrt{m}\}$$

$(\mathbf{x}_0, \mathbf{x}_1)$ will be an injective pair such that $f_{0,k}(\mathbf{x}_0) = f_{1,k}(\mathbf{x}_1)$ if and only if $\mathbf{x}_0 = \mathbf{x}_1 + \mathbf{s}$.

7.2.3 Noisy Trapdoor Injective Families (Injective Mode)

We now describe the trapdoor injective functions, or informally, the "injective mode" of trapdoor claw-free functions. Let Let $\mathcal{X} = \mathbb{Z}_q^n$ and $\mathcal{Y} = \mathbb{Z}_q^m$. The key space is $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$.

For $b \in \{0, 1\}$, $\mathbf{x} \in \mathcal{X}$ and key $k = (\mathbf{A}, \mathbf{u})$, where $(\mathbf{A}, \mathsf{td}_A) \leftarrow \mathsf{GenTrap}(1^n, 1^m, q)$, $\mathbf{u}$ is sampled uniformly random up to the condition that there *does not* exist $\mathbf{s}, \mathbf{e}$ such that $\mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top = \mathbf{u}$ and $\|\mathbf{e}\| \leq \frac{q}{C_T\sqrt{n \log q}}$, which happens with all but negligible probability.

The density $g_{k,b}(x)$ is defined as:

$$\forall \mathbf{y} \in \mathcal{Y} : (g_{k,b}(\mathbf{x}))(\mathbf{y}) = \mathcal{D}_{\mathbb{Z}_q^m, B_P}(\mathbf{y} - \mathbf{x}\mathbf{A} - b \cdot \mathbf{u})$$

The injective trapdoor functions $g_{b,k}$ looks like follows:

$$\text{SUPP}(g_{0,k}(\mathbf{x})) = \{\mathbf{y} = \mathbf{x}^\top \mathbf{A} + \mathbf{e}' | \|\mathbf{e}'\| \leq B_P\sqrt{m}\}$$
$$\text{SUPP}(g_{1,k}(\mathbf{x})) = \{\mathbf{y} = \mathbf{x}\mathbf{A} + \mathbf{e}' + \mathbf{u} | \|\mathbf{e}'\| \leq B_P\sqrt{m}\}$$

$\text{SUPP}(g_{0,k}(\mathbf{x}))$ and $\text{SUPP}(g_{1,k}(\mathbf{x}))$ are disjoint as long as $B_P \leq \frac{q}{2C_T\sqrt{mn \log q}}$.

There is also an inversion function INV in the injective mode: $\mathsf{INV}(\mathsf{td}, \mathbf{y}) \to \mathbf{x}$ will give some unique $\mathbf{x}$ on input $(\mathsf{td}, \mathbf{y})$.

Lemma 7.3. *[Mah18] The 2-to-1 mode and injective mode are computationally indistinguishable by* LWE_{n,m,q,B_L}.

7.2.4 Efficient Range Preparation for NTCF with Small LWE Secrets

We refer the reader to Section 4.3 of [BCM+21] for a detailed description of $\text{SAMP}_{\mathsf{LWE}}$ procedure to efficiently prepapre the claw state: $(|0, \mathbf{x}_0\rangle + |1, \mathbf{x}_1\rangle), y$. We describe it here briefly for the sake of coherence in presentation.

In order for our security parameters in the reduction to go through, we deviate slightly from the exact parameters on $\mathcal{X}$ and $\mathbf{s}$ in [BCM+21]. We work with small secrets $\mathcal{X} = [B_X]^n$ and $\mathbf{s} \in [B_S]^n$.

At the first step, the procedure creates the following superposition:

$$\sum_{\mathbf{e}' \in \mathbb{Z}_q^m} \sqrt{\mathcal{D}_{\mathbb{Z}_q, B_P}(\mathbf{e}')}\, |\mathbf{e}'\rangle$$

In step 2 of $\text{SAMP}_{\mathsf{LWE}}$, we prepare uniform superposition of $\mathbf{x} \in [B_X]$.

$$\frac{1}{\sqrt{2|B_X|^n}} \sum_{b \in \{0,1\}, \mathbf{x}, \mathbf{e}'} \sqrt{\mathcal{D}_{\mathbb{Z}_q, B_P}(\mathbf{e}')} \, |b, \mathbf{x}\rangle \, |\mathbf{e}'\rangle$$

The rest of the steps are the same. In step 3, we apply the key $(\mathbf{A}, \mathbf{u})$, controlled by the bit b to indicate whether to use $\mathbf{u}$, where $\mathbf{u} = \mathbf{s}\mathbf{A} + \mathbf{e}$ in the 2-to-1 setting and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ in the injective mode. We get a state:

$$\frac{1}{\sqrt{2|B_X|^n}} \sum_{b \in \{0,1\}, \mathbf{x}, \mathbf{e}'} \sqrt{\mathcal{D}_{\mathbb{Z}_q, B_P}(\mathbf{e}')} \, |b, \mathbf{x}\rangle \, |\mathbf{e}'\rangle \, |\mathbf{x}\mathbf{A} + \mathbf{e}' + b \cdot \mathbf{u}\rangle$$

Next, we uncompute the register containing $\mathbf{e}'$ using the information in register containing $\mathbf{x}$, the key $(\mathbf{A}, \mathbf{u})$ and the last register.

It is easy to observe that the efficient range preparation in [BCM+21] Section 4.3 and acquirance of the claw state also works in our setting with our choice of parameters having B_X/B_S superpolynomially large, or simply letting $B_S = \{0, 1\}$ and B_X a superpolynomial.

With probability $(1 - \mathsf{negl}(\lambda))$, when one measures the image register to obtain a value $\mathbf{y}$, we will obtain the state $\frac{1}{\sqrt{2}}(|0, \mathbf{x}_0\rangle + |1, \mathbf{x}_1\rangle)$ where $f_{0,k}(\mathbf{x}_0) = f_{1,k}(\mathbf{x}_1) = \mathbf{y}$

7.3 Parallel Repetition of An NTCF-based Protocol

We first define the single-instance game from [RS19]. The game is abstracted as a "1-of-2" puzzle with "2-of-2 soundness", where the verifier randomly asks the prover to output a preimage $\mathbf{x} \in \mathcal{X}$ or an adaptive hardcore bit for the same image $\mathbf{y} \in \mathcal{Y}$.

Definition 7.4 (1-of-2 Puzzle from NTCF [RS19]). *The protocol proceeds as follows, using the notations from Section 7.1.*

- *The verifier samples a key* $(k, \mathsf{td}) \leftarrow \mathsf{GEN}_{\mathcal{F}}(1^\lambda)$ *and send* k *to the prover. The verifier keeps the trapdoors* td
- *The prover sends back a committed image value* $\mathbf{y}$.
- *The verifier samples a random bit* $\delta \in \{0, 1\}$ *and sends* δ *to the prover.*
- *If* $\delta = 0$, *the prover sends back some* $\mathbf{x} \in \mathcal{X}$; *else if* $b = 1$, *the prover sends back a string* $(c, \mathbf{d})$.
- *The verifier does the following checks on each* $(\mathbf{y}, \mathbf{x})$ *or* $(\mathbf{y}, c, \mathbf{d})$:
  - *When* $\delta = 0$: *Check* $\mathbf{x} \in \mathsf{INV}(\mathsf{td}, b \in \{0, 1\}, \mathbf{y})$ [6].
  - *When* $\delta = 1$: *Find both* $\mathbf{x}_0, \mathbf{x}_1$ *using* $\mathsf{INV}(\mathsf{td}, b \in \{0, 1\}, \mathbf{y})$. *Check if* $c = \mathbf{d} \cdot (\mathcal{J}(\mathbf{x}_0) \oplus \mathcal{J}(\mathbf{x}_1))$.

[RS19] showed the following property for the above protocol using the LWE-based NTCF from [BCM+21].

1-of-2 Completeness: Any BQP prover will answer one of the challenges for $\delta = 0$ or $\delta = 1$ with probability 1.

2-of-2 Soundness: The 2-of-2 soundness error in the above protocol is the probability that a prover can provide both the 1-challenge answer $\mathbf{x}$ and the 0-challenge answer $(c, \mathbf{d})$ correctly.

The above protocol has 2-of-2 soundness $1/2$ for any BQP prover [RS19, BCM+21].

[6] This step can also be performed publicly using $CHK_{\mathcal{F}}$.

Parallel Repitition We now describe a special type of parallel-repeated protocol based on the NTCF. In this protocol, we only consider the "2-of-2" setting: the verifier samples multiple keys independently; for every single key, the verifier simply asks the prover to provide both the answer to the 0-challenge and the answer to the 1-challenge.

Its parallel repetition soundness was shown in [RS19].

Definition 7.5 (Parallel-Repeated 2-of-2 NTCF-protocol))**.** *The protocol proceeds as follows, using the notations from Section 7.1.*

- *The verifier samples ℓ number of keys $(k_i, \mathsf{td}_i) \leftarrow \mathsf{GEN}_{\mathcal{F}}(1^\lambda), i \in [\ell]$ independently and send $\{k_i\}_{i\in[\ell]}$ to the prover. The verifier keeps the trapdoors $\{\mathsf{td}_i\}_{i\in[\ell]}$*
- *The prover sends back ℓ tuple of values $\{(\mathbf{y}_i, \mathbf{x}_i, c_i, \mathbf{d}_i)\}_{i\in[\ell]}$.*
- *The verifier does the following checks on each $(\mathbf{y}_i, \mathbf{x}_i, c_i, \mathbf{d}_i)$ for $i \in [\ell]$:*
  - *Find both $\mathbf{x}_{i,0}, \mathbf{x}_{i,1}$ using $\mathsf{INV}(\mathsf{td}_i, b \in \{0,1\}, \mathbf{y}_i)$.*
  - *Check if $c_i = \mathbf{d}_i \cdot (\mathcal{J}(\mathbf{x}_{i,0}) \oplus \mathcal{J}(\mathbf{x}_{i,1}))$.*
- *If all the checks pass, the verifier outputs 1; else outputs 0.*

Theorem 7.6 (Parallel Repetition Soundness of NTCF-based Protocol, [RS19] Theorem 15 rephrased)**.** *The above protocol has soundness $(1 - \mathsf{negl}(\ell))$ for any BQP prover.*

Remark 7.7. *Note that in our construction Section 9.2 we let the verifier further check if $\mathbf{y}_i, \mathbf{x}_i$ are well-formed. This will not affect the above soundness because it only puts more restrictions on the prover/adversary.*

Remark 7.8. *In this work, we only need the soundness to use the above simple protocol where we require the adversary to produce both the "preimage answer" and the "adaptive hardcore bit answer" at the same time. Clearly, the "completeness" of the above protocol is not well-defined, but we omit this property in our setting.*

We do not need the more complicated version of repetition in [Mah18] studied in [CCY20, AF16].

8 Secure Key Leasing with Classical Communication: Definition

8.1 Secure Key Leasing PKE with Classical Lessor

Definition 8.1 (Public Key Encryption with Classcal Leaser)**.** *A PKE with secure key leasing with a classical vendor consists of the algorithms (*Setup, KeyGen, Enc, Dec, Delete, VerDel*) defined as follows:*

$\mathsf{Setup}(1^\lambda)$*: take input a security parameter λ, output a (classical) master public key* mpk *and a (classical) trapdoor* td.

$\mathsf{KeyGen}(1^\lambda)$*: take as input a (classical) public key* mpk*, output a quantum decryption key ρ_{sk} and a classical public key* pk.

$\mathsf{Enc}(\mathsf{pk}, \mu)$*: given a public key* pk *and a plaintext $\mu \in \{0,1\}$, output a ciphertext* ct.

$\mathsf{Dec}(\rho_{\mathsf{sk}}, \mathsf{ct})$*: given a quantum state ρ_{pk} and a ciphertext* ct*, output a message μ and the state ρ'_{sk}*

$\mathsf{Delete}(\rho_{\mathsf{sk}})$*: given the quantum state ρ_{sk}, output a classical deletion certificate* cert

$\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert})$*: given a public key* pk*, a classical certificate* cert *and the trapdoor* td*, output* Valid *or* $\mathsf{Invalid}$.

We refer the readers to Appendix A on a few side comments and remarks about the definition.

Correctness A PKE Scheme with secure quantum key leasing (Setup, KeyGen, Enc, Dec, Delete, VerDel) satisfies correctness if the following hold.

Decryption Correctness: There exists a negligible function $\mathsf{negl}(\cdot)$, for all $\lambda \in \mathbb{N}$, for all $\mu \in \mathcal{M}$:

$$\Pr\left[\mathsf{Dec}(\rho_{\mathsf{sk}}, \mathsf{ct}) = \mu : \begin{array}{c}(\mathsf{mpk}, \mathsf{td}) \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathsf{pk}, \rho_{\mathsf{sk}}) \leftarrow \mathsf{KeyGen}(\mathsf{mpk}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mu)\end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

Reusability the above decryption correctness should hold for an arbitrary polynomial number of uses.

Verifying Deletion Correctness : There exists a negligible function $\mathsf{negl}(\cdot)$, for all $\lambda \in \mathbb{N}$:

$$\Pr\left[\mathsf{Valid} \leftarrow \mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert}) : \begin{array}{c}(\mathsf{mpk}, \mathsf{td}) \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathsf{pk}, \rho_{\mathsf{sk}}) \leftarrow \mathsf{KeyGen}(\mathsf{mpk}) \\ \mathsf{cert} \leftarrow \mathsf{Delete}(\rho_{\mathsf{sk}})\end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

IND-SKL-PKE Security We give a "classical friendly" security definition same as the one used in [APV23, AKN+23], except that we have a classical leaser.

Then in the next subsection Section 8.2, we will then present a "strong" security definition in the following section, section 8.2. The latter is the one we will actually use in the proof and will imply the following IND-PKE-SKL security.

Definition 8.2 (IND-PKE-SKL Security(Classical Client)). *where the experiment IND-PKE-SKL$(\mathcal{A}, 1^\lambda, b \in \{0,1\})$ between a challenger and the adversary $\mathcal{A}$ is defined as follows:*

- *The challenger runs* $\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{mpk}, \mathsf{td})$. *It sends* mpk *to the adversary* $\mathcal{A}$. $\mathcal{A}$ *computes* $(\mathsf{pk}, \rho_{\mathsf{sk}}) \leftarrow \mathsf{KeyGen}(\mathsf{mpk})$ *and publishes* pk.
- *The challenger requests that* $\mathcal{A}$ *runs the deletion algorithm* Delete. $\mathcal{A}$ *returns a deletion certificate* cert.
- *The challenger runs* $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert})$ *and continues if* $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert})$ *outputs* Valid*; else the challenger outputs* $\perp$ *and aborts.*
- $\mathcal{A}$ *submits a plaintext* $\mu \in \{0,1\}^\ell$ *to the challenger.*
- *The challenger flips a bit* $b \xleftarrow{\$} \{0,1\}$.
- *If* $b = 0$, *the challenger sends back the ciphertext* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mu)$. *If* $b = 1$ *the challenger sends a random ciphertext from the possible space of all ciphertexts for of* ℓ*-bit messages,* $\mathsf{ct} \xleftarrow{\$} \mathcal{C}$
- *Output* $\mathcal{A}$*'s guess for the bit* b, b'.

*A PKE Scheme with Secure Key Leasing and fully classical communication (*Setup, KeyGen, Enc, Dec, Delete, VerDel*) is secure if, for every* QPT *adversary* $\mathcal{A}$ *if there exists negligible functions* $\mathsf{negl}(\cdot)$ *such that one of the following holds for all* $\lambda \in \mathbb{N}$*:*

$$|\Pr\left[\textit{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b = 0) = 1\right] - \Pr\left[\textit{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b = 1) = 1\right]| \leq \mathsf{negl}(\lambda)$$

Remark 8.3. *Regarding the security in Definition 8.2: In other words, in order to win, the adversary* $\mathcal{A}$ *needs to do both of the following for some noticeable* ϵ_1, ϵ_2*:*

1. $|\Pr\left[\textit{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b = 0) = 1\right] - \Pr\left[\textit{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b = 1) = 1\right]| \geq \epsilon_1(\lambda)$ *and*
2. $\Pr[\textit{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b \in \{0,1\}) \neq \bot] \geq \epsilon_2(\lambda)$

We need the second inequality above to hold because in the case where $\Pr[\textit{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b \in \{0,1\}) = \bot] \geq 1 - \mathsf{negl}_2(\lambda)$,*for some negligible* $\mathsf{negl}_2(\cdot)$, *the probabilities* $\Pr\left[\textit{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b = 0) = 1\right]$ *and* $\Pr\left[\textit{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b = 1) = 1\right]$ *will also have negligible difference.*

8.2 Strong SKL-PKE PKE Security: Threshold Implementation Version

In this section, we define a security notion we call Strong SKL-PKE, which is described via the measurement TI in Section 5. We show that it implies the regular security notion in the previous section.

To define the strong security/implementable security, we first define what it means to test the success probability of a quantum decryptor.

Definition 8.4 (Testing a quantum decryptor)**.** *Let* $\gamma \in [0,1]$. *Let* pk *be a public key and* μ *be a message. We refer to the following procedure as a test for a* γ*-good quantum decryptor with respect to* pk *and* μ *used in the following sampling procedure:*

- *The procedure takes as input a quantum decryptor* ρ.
- *Let* $\mathcal{P} = (P, I - P)$ *be the following mixture of projective measurements (in terms of Definition 5.5) acting on some quantum state* ρ*:*
  - *Compute* $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{pk}, \mu)$, *the encryption of message* $\mu \in \{0,1\}$.
  - *Compute* $\mathsf{ct}_1 \leftarrow \mathcal{C}$, *a random ciphertext from the possible space of all ciphertexts for 1-bit messages.*
  - *Sample a uniform* $b \leftarrow \{0,1\}$.
  - *Run the quantum decryptor* ρ *on input* ct_b. *Check whether the outcome is* b. *If so, output* 1, *otherwise output* 0.
- *Let* $\mathsf{TI}_{1/2+\gamma}(\mathcal{P})$ *be the threshold implementation of* $\mathcal{P}$ *with threshold value* $\frac{1}{2} + \gamma$, *as defined in Definition 5.3. Run* $\mathsf{TI}_{1/2+\gamma}(\mathcal{P})$ *on* ρ, *and output the outcome. If the output is* 1, *we say that the test passed, otherwise the test failed.*

By Lemma 5.4, we have the following corollary.

Corollary 8.5 (γ-good Decryptor)**.** *Let* $\gamma \in [0,1]$. *Let* ρ *be a quantum decryptor. Let* $\mathsf{TI}_{1/2+\gamma}(\mathcal{P})$ *be the test for a* γ*-good decryptor defined above. Then, the post-measurement state conditioned on output* 1 *is a mixture of states which are in the span of all eigenvectors of* P *with eigenvalues at least* $1/2 + \gamma$.

Now we are ready to define the strong γ-anti-piracy game.

Definition 8.6 (γ-Strong Secure Key Leasing Security Game)**.** *Let* $\lambda \in \mathbb{N}^+$, *and* $\gamma \in [0,1]$. *The strong* γ*-PKE-SKL game is the following game between a challenger and an adversary* $\mathcal{A}$.

1. *The challenger runs* $\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{mpk}, \mathsf{td})$. *It sends* mpk *to the adversary* $\mathcal{A}$. $\mathcal{A}$ *computes* $(\mathsf{pk}, \rho_{\mathsf{sk}}) \leftarrow \mathsf{KeyGen}(\mathsf{mpk})$ *and publishes* pk.
2. *The challenger requests that* $\mathcal{A}$ *runs the deletion algorithm* $\mathsf{Delete}(\rho_{\mathsf{sk}})$. $\mathcal{A}$ *returns a deletion certificate* cert *to the challenger.*

3. *The challenger runs* $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert})$ *and continues if* $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert})$ *returns* Valid*; else it outputs* $\bot$ *and aborts, if* $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert})$ *returns* $\mathsf{Invalid}$.
4. $\mathcal{A}$ *outputs a message* μ *and a (possibly mixed) state* ρ_{Delete} *as a quantum decryptor.*
5. *The challenger runs the test for a* γ*-good decryptor on* ρ_{Delete} *with respect to* pk *and* μ*. The challenger outputs* 1 *if the test passes, otherwise outputs* 0.

We denote by $\mathsf{StrongSKL}(1^\lambda, \gamma, \mathcal{A})$ *a random variable for the output of the game.*

Definition 8.7 (Strong PKE-SKL Security)**.** *Let* $\gamma : \mathbb{N}^+ \to [0, 1]$*. A secure key leasing scheme satisfies strong* γ*-SKL security, if for any QPT adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$*:*

$$\Pr\left[b = 1, b \leftarrow \mathsf{StrongSKL}(1^\lambda, \gamma(\lambda), \mathcal{A})\right] \leq \mathsf{negl}(\lambda) \tag{3}$$

Next, we show that the strong PKE-SKL definition(Definition 8.7) implies the IND-PKE-SKL definition(Definition 8.2).

Claim 8.8. *Suppose a secure key leasing scheme satisfies strong* γ*-SKL security (Definition 8.7) for any inverse polynomial* γ*, then it also satisfies IND-PKE-SKL security (Definition 8.2).*

Proof. We refer the reader to appendix A for the proof. □

9 Secure Key Leasing with Classical Lessor/Client: Construction

9.1 Parameters

We present our parameters requirements for the following construction. All the parameters are the same as in Section 7.2.1, with a few new paramters added.

Let λ be the security parameter. All other parameters are functions of λ. Let $q \geq 2$ be a prime integer. Let $\ell, k, n, m, w \geq 1$ be polynomially bounded functions of λ and $B_L, B_V, B_P, B_{P'}$ be Gaussian parameters and B_X, B_S are norm parameters such that the following conditions hold:

1. $n = \Omega(\ell \log q)$ and $m = \Omega(n \log q)$
2. k can be set to λ.
3. $w = \lceil n \log q \rceil$.
4. $B_P = \frac{q}{2C_T\sqrt{mn \log q}}$ where C_T is the constant in Theorem 6.4.
5. $2\sqrt{n} \leq B_L \leq B_V \leq B_P \leq B_X \leq B_{P'}$
6. The ratios $\frac{B_V}{B_L}$, $\frac{B_P}{B_V}$, $\frac{B_{P'}}{B_P}$, $\frac{B_X}{B_S}$, and $\frac{B_{P'}}{B_X^2 \cdot B_P}$ are super-polynomial in λ.
7. $B_{P'} \cdot m^d \leq q$ where d is the depth of the circuit in FHE evaluation, to be further discussed in Appendix B.
8. We denote $[B_X]$ as all integer taking values $[-B_X, \cdots, B_X]$. Similarly for B_S. We can also simply take $B_S := \{0, 1\}$ in our scheme.

9.2 Scheme Construction

We first present our construction for a secure key leasing protocol. How the protocol works with a classical client comes naturally with our defintion in Definition 8.2. Nevertheless, we provide a protocol description in Appendix D.

- $\mathsf{Setup}(1^\lambda)$: On input the security parameter 1^λ, the Setup algorithm works as follows:
  - Sample $k = k(\lambda)$ matrices $\mathbf{A}_i \in \mathbb{Z}_q^{n\times m}$ along with their trapdoors td_i using the procedure $\mathsf{GenTrap}(1^n, 1^m, q)$(Theorem 6.4): $(\mathbf{A}_i, \mathsf{td}_i) \xleftarrow{\$} \mathsf{GenTrap}(1^n, 1^m, q)$, $\forall i \in [k]$.
  - Sample $\mathbf{s}_i \xleftarrow{\$} [B_s]^n, \forall i \in [k]$ and $\mathbf{e}_i \xleftarrow{\$} D_{\mathbb{Z}_q^{1\times m}, B_V}, , \forall i \in [k]$.
  - Output $\mathsf{mpk} = \{f_{i,0}, f_{i,1}\}_{i=1}^k = \{(\mathbf{A}_i, \mathbf{s}_i\mathbf{A}_i + \mathbf{e}_i)\}_{i=1}^k$ and the trapdoor $\mathsf{td} = \{\mathsf{td}_i\}_{i=1}^k$.
- $\mathsf{KeyGen}(\mathsf{mpk})$:
  - Take in $\mathsf{mpk} = \{f_{i,0}, f_{i,1}\}_{i=1}^k = \{(\mathbf{A}_i, \mathbf{s}_i\mathbf{A}_i + \mathbf{e}_i)\}_{i=1}^k$.
  - Prepare the quantum key of the form $\rho_{\mathsf{sk}} = \bigotimes_{i=1}^k \left(\frac{1}{\sqrt{2}}(|0, \mathbf{x}_{i,0}\rangle + |1, \mathbf{x}_{i,1}\rangle)\right)$ along with $\{\mathbf{y}_i\}_{i\in[k]}$ where $\mathbf{y}_i = f_{i,0}(\mathbf{x}_{0,i}) = f_{i,1}(\mathbf{x}_{1,i})$ for each $i \in [k]$, according to the procedure in Section 7.2.4.
    Note that $f_{i,b}(\mathbf{x}_{b,i}) = \mathbf{x}_{b,i}\mathbf{A}_i + \mathbf{e}'_i + b_i \cdot \mathbf{s}_i\mathbf{A}_i,\ \mathbf{e}'_i \xleftarrow{\$} D_{Z_q^m, B_P}; \mathbf{x}_{i,b_i} \in [B_X]^n, \forall i \in [k]$.
  - Output public key $\mathsf{pk} = \{(\mathbf{A}_i, \mathbf{s}_i\mathbf{A}_i + \mathbf{e}_i, \mathbf{y}_i)\}_{i=1}^k$ and quantum decryption key ρ_{sk}.
- $\mathsf{Enc}(\mathsf{pk}, \mu)$: On input a public key $\mathsf{pk} = \{(\mathbf{A}_i, \mathbf{s}_i\mathbf{A}_i + \mathbf{e}_i, \mathbf{y}_i)\}_{i=1}^k$ and a plaintext $\mu \in \{0,1\}$ the algorithm samples $\mathbf{R} \xleftarrow{\$} \{0,1\}^{m\times m}$ and computes a ciphertext as follows:

$$\mathsf{ct} = \begin{bmatrix} \mathbf{s}_1\mathbf{A}_1 + \mathbf{e}_1 \\ \mathbf{A}_1 \\ \cdots \\ \cdots \\ \mathbf{s}_k\mathbf{A}_k + \mathbf{e}_k \\ \mathbf{A}_k \\ \sum_{i\in[k]} \mathbf{y}_i \end{bmatrix} \cdot \mathbf{R} + \mathbf{E} + \mu \cdot \mathbf{G}_{(n+1)k+1}$$

  where $\mathbf{G}_{(n+1)k+1}$ is the gadget matrix of dimensions $(nk + k + 1) \times m$.
  $\mathbf{E} \in \mathbb{Z}_q^{(nk+k+1)\times m}$ is a matrix with all rows having $\mathbf{0}^m$ except the last row being $\mathbf{e}''$, where $\mathbf{e}'' \leftarrow \mathcal{D}_{\mathbb{Z}_q^m, B_{P'}}$.
  Output ct.
- $\mathsf{Dec}(\rho_{\mathsf{sk}}, \mathsf{ct})$: On quantum decryption key $\rho_{\mathsf{sk}} = \bigotimes_{i=1}^k \left(\frac{1}{\sqrt{2}}(|0, \mathbf{x}_{i,0}\rangle + |1, \mathbf{x}_{i,1}\rangle)\right)$ and a ciphertext $\mathsf{ct} \in \mathbb{Z}_q^{(nk+k+1)\times m}$, the decryption is performed in a coherent way as follows.
  - View the key as a vector of dimension $1 \times (n+1)k$; pad the key with one-bit of classical information, the value -1 at the end, to obtain a vector with dimension $1 \times (nk + k + 1)$:

$$\begin{aligned} &\left[\tfrac{1}{\sqrt{2}}((|0, \mathbf{x}_{1,0})\rangle + |(1, \mathbf{x}_{1,1})\rangle)| \cdots |\tfrac{1}{\sqrt{2}}(|(0, \mathbf{x}_{k,0})\rangle + |(1, \mathbf{x}_{k,1})\rangle)|\,|-1\rangle\right] \\ &= \frac{1}{\sqrt{2^k}} \sum_{\mathbf{b}_j\in\{0,1\}^k} |\mathbf{b}_{j,1}, \mathbf{x}_{1,\mathbf{b}_{j,1}}, \cdots, \mathbf{b}_{j,k}, \mathbf{x}_{k,\mathbf{b}_{j,k}}\rangle\,|-1\rangle \end{aligned}$$

    Denote this above vector by $\mathbf{sk}$.
  - Compute $(-\mathbf{sk}) \cdot \mathsf{ct} \cdot \mathbf{G}^{-1}(\mathbf{0}^{nk+k}|\lfloor\frac{q}{2}\rfloor)$ coherently and write the result on an additional empty register work (a working register other than the one holding $\mathbf{sk}$).

- Make the following computation in another additional register out: write $\mu' = 0$ if the outcome in the previous register of the above computation is less than $\frac{q}{4}$; write $\mu' = 1$ otherwise. Uncompute the work register in the previous step using $\mathbf{sk}$ and ct. Measure the final out register and output the measurement outcome.

- $\mathsf{Delete}(\rho_{\mathsf{sk}})$:
  - For convenience, we name the register holding state $\frac{1}{\sqrt{2}}(|0, \mathbf{x}_{i,0}\rangle + |1, \mathbf{x}_{i,1}\rangle)$ in ρ_{sk} as register $\mathsf{reg_i}$.
  - For each register $\mathsf{reg_i}$, $i \in [k]$: apply invertible function $\mathcal{J} : \mathcal{X} \to \{0,1\}^w$ where $\mathcal{J}(x)$ returns the binary decomposition of x. Since it is invertible, we can uncompute the original $\mathbf{x}_{i,b_i}$'s in $\mathsf{reg_i}$ and leave only $|0, \mathcal{J}(\mathbf{x}_{i,0})\rangle + |1, \mathcal{J}(\mathbf{x}_{i,1})\rangle$.
  - Apply a quantum Fourier transform over $\mathbb{F}_2$ to all $k(w+1)$ qubits in registers $\{\mathsf{reg}_i\}_{i\in[k]}$
  - measure in computational basis to obtain a string $(c_1, \mathbf{d}_1, \cdots, c_k, \mathbf{d}_k) \in \{0,1\}^{wk+k}$.

- $\mathsf{VerDel}(\mathsf{td}, \mathsf{pk}, \mathsf{cert})$: On input a deletion certificate $\mathsf{cert}\{c_i, \mathbf{d}_i\}_{i=1}^n$, public key $\mathsf{pk} = \{f_{i,0}, f_{i,1}, \mathbf{y}_i\}_{i\in k}$ and the trapdoor $\mathsf{td} = \{\mathsf{td}_i\}_{i\in k}$.
  - Compute $\mathbf{x}_{i,b_i} \leftarrow \mathsf{INV}(\mathsf{td}_i, b, \mathbf{y}_i)$ for both $b = 0, 1$.
  - Check if $\mathbf{x}_{i,b_i} \in [B_X]^n$ for all $i \in [k], b_i \in \{0,1\}$. If not, output Invalid. If yes, continue.
  - Check if $\|\mathbf{y}_i - \mathbf{x}_{i,b_i} - b_i \cdot \mathbf{s}_i\mathbf{A}_i\| \le B_P\sqrt{m}$, for all $i \in [k], b_i \in \{0,1\}$. If not, output Invalid. If yes, continue.
  - Output Valid if $c_i = \mathbf{d}_i \cdot (\mathcal{J}(\mathbf{x}_{i,0}) \oplus \mathcal{J}(\mathbf{x}_{i,1}) (\mod 2)), \forall i \in [k]$ and Invalid otherwise.

We defer the proof on correctness to Section 9.3 and the security proof to Section 10.

9.3 Correctness

Decryption Correctness The above scheme satisfies decryption correctness.

Proof. Note that we can write out the entire quantum key as $\rho_{\mathsf{sk}} = \frac{1}{\sqrt{2^k}} \sum_{\mathbf{b}_j \in\{0,1\}^k} |\mathbf{b}_{j,1}, \mathbf{x}_{k,\mathbf{b}_{j,1}}, \cdots, \mathbf{b}_{j,k}, \mathbf{x}_{1,\mathbf{b}_{j,k}}\rangle$ By applying the decryption procedure, we will have:

$$= \frac{1}{\sqrt{2^k}} \sum_{\mathbf{b}_j\in\{0,1\}^k} |\mathbf{b}_{j,1}, \mathbf{x}_{1,\mathbf{b}_{j,1}}, \cdots, \mathbf{b}_{j,k}, \mathbf{x}_{k,\mathbf{b}_{j,k}}\rangle|(-\sum_{i\in[k]}(\mathbf{x}_{i,\mathbf{b}_{j,i}}\mathbf{A}_i + \mathbf{b}_{j,i}(\mathbf{s}_i\mathbf{A}_i + \mathbf{e}_i)) + \sum_{i\in[k]}\mathbf{y}_i)$$

$$\cdot \mathbf{R}\mathbf{G}^{-1}(\mathbf{0}|\lfloor\frac{q}{2}\rfloor) + \mathbf{e}'' \cdot \mathbf{G}^{-1}(\mathbf{0}|\lfloor\frac{q}{2}\rfloor) + \mathbf{sk} \cdot \mu\mathbf{G}\mathbf{G}^{-1}(\mathbf{0}|\lfloor\frac{q}{2}\rfloor)\rangle_{\mathsf{work}} |0\rangle_{\mathsf{out}}$$

$$= \frac{1}{\sqrt{2^k}} \sum_{\mathbf{b_j}\in(\{0,1\}^k} |\mathbf{b}_{j,1}, \mathbf{x}_{1,\mathbf{b}_{j,1}}, \cdots, \mathbf{b}_{j,k}, \mathbf{x}_{k,\mathbf{b}_{j,k}}\rangle|((\sum_{i\in[k]} -\mathbf{b}_{j,i} \cdot \mathbf{e}_i + \mathbf{e}'_i) \cdot \mathbf{R} + \mathbf{e}'')\mathbf{G}^{-1}(\mathbf{0}|\lfloor\frac{q}{2}\rfloor) + \lfloor\frac{q}{2}\rfloor \cdot \mu\rangle_{\mathsf{work}} |0\rangle_{\mathsf{out}}$$

$$= \frac{1}{\sqrt{2^k}} \sum_{\mathbf{b_j}\in(\{0,1\}^k} |\mathbf{b}_{j,1}, \mathbf{x}_{1,\mathbf{b}_{j,1}}, \cdots, \mathbf{b}_{j,k}, \mathbf{x}_{k,\mathbf{b}_{j,k}}\rangle\, |\mu\rangle_{\mathsf{out}} \text{ (round, write on out and uncompute work)}$$

□

Since we have $\|\sum_i(\mathbf{e}'_i + \mathbf{e}_i)\| \le k \cdot \sqrt{m} \cdot B_P$, $\|\mathbf{e}''\| \le \sqrt{m} \cdot B_{P'}$ and $\|(\sum_{i\in[k]}(\mathbf{b}_i \cdot \mathbf{e}'_i + \mathbf{e}_i) \cdot \mathbf{R} + \mathbf{e}'')\mathbf{G}^{-1}(\mathbf{0}|\lfloor\frac{q}{2}\rfloor)\| \le \frac{q}{4}$ for all support $\mathbf{b}_j \in \{0,1\}^k$ in the final outcome, we will obtain μ with all but negligible probability, for all support in the above state. We can write the final output for μ in the third (rightmost) register and use ct to uncompute the second register and recover ρ_{sk}.

Reusability Reusability of the decryption key follows from correctness and the gentle measurement lemma [Aar18].

Deletion Verification Correctness The first two steps in VerDel will pass for any honestly prepared $\{\mathbf{y}_i\}_{i\in[k]}$ with $(1 - \mathsf{negl}(\lambda))$ probability.

The Delete procedure will operate on the state ρ_{sk} as follows. For each $\frac{1}{\sqrt{2}}(|0, \mathbf{x}_{i,0}\rangle + |1, \mathbf{x}_{i,1}\rangle)$, $i \in [k]$, the Delete procedure will turn the state into:

$$\begin{aligned}
&\frac{1}{\sqrt{2}}\sum |0, \mathcal{J}(\mathbf{x}_{i,0})\rangle + |1, \mathcal{J}(\mathbf{x}_{i,1})\rangle \text{ after applying } \mathcal{J}(\cdot) \text{ and uncomputing } \mathbf{x}_{i,b_i} \text{ register}\\
&\to \frac{1}{\sqrt{2^{w+2}}}\sum_{\mathbf{d}_i,b,u}(-1)^{\mathbf{d}\cdot\mathcal{J}(x_{i,b})\oplus ub}\,|u\rangle\,|\mathbf{d}_i\rangle \text{ after QFT}\\
&= \frac{1}{\sqrt{2^w}}\sum_{\mathbf{d}_i\in\{0,1\}^w}(-1)^{\mathbf{d}_i\cdot\mathcal{J}(\mathbf{x}_{i,0})}\,|\mathbf{d}_i \cdot (\mathcal{J}(\mathbf{x}_{i,0}) \oplus \mathcal{J}(\mathbf{x}_{i,1}))\rangle\,|\mathbf{d}_i\rangle
\end{aligned}$$

A measurement in the computational basis will give us result $(c_i = \mathbf{d}_i \cdot (\mathcal{J}(\mathbf{x}_{i,0}) \oplus \mathcal{J}(\mathbf{x}_{i,1})), \mathbf{d}_i)$, for any $i \in [k]$. Correctness of deletion verification thus follows.

Remark 9.1. *Note that in our construction, using* $\mathtt{td}$ *one can also decrypt. It is easy to define a decryption procedure using* $\mathtt{td}$ *by the* $\mathsf{INV}(\mathtt{td}, \mathtt{b}, \mathbf{y})$*-procedure in Section 7.2. We omit the details here.*

10 Security Proof for SKL-PKE

Theorem 10.1 (Security)**.** *Assuming the post-quantum sub-expoential hardness of* LWE_{n,m,q,B_L} *with parameter choice in Section 7.2.1, the construction in Section 9.2 satisfies the* γ*-strong SKL-PKE security define in Definition 8.7 for any noticeable* γ*.*

To prove security we consider two hybrids. The first hybrid $\mathbf{Hybrid}_0$ corresponds to the real security game whereas $\mathbf{Hybrid}_1$ a correspond to modified games. We will show that these hybrids are indistinguishable and so the winning probability in the three hybrids are negligibly close and then in the final hybrid $\mathbf{Hybrid}_1$ it must be negiligible.

We will prove the following statements:

1. Probability of winning in $\mathbf{Hybrid}_0$ and $\mathbf{Hybrid}_1$ are close by a negligible amount if δ are set to be $\lambda^{-\omega(1)}$ for a tiny super constant $\omega(1)$ (we can in fact set it to be exponentially small).
2. We will then prove that if LWE satisfies subexponential security, for the set parameters LWE_{n,m,q,B_L} probability of winning in $\mathbf{Hybrid}_1$ is negligible.

Together these claims imply that the probability of winning in $\mathbf{Hybrid}_0$ is negligible.

Claim 1 follows from Lemma 5.7: if the inefficient γ-good decryptor test outputs 1 with probability p on a state ρ, then the efficient $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D},1/2+\gamma-\epsilon}$ will output 1 on the state ρ with probability $p - \delta$. Since δ is negligible, $\mathcal{A}$'s overall wining probability will have negligible difference.

Figure 1: Hybrid 0

Hybrid 0

In this hybrid, the adversary and the challenger play the security game as in Definition 8.6.

1. The challenger runs $\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{td})$. It sends mpk to the adversary $\mathcal{A}$. $\mathcal{A}$ computes $(\mathsf{pk}, \rho_{\mathsf{sk}}) \leftarrow \mathsf{KeyGen}(\mathsf{mpk})$ and publishes pk.
2. The challenger requests that $\mathcal{A}$ runs the deletion algorithm $\mathsf{Delete}(\rho_{\mathsf{sk}})$. $\mathcal{A}$ returns a deletion certificate cert to the challenger.
3. The challenger runs $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert})$ and continues if $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert})$ returns Valid; else it outputs $\perp$ and aborts, if $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert})$ returns $\mathsf{Invalid}$.
4. $\mathcal{A}$ outputs a message μ and a (possibly mixed) state ρ_{Delete} as a quantum decryptor.
5. The challenger runs the test for a γ-good decryptor on ρ_{Delete} with respect to pk and μ (using $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}})$). The challenger outputs 1 if the test passes, otherwise outputs 0.

Figure 2: Hybrid 1

Hybrid 1

In this hybrid, we replace the check for γ good decryptor with an efficient check $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma}$ where we set δ to be $\lambda^{-\omega(1)}$ for a tiny super-constant $\omega(1)$ and ϵ to be an inverse polynomial smaller than γ by a large constant factor, e.g. $\epsilon = \frac{\gamma}{100}$.

1. The challenger runs $\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{td})$. It sends mpk to the adversary $\mathcal{A}$. $\mathcal{A}$ computes $(\mathsf{pk}, \rho_{\mathsf{sk}}) \leftarrow \mathsf{KeyGen}(\mathsf{mpk})$ and publishes pk.
2. The challenger requests that $\mathcal{A}$ runs the deletion algorithm $\mathsf{Delete}(\rho_{\mathsf{sk}})$. $\mathcal{A}$ returns a deletion certificate cert to the challenger.
3. The challenger runs $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert})$ and if $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert})$ returns Valid it outpus z; else it outputs $\perp$ if $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert})$ returns $\mathsf{Invalid}$.
4. $\mathcal{A}$ outputs a message μ and a (possibly mixed) state ρ_{Delete} as a quantum decryptor.
5. The challenger runs the (efficient) test $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma+1/2-\epsilon}(\rho_{\mathsf{Delete}})$ with respect to μ and pk. The challenger sets $z = 1$ if the test passes, otherwise it sets $z = 0$.

10.1 Winning Probability in Hybrid 1

Next, we show that $\Pr[\mathbf{Hybrid}_1 = 1] \leq \mathsf{negl}(\lambda)$ for some negligible $\mathsf{negl}(\cdot)$. We will reduce to the security of the parallel repeated NTCF game in Theorem 7.6.

Lemma 10.2. *Assuming post-quantum subexponential hardness of* LWE_{m,m,q,B_V} *with parameter choice in Section 7.2.1, we have* $\Pr[\mathbf{Hybrid}_1 = 1] \leq \mathsf{negl}(\lambda)$ *for some negligible* $\mathsf{negl}(\cdot)$.

To show that the winning probability in **Hybrid 1** is negligible, we consider a world where we **do not check the deletion certificate and let the adversary pass all the time**. In this world, through a sequence of hybrid games: we will call them **Games** instead of Hybrids to distinguish from the above Hybrids 0, 1.

Later, we will show how to put back the condition about the deletion certificate check for our analysis via the following argument:

Notations for Events For simplicity, we make a few notations for the events that take place in **Hybrid 1**:

- We denote the event that the adversary hands in a valid deletion cerficate, i.e. $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert}) = \mathsf{Valid}$, as $\mathsf{CertPass}$.
- We denote the event that test $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma+1/2}(\rho_{\mathsf{Delete}})$ outputs 1 with respect to μ and pk, as $\mathsf{GoodDecryptor}$. To simplify notations, we define the new γ here to be the $\gamma - \epsilon$ in Hybrid 1.
- We denote the event that we can obtain the preimages $\{\mathbf{x}_i\}_{i\in[k]} \in \{\mathrm{INV}(\mathsf{td}_i, b \in \{0,1\}, \mathbf{y}_i)\}_{i\in[k]}$ as Ext.

Suppose the probability that final output 1 in Hybrid 1 (Figure 2) is some noticeable ϵ, then we must have $\Pr[\mathsf{CertPass} \wedge \mathsf{GoodDecryptor}] \geq \epsilon_1$. To build a reduction that breaks the security of parallel repeated NTCF game Theorem 7.6, we need the following statement to hold: $\Pr[\mathsf{CertPass} \wedge \mathsf{Ext}] \geq \epsilon'$, for some noticeable ϵ', because in this case the reduction can obtain both the deletion certificates $\{c_i, \mathbf{d}_i\}_{i\in[k]}$ and the preimages $\{\mathbf{x}_{i,b}\}_{i\in[k]}$, which allow it to win the parallel repeated NTCF game.

Our proof outline is the follows: we would show that when $\mathsf{GoodDecryptor}$ happens, Ext *always happens* except with negligible probability. Therefore, we have $\Pr[\mathsf{CertPass} \wedge \mathsf{Ext}] \geq \epsilon_1 - \mathsf{negl}(\lambda)$ by a simple proabability observaion (Claim C.1).

We analyze the probabilities by defining some games in the world where we don't check the deletion certificate and reasoning about them.

Game 0 This is an experiment same as the one in Figure 2, using the construction Section 9.2, except that *the challenger does not perform check on the deletion certificate, i.e. step 5 in Figure 2*.

1. The challenger runs $\mathsf{Setup}(1^\lambda)$: the challenger prepares $\mathsf{mpk} = \{\mathbf{A}_i, \mathbf{s}_i\mathbf{A}_i + \mathbf{e}_i\}_{i\in[k]}$, where $(\mathbf{A}_i, \mathsf{td}_i) \leftarrow \mathsf{GenTrap}(1^n, 1^m, q), \forall i \in [k]$ and sends it to $\mathcal{A}$. The challenger keeps $\mathsf{td} = \{\mathsf{td}_i\}_{i\in[k]}$
2. $\mathcal{A}$ receives mpk and obtains the classical public key $\mathsf{pk} = \{\mathbf{A}_i, \mathbf{s}_i\mathbf{A}_i + \mathbf{e}_i, \mathbf{y}_i\}_{i\in[k]} \leftarrow \mathsf{KeyGen}(1^\lambda)$ and one copy of quantum decryption key ρ_{sk}. $\mathcal{A}$ publishes pk.
3. $\mathcal{A}$ outputs a message μ and a (possibly mixed) state ρ_{Delete} as a quantum decryptor.
4. The challenger runs the (efficient) test $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma+1/2}(\rho_{\mathsf{Delete}})$ with respect to μ and pk. The challenger outputs 1 if the test passes, otherwise it outputs 0.

Game 1 This is the same as Game 0 except that all $\mathbf{A}_i$ are sampled uniformly at random, without a trapdoor.

1. The challenger runs $\mathsf{Setup}(1^\lambda)$: the challenger prepares $\mathsf{mpk} = \{\mathbf{A}_i, \mathbf{s}_i\mathbf{A}_i + \mathbf{e}_i\}_{i\in[k]}$, where $\mathbf{A}_i \leftarrow \mathbb{Z}_q^{n\times m}, \forall i \in [k]$ and sends it to $\mathcal{A}$.
2. $\mathcal{A}$ receives mpk and obtains the classical public key $\mathsf{pk} = \{\mathbf{A}_i, \mathbf{s}_i\mathbf{A}_i+\mathbf{e}_i, \mathbf{y}_i\}_{i\in[k]} \leftarrow \mathsf{KeyGen}(1^\lambda)$ and one copy of quantum decryption key ρ_{sk}. $\mathcal{A}$ publishes pk.
   $\mathcal{A}$ outputs a message μ and a (possibly mixed) state ρ_{Delete} as a quantum decryptor.
3. The challenger runs the (efficient) test $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma+1/2}(\rho_{\mathsf{Delete}})$ with respect to μ and pk. The challenger outputs 1 if the test passes, otherwise it outputs 0.

Game 2.j: $j = 1, \cdots, k$ This is the same as Game 0 except the following:

1. During Setup,
   - For $i \leq j$: the challenger prepares $\mathsf{mpk}_i = (\mathbf{A}_i, \mathbf{u}_i)$, where $\mathbf{A}_i \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{u}_i \leftarrow \mathbb{Z}_q^{1\times m}$ uniformly random.
   - For $i > j$: the challenger prepares $\mathsf{mpk}_i = (\mathbf{A}_i, \mathbf{u}_i = \mathbf{s}_i\mathbf{A}_i + \mathbf{e}_i)$ the same as in hybrid 0.
2. $\mathcal{A}$ accordingly obtains public key $\mathsf{pk} = \{\mathbf{A}_i, \mathbf{u}_i, \mathbf{y}_i\}_{i\in[k]}$. and one copy of quantum decryption key ρ_{sk}. $\mathcal{A}$ publishes pk.
3. $\mathcal{A}$ outputs a message μ and a (possibly mixed) state ρ_{Delete} as a quantum decryptor.
4. The challenger runs the (efficient) test $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma+1/2}(\rho_{\mathsf{Delete}})$ with respect to μ and pk. The challenger outputs 1 if the test passes, otherwise it outputs 0.

We then prove the following claims about the above games:

Claim 10.3. *Game 0 and Game 1 are statistically indistinguishable.*

This follows directly from the property of $\mathsf{GenTrap}$ inTheorem 6.4.

Claim 10.4. *Assuming the hardness of* LWE_{n,m,q,B_L}*, Game 1 and Game 2.k are indistinguishable.*

Proof. We claim that each pair in (Game 1, Game 1.1), (Game 1.1, Game 1.2) ... (Game 1.(k-1), Game 1.k) is indistinguishable. If anyone of them are distinguishable, then there exists some j such that there is an adversary that distinguishes $(\mathbf{A}_j, \mathbf{s}_j\mathbf{A}_j + \mathbf{e}_j)$ and $(\mathbf{A}_j, \mathbf{u}_j \leftarrow \mathbb{Z}_q^m)$. □

Remark 10.5. *The above property can also directly follow from the indistinguishability between the 2-to-1 mode and injective mode of NTCF [Mah18] (see Lemma 7.3). Note that after switching to* $(\mathbf{A}_i, \mathbf{u}_i)$*, some of the* $\mathbf{y}_i$*'s in the public key* $(\{\mathbf{A}_i, \mathbf{u}_i, \mathbf{y}_i\}$ *may have the format* $\mathbf{y}_i = \mathbf{x}_{i,1}\mathbf{A}_i + \mathbf{e}'_i + \mathbf{u}_i$ *or* $\mathbf{y}_i = \mathbf{x}_{i,0}\mathbf{A}_i + \mathbf{e}'_i$.

Thus, an honestly encrypted ciphertext ct *in **Game** 2.k for message* μ *should have the following format:*

$$\mathsf{ct} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{A}_1 \\ \\ \cdots \\ \mathbf{u}_k \\ \mathbf{A}_k \\ \sum_{i\in[k]}(\mathbf{x}_i\mathbf{A}_i + \mathbf{e}'_i + b_i \cdot \mathbf{u}_i) \end{bmatrix} \cdot \mathbf{R} + \mathbf{E} + \mu \cdot \mathbf{G}_{(n+1)k+1} \tag{4}$$

where $\mathbf{A}_i \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{u}_i \leftarrow \mathbb{Z}_q^m, \mathbf{R} \leftarrow \{0,1\}^{m\times m}$ *and the* $(nk+k+1)$*-th row in* $\mathbf{E}$ *is* $\mathbf{e}'' \leftarrow \mathcal{D}_{\mathbb{Z}_q^m, B_{P'}}$*. Note that* $b_i = 0$ *or* $, 1, i \in [k]$ *are some adversarially chosen bits that come in the* $\{\mathbf{y}_i\}_{i\in[k]}$ *part of* pk.

Switching to Plaintext 0 From now on, without loss of generality, we always consider encrypting message $\mu = 0$. The analysis for the case when $\mu = 1$ should follow symmetrically.

10.2 Extraction of Preimages via LWE Search-Decision Reduction

Now we are ready to argue that in **Game** $2.k$, if the game outputs 1, then there exists an extractor that extracts all preimages $\{\mathbf{x}_{i,b_i}\}_{i\in[k]}, b_i = 0$ or 1 for $\{\mathbf{y}_i\}_{i\in[k]}$.

Theorem 10.6. *In the last Game* $2.k$*, if we have* $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D},1/2+\gamma}(\rho_{\mathsf{Delete}})$ *outputs 1, for some noticeable* γ*, then there exists an extractor* Ext *such that there is a negligible function* $\mathsf{negl}(\cdot)$*:*

$$\Pr[\mathsf{Ext}(\rho_{\mathsf{Delete}}, \mathsf{pk}) \to (\mathbf{x}_1, \cdots, \mathbf{x}_k) : \mathbf{x}_i \textit{ is the secret in } \mathbf{x}_i\mathbf{A}_i + \mathbf{e}_{i,0}] \geq 1 - \mathsf{negl}(\lambda)$$

Ext *runs in time* $T' = T \cdot knB_{\mathbf{x}} \cdot \mathsf{poly}(1/\epsilon, 1/\log\delta)$*, where* T *is the running time of the decryptor* ρ_{Delete} *and* $B_{\mathbf{x}} = \mathsf{max}_{\mathbf{x}_{i,j}}|\mathbf{x}_{i,j}|$ *and* $\mathbf{x}_{i,j}$ *is the* j*-th entry in vector* $\mathbf{x}_i$*.*

In other words, we have in Game 2.k:

$$\Pr\left[\begin{array}{c}\mathsf{Ext}(\rho_{\mathsf{Delete}}, \mathsf{pk}) \to (\mathbf{x}_1, \cdots \mathbf{x}_k) : \\ \mathbf{x}_i \textit{ is the secret in } \mathbf{x}_i\mathbf{A}_i + \mathbf{e}'_i\end{array} |\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_{2.k},1/2+\gamma}(\rho_{\mathsf{Delete}}) = 1\right] \geq 1 - \mathsf{negl}(\lambda)$$

We obtain the following corollary from Theorem 10.6:

Corollary 10.7. *Assuming the subexponential post-quantum hardness security of* LWE_{n,m,q,B_L}*, in Game 0, for any QPT* $\mathcal{A}$ *with auxiliary quantum input, there exists some negligible* $\mathsf{negl}(\cdot)$*, it holds that:*

$$\Pr\left[\begin{array}{c}\mathsf{Ext}(\rho_{\mathsf{Delete}}, \mathsf{pk}) \to (\mathbf{x}_1, \cdots \mathbf{x}_k) : \\ \mathbf{x}_i \in \mathrm{INV}(\mathsf{td}_i, b \in \{0,1\}, \mathbf{y}_i)\end{array} |\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_0,1/2+\gamma}(\rho_{\mathsf{Delete}}) = 1\right] \geq 1 - \mathsf{negl}(\lambda)$$

We refer to Appendix C for proof details.

10.2.1 Proof Outline

We first give a high level description of the proof for Theorem 10.6.

As discussed above, we need to show the following: given that we have a good decryptor ρ_{Delete}, we will have to extract all preimages $\{\mathbf{x}_{i,b_i}\}_{i\in[k]}, b_i = 0$ or 1 for $\{\mathbf{y}_i\}_{i\in[k]}$ with probability negligibly close to 1. Accordingly, we have that if the event $\mathsf{GoodDecryptor}$ happens with some probability ϵ_1, then the event Ext happens with probability $\epsilon_1 - \mathsf{negl}(\lambda)$.

The procedure will be a quantum analogue of a LWE search-to-decision reduction with extraction probability close to 1, where the input distributions to the search-game adversary are not exactly LWE versus random, but statistically close to such.

Three TI Distributions For clarity, we consider a few simplifications:

- We use the ideal, projective threshold implementation TI in our algorithm.
- We consider the number of instances/repetions to be $k = 1$.

In the full proof, we will use the efficient ATI and polynomial k, via similar arguments.

To prove our theorem, it suffices to show that $\Pr[\text{Extraction of}\mathbf{x}] \geq 1 - \mathsf{negl}(\lambda)$ in a world where we are given the following condition at the beginning of the algorithm:

$$\mathsf{TI}_{\gamma+1/2}(\mathcal{P}_{\mathcal{D}_{\mathsf{ct}}}) \cdot \rho_{\mathsf{Delete}} = 1$$

where $\mathcal{P}_{\mathcal{D}_{\mathsf{ct}}}$ is the following mixture of projections, acting on the state ρ_{Delete}:

- Compute $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{pk}, 0)$ *in Game* $2.k$. ct_0 will have the format of Equation (4) with $\mu = 0$ and $k = 1$:

$$\mathsf{ct}_0 = \begin{bmatrix} \mathbf{u} \\ \mathbf{A} \\ \mathbf{x}\mathbf{A} + \mathbf{e}' + b \cdot \mathbf{u} \end{bmatrix} \cdot \mathbf{R} + \mathbf{E} = \begin{bmatrix} \mathbf{u}\mathbf{R} \\ \mathbf{A}\mathbf{R} \\ \mathbf{x}\mathbf{A}\mathbf{R} + \mathbf{e}'\mathbf{R} + b \cdot \mathbf{u}\mathbf{R} + \mathbf{e}'' \end{bmatrix}$$

  where $(\mathbf{A}, \mathbf{u})$ are already given in the public key pk; the rest is sampling $\mathbf{R} \leftarrow \{0,1\}^{m\times m}$ and the $(n+2)$-th row in $\mathbf{E}$ is $\mathbf{e}_i'' \leftarrow \mathcal{D}_{\mathbb{Z}_q^m, B_{P'}}$. Note that $b = 0$ or 1, is an adversarially chosen bit that come in the $\mathbf{y}$ part of pk.
- Compute $\mathsf{ct}_1 \leftarrow \mathcal{C}$, a random ciphertext from the possible space of all ciphertexts for 1-bit messages.
- Sample a uniform bit $\ell \leftarrow \{0,1\}$ [7].
- Run the quantum decryptor ρ on input ct_ℓ. Check whether the outcome is ℓ. If so, output 1, otherwise output 0.

We then consider a second threshold implementation $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}(g_i)})$. $\mathcal{P}_{\mathcal{D}(g_i)}$ is the following mixture of measurements (we denote the following distribution we sample from as $\mathcal{D}(g_i)$.):

- Let g_i be a guess for the i-th entry in vector $\mathbf{x}$.
- Sample a random $\mathbf{c} \leftarrow \mathbb{Z}_q^{1\times m}$, and let matrix $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$ be a matrix where the i-th row is $\mathbf{c}$ and the rest of rows are $\mathbf{0}$'s.
- Prepare ct_0 as follows:

$$\mathsf{ct}_0 = \begin{bmatrix} \mathbf{u}\mathbf{R} \\ \mathbf{A}\mathbf{R} + \mathbf{C} \\ \mathbf{x}\mathbf{A}\mathbf{R} + \mathbf{e}'\mathbf{R} + b \cdot \mathbf{u}\mathbf{R} + \mathbf{e}'' + g_i \cdot \mathbf{c} \end{bmatrix}$$

  where $(\mathbf{A}, \mathbf{u})$ are already given in the public key pk; $\mathbf{R} \leftarrow \{0,1\}^{m\times m}$ and $\mathbf{e}_i'' \leftarrow \mathcal{D}_{\mathbb{Z}_q^m, B_{P'}}$. Note that $b = 0$ or 1, is an adversarially chosen bit that comes in the $\mathbf{y}$ part of pk.
- Compute $\mathsf{ct}_1 \leftarrow \mathcal{C}$, a random ciphertext from the possible space of all ciphertexts for 1-bit messages. In our case, that is: $\mathsf{ct}_1 \leftarrow \mathbb{Z}_q^{(n+2)\times m}$.
- Flip a bit $\ell \leftarrow \{0,1\}$.
- Run the quantum distinguisher ρ on input ct_ℓ. Check whether the outcome is ℓ. If so, output 1, otherwise output 0.

[7] To distinguish from the bit b in ct, we use the notation ℓ for this random coin

We finally consider a third threshold implementation, we call $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}_{\mathsf{unif}}})$:

- Compute both $\mathsf{ct}_0, \mathsf{ct}_1 \leftarrow \mathcal{C}$, as random ciphertexts from the possible space of all ciphertexts for 1-bit messages. In our case, that is: $\mathsf{ct}_0, \mathsf{ct}_1 \leftarrow \mathbb{Z}_q^{(n+2)\times m}$.
- Flip a bit $\ell \leftarrow \{0, 1\}$.
- Run the quantum distinguisher ρ on input ct_ℓ. Check whether the outcome is ℓ. If so, output 1, otherwise output 0.

The Extraction Algorithm We describe the extraction algorithm as follows. It takes input $\mathsf{pk} = (\mathbf{A}, \mathbf{u})$ and a quantum state ρ_{Delete}.

- Set our guess for $\mathbf{x}$ as $\mathbf{x}' = \mathbf{0} \in \mathbb{Z}_q^{1\times n}$ and $\mathbf{x}'_i$ is the i-th entry of $\mathbf{x}'$.
- For $i = 1, 2, \cdots, n$:
  - For $g_i = [-B_{\mathbf{x}}, B_{\mathbf{x}}]$, where $[-B_{\mathbf{x}}, B_{\mathbf{x}}]$ is the possible value range for $\mathbf{x}_i \in \mathbb{Z}_q$:
    1. Let ρ_{Delete} be the current state from the quantum distinguisher.
    2. Run $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}(g_i)})$ on ρ_{Delete} wih respect to pk.
    3. If $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}(g_i)})$ outputs 1, then set $\mathbf{x}'_i := g_i$ and let $i := i + 1$.
    4. If $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}(g_i)})$ outputs 0, the let $g_i := g_i + 1$ and go to step 1.
- Output $\mathbf{x}'$.

State-preserving Properties in the Extraction Algorithm We will show that the output in the above algorithm $\mathbf{x}'$ is equal to $\mathbf{x}$ in the $\mathbf{x}\mathbf{A} + \mathbf{e}'$ with probability $(1 - \mathsf{negl}(\lambda))$.

First recall that at the beginning of our algorithm, we are given the following condition:

$$\Pr[\mathsf{TI}_{\gamma+1/2}(\mathcal{P}_{\mathcal{D}_{\mathsf{ct}}}) \cdot \rho_{\mathsf{Delete}} \to 1] = 1$$

We then prove the following claims:

1. In the above algorithm, when our guess $g_i = \mathbf{x}_i$, we have:
   - $\mathcal{D}(g_i)$ is statistically indistinguishable from $\mathcal{D}_{\mathsf{ct}}$.
   - $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}(g_i)})$ outputs 1 on the current state ρ_{Delete} with $(1 - \mathsf{negl}(\lambda))$ probability.
   - After $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}(g_i)})$ outputs 1, the remaining state ρ_{Delete} is negligibly close in trace distance to the state before we perform $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}(g_i)})$.
2. In the above algorithm, when our guess $g_i \neq \mathbf{x}_i$, we have:
   - $\mathcal{D}(g_i)$ is statistically indistinguishable from $\mathcal{D}_{\mathsf{unif}}$.
   - $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}(g_i)})$ outputs 0 on the current state ρ_{Delete} with $(1 - \mathsf{negl}(\lambda))$ probability.
   - After $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}(g_i)})$ outputs 0, the remaining state ρ_{Delete} is negligibly close in trace distance to the state before we perform $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}(g_i)})$.

Combining the above arguments, we can conclude that the algorithm outputs $\mathbf{x}$ with probability $(1-\mathsf{negl}(\lambda))$: whenever $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}(g_i)})$ outputs 1, we know our guess is correct with probability $(1 - \mathsf{negl}(\lambda))$ and our state is almost undisturbed, so we can move on to guessing the next entry; whenever $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}(g_i)})$ outputs 0, we know our guess is incorrect with probability $(1 - \mathsf{negl}(\lambda))$ and nevetheless our state is almost undisturbed, we can move on to the next value for our guess. These invariants preserves throughout the above algorithm. Negligible errors will accumulate on the trace distance of ρ_{Delete} after every TI-measurement, but will stay negligible throughout the algorithm with our choice of parameters. By these guarantees, we obtain $\mathbf{x}$ with $(1 - \mathsf{negl}(\lambda))$ probability in the end.

We refer the readers to the following section, Section 11 for the full algorithm and analysis.

10.3 Reduction to Parallel NTCF Soundness

Now we prove Lemma 10.2 and Theorem 10.1 follows accordingly.

Now we can build a reduction to break the security of the parallel-repeated NTCF-based protocol in Theorem 7.6 as follows:

We plug in the condition that **we have to check the validity of the deletion certificate**. Recall our notations for the events happening in Hybrid 1:

- We denote the event that the adversary hands in a valid deletion cerficate, i.e. $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert}) = \mathsf{Valid}$, as $\mathsf{CertPass}$.
- We denote the event that test $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D,\gamma+1/2}(\rho_{\mathsf{Delete}})$ outputs 1 with respect to μ and pk, as $\mathsf{GoodDecryptor}$.
- We denote the event that we can obtain the preimages $\{\mathbf{x}_i\}_{i\in[k]} \in \{\mathrm{INV}(\mathsf{td}_i, b \in \{0,1\}, \mathbf{y}_i)\}_{i\in[k]}$ as Ext.

Suppose the adversary wins the γ-strong SKL-PKE game in Definition 8.7, then we must have $\Pr[\mathsf{CertPass} \wedge \mathsf{GoodDecryptor}] \geq 1/p$ for some noticeable $1/p$. By Corollary 10.7, we have $\Pr[\mathsf{Ext} \,|\mathsf{GoodDecryptor}] \geq 1 - \mathsf{negl}(\lambda)$. Therefore we have $\Pr[\mathsf{CertPass} \wedge \mathsf{Ext}] \geq 1/p - \mathsf{negl}'(\lambda)$ for some negligible $\mathsf{negl}'(\lambda)$. The relation can be easily observed from drawing a Venn diagram; we deduce it formally in Appendix C.3.

Now we can build a reduction to break the security of the parallel-repeated NTCF-based protocol in Theorem 7.6 as follows: the reduction plays as the challenger in the strong-SKL game. It passes the $\{f_{i,b}\}_{i\in[k],b\in\{0,1\}}$ from the NTCF challenger to the adversary $\mathcal{A}$ as mpk.

It takes $\{\mathbf{y}_i\}_{i\in[k]}$ and the deletion certificate $\{(c_i, \mathbf{d}_i)\}_{i\in[k]}$ from the adversary and runs the γ-good test on the its post-deletion state ρ_{Delete}. If the test fails, then abort; if the test passes, run the extractor Ext (from Theorem 10.6) on the post-test state ρ'_{Delete} to obtain $\{\mathbf{x}_i\}_{i\in[k]}$. Then it gives $\{\mathbf{y}_i\, c_i, \mathbf{d}_i), \mathbf{x}_i\}$ all to the NTCF-protocol challenger. By the above argument, if $\mathcal{A}$ wins with probability ϵ, then the reduction wins with probability $1/p - \mathsf{negl}(\lambda)$.

11 Proof for Theorem 10.6: Quantum LWE Search-to-Decision Algorithm with Almost-Perfect Extraction (Noisy Version)

In this section, we provide a full extraction algorithm in Game $2.k$ for extracting the $\{\mathbf{x}_{i,b_i}\}_{i\in[k]}$ and its analysis, to prove Theorem 10.6. We call it "noisy" search-to-decision algorithm because the distribution in Theorem 10.6 is not exactly LWE versus random, but statistically close to.

In Appendix E we provide a cleaner version of the algorithm and analysis where the input distributions are real LWE instances versus real uniform random instances, so that we have a first quantum LWE search-to-decision with almost perfection extraction, even with auxiliary quantum inputs. We believe such a statement for (plain) LWE search-to-decision reduction will be of independent use.

Notations For clarity, we take away the index b_i in the vector $\mathbf{x}_{i,b_i}$ since obtaining $\mathbf{x}_{i,0}$ or $\mathbf{x}_{i,1}$ does not affect our analysis. From now on, the subscripts i, j in $\mathbf{x}_{i,j}$ represents the j-th entry in the i-th vector $\mathbf{x}_i$.

11.1 Three Distributions for ATI

Similar to the proof outline in Section 10.2.1, we first describe three ATI's with respect to different distributions $\mathcal{D}$ (and accordingly, the mixture of projections $\mathcal{P}$).

ATI **for** $\mathcal{D}_{2.k}$**:** $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_{2.k},1/2+\gamma}$ is the approximate threshold implementation algorithm for the following mixture of projections $\mathcal{P}_{\mathcal{D}_{\mathsf{ct}}}$, acting on the state ρ_{Delete}:

- Compute $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{pk}, 0)$ *in Game* $2.k$. ct_0 will have the format of Equation (4) with $\mu = 0$:

$$\mathsf{ct}_0 = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{A}_1 \\ \\ \cdots \\ \mathbf{u}_k \\ \mathbf{A}_k \\ \sum_{i\in[k]}(\mathbf{x}_i\mathbf{A}_i + \mathbf{e}'_i + b_i \cdot \mathbf{u}_i) \end{bmatrix} \cdot \mathbf{R} + \mathbf{E}$$

$$= \begin{bmatrix} \mathbf{u}_1 \cdot \mathbf{R} \\ \mathbf{A}_1 \cdot \mathbf{R} \\ \\ \cdots \\ \mathbf{u}_k \cdot \mathbf{R} \\ \mathbf{A}_k \cdot \mathbf{R} \\ \sum_{i}(\mathbf{x}_i\mathbf{A}_i\mathbf{R} + \mathbf{e}'_i\mathbf{R} + b_i \cdot \mathbf{u}_i\mathbf{R}) + \mathbf{e}'' \end{bmatrix}$$

  where $\{\mathbf{A}_i, \mathbf{u}_i\}_{i\in[k]}$ are given in the public key pk; $\mathbf{R} \leftarrow \{0,1\}^{m\times m}$ and the $(n+2)$-th row in $\mathbf{E}$ is $\mathbf{e}'' \leftarrow \mathcal{D}_{\mathbb{Z}_q^m, B_{P'}}$. Note that $b_i = 0$ or 1, is an adversarially chosen bit that come in the $\mathbf{y}$ part of pk.
- Compute $\mathsf{ct}_1 \leftarrow \mathcal{C}$, a random ciphertext from the possible space of all ciphertexts for 1-bit messages. In our case, that is: $\mathsf{ct}_1 \leftarrow \mathbb{Z}_q^{(nk+k+1)\times m}$.
- Sample a uniform bit $b \leftarrow \{0,1\}$
- Run the quantum decryptor ρ on input ct_b. Check whether the outcome is b. If so, output 1, otherwise output 0.

ATI **for** $\mathcal{D}(g_{\ell,j})$**:** We then consider a second approximate threshold implementation $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}(g_{\ell,j}),1/2+\gamma}$. $\mathcal{P}_{\mathcal{D}(g_{\ell,j})}$ is the following mixture of measurements (we denote the following distribution we sample from as $\mathcal{D}(g_{\ell,j})$.):

- Let $g_{\ell,j}$ be a guess for the j-th entry in vector $\mathbf{x}_\ell \in (\mathbf{x}_1, \cdots, \mathbf{x}_k)$.
- Sample a random $\mathbf{c} \leftarrow \mathbb{Z}_q^{1\times m}$, and let matrix $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$ be a matrix where the j-th row is $\mathbf{c}$ and the rest of rows are $\mathbf{0}$'s.

- Prepare ct_0 as follows:

$$\mathsf{ct}_0 = \begin{bmatrix} \mathbf{u}_1\mathbf{R}_1 \\ \mathbf{A}_1\mathbf{R} \\ \cdots \\ \mathbf{u}_\ell\mathbf{R} \\ \mathbf{A}_\ell\mathbf{R} + \mathbf{C}_\ell \\ \cdots \\ \mathbf{u}_k\mathbf{R} \\ \mathbf{A}_k\mathbf{R} \\ \sum_{i\in[k]}(\mathbf{x}_i\mathbf{A}_i\mathbf{R} + \mathbf{e}_i'\mathbf{R} + b_i \cdot \mathbf{u}_i\mathbf{R}) + \mathbf{e}'' + g_{\ell,j} \cdot \mathbf{c} \end{bmatrix}$$

  where $\{\mathbf{A}_i, \mathbf{u}_i\}_{i\in[k]}$ are given in the public key pk; $\mathbf{R} \leftarrow \{0,1\}^{m\times m}$ and the $(n+2)$-th row in $\mathbf{E}$ is $\mathbf{e}'' \leftarrow \mathcal{D}_{\mathbb{Z}_q^m, B_{P'}}$. Note that $b_i = 0$ or 1, is an adversarially chosen bit that come in the $\mathbf{y}$ part of pk.
- Compute $\mathsf{ct}_1 \leftarrow \mathcal{C}$, a random ciphertext from the possible space of all ciphertexts for 1-bit messages. In our case, that is: $\mathsf{ct}_1 \leftarrow \mathbb{Z}_q^{(nk+k+1)\times m}$.
- Flip a bit $b \leftarrow \{0,1\}$.
- Run the quantum distinguisher ρ on input ct_b. Check whether the outcome is b. If so, output 1, otherwise output 0.

ATI for $\mathcal{D}_{\mathsf{unif}}$: We finally consider a third threshold implementation, we call $\mathsf{ATI}^{\epsilon,\delta}_{1/2+\gamma,\mathcal{P},\mathcal{D}_{\mathsf{unif}}}$:

- Compute both $\mathsf{ct}_0, \mathsf{ct}_1 \leftarrow \mathcal{C}$, as random ciphertexts from the possible space of all ciphertexts for 1-bit messages. In our case, that is: $\mathsf{ct}_0, \mathsf{ct}_1 \leftarrow \mathbb{Z}_q^{(nk+k+1)\times m}$.
- Flip a bit $b \leftarrow \{0,1\}$.
- Run the quantum distinguisher ρ on input ct_b. Check whether the outcome is b. If so, output 1, otherwise output 0.

11.2 Extraction Algorithm

We describe the extraction algorithm as follows. It takes input $\mathsf{pk} = (\{\mathbf{A}_i, \mathbf{u}_i\}_{i\in[k]}$ and a quantum state ρ_{Delete}.

- Let $\mathbf{x}'_{\ell,j}$ be register that stores the final guess for the j-th entry of $\mathbf{x}_\ell$.
- For $\ell = 1, \cdots, k$:
  - For $j = 1, 2, \cdots, n$:
    * For $g_{\ell,j} = [-B_x, B_x]$, where $[-B_x, B_x]$ is the possible value range for $\mathbf{x}_{\ell,j} \in \mathbb{Z}_q$:
      1. Let ρ_{Delete} be the current state from the quantum distinguisher.
      2. Run $\mathsf{ATI}^{\epsilon,\delta}_{1/2+\gamma-4\epsilon,\mathcal{P},\mathcal{D}(g_{\ell,j})}$ on ρ_{Delete} wih respect to pk, for some inverse-polynomial $\epsilon = \gamma/100$.
      3. If $\mathsf{ATI}^{\epsilon,\delta}_{1/2+\gamma-4\epsilon,\mathcal{P},\mathcal{D}(g_{\ell,j})}$ outputs 1, then set $\mathbf{x}'_{\ell,j} := g_{\ell,j}$ and move on to the next coordinate, let $j := j+1$ if $j < n$, else let $\ell := \ell+1, j = 1$.
      4. If $\mathsf{ATI}^{\epsilon,\delta}_{1/2+\gamma-4\epsilon,\mathcal{P},\mathcal{D}(g_{\ell,j})}$ outputs 0, the let $g_i := g_i + 1$ and go to step 1.
- Output $\mathbf{x}'$.

11.3 Analysis of the Extractor

We make the following claims:

Claim 11.1. *When the guess $g_{\ell,j} = \mathbf{x}_{\ell,j}$, the distributions $\mathcal{D}_{2.k}, \mathcal{D}(g_{\ell,j})$ are statistically close by distance $\eta_0 = \mathsf{poly}(k,m) \cdot (\frac{1}{q^n} + \frac{B_P}{B_{P'}})$.*

Proof. Note that the two distributions are the same except on how they sample ct_0.

The ciphertext distribution for ct_0 in $\mathcal{D}_{2.k}$ is statistically close to the following distribution:

$$\mathsf{ct}_0 = \begin{bmatrix} \mathbf{u}'_1 \\ \mathbf{A}'_1 \\ \\ \cdots \\ \mathbf{u}'_\ell \\ \mathbf{A}'_\ell \\ \cdots \\ \mathbf{u}'_k \\ \mathbf{A}'_k \\ \sum_i (\mathbf{x}_i \mathbf{A}'_i + \mathbf{e}'_{i,0} + b_i \cdot \mathbf{u}'_i) \end{bmatrix} \tag{5}$$

where $\mathbf{A}'_i \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{u}'_i \leftarrow \mathbb{Z}_q^{1\times m}$ are uniform random and given in the public key pk_i; $\mathbf{e}'_{i,0} \leftarrow \mathcal{D}_{\mathbb{Z}_q^m, B'_P}$, where B'_P/B_P is superpolynomial, for all $i \in [k]$; $b_i = 0$ or $1, i \in [k]$ are some arbitrary, adversarially chosen bits.

We can view the distribution change as two differences: We can view the first change as: $\sum \mathbf{y}_i \cdot \mathbf{R} + \mathbf{E}_{nk+k+1} = \sum(\mathbf{x}_{i,b_i}\mathbf{A}_i\mathbf{R} + \mathbf{e}'^{\top}_i\mathbf{R} + \mathbf{e}''_i + b_i \cdot \mathbf{u}_i\mathbf{R})$ in Equation (4) to $\sum(\mathbf{x}_i^\top \mathbf{A}_i\mathbf{R} + \mathbf{e}'^{\top}_{i,0})$, where both noise $\mathbf{e}'_{i,0}$ and $\mathbf{e}''_i$ are sampled from $\mathcal{D}_{\mathbb{Z}_q^m, B'_P}$. Since $B_{P'}/B_P(B_P$ for the Gaussian parameter of the error $\mathbf{e}'_i$ in $\mathbf{y}_j)$ is superpolynomial, we can apply noise-flooding/smudging (Lemma 6.5) and the two are statistically close by $k \cdot B_P/B_{P'}$.

The second difference is from using $(\mathbf{A}_i\mathbf{R}, \mathbf{u}_i\mathbf{R}, \sum(\mathbf{x}_{i,b_i}\mathbf{A}_i\mathbf{R} + \mathbf{e}'^{\top}_{i,0} + b_i \cdot \mathbf{u}_i\mathbf{R}))$ to using $(\mathbf{A}'_i \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{u}'_i \leftarrow \mathbb{Z}_q^m, \sum(\mathbf{x}_{i,b_i}\mathbf{A}'_i + \mathbf{e}'_{i,0} + b_i \cdot \mathbf{u}'_i))$. These two are $k \cdot q^{-n}$-close by the leftover hash lemma.

We then show that ct_0 in $\mathcal{D}(g_{\ell,j})$ is also close to this distribution. First we also apply noise flooding to replace every $\mathbf{e}'_i\mathbf{R} + \mathbf{e}''_i$ with $\mathbf{e}'_{i,0}$. We next replace $\mathbf{A}_i\mathbf{R}, \mathbf{u}_i\mathbf{R}$'s by the LHL with random $\mathbf{A}'_i, \mathbf{u}'_i$. We can ignore $\sum_i b_i \cdot \mathbf{u}'_i$ in the last row from our distribution, because b_i is known to the adversary and they are the same in both distributions.

Then we observe that when $g_{\ell,j} = \mathbf{x}_{\ell,j}$, we let $\mathbf{A}''_\ell = \mathbf{A}'_\ell + \mathbf{C}$ where $\mathbf{C}$ is everywhere 0 except the j-th row being uniformly random $\mathbf{c}$. We also have $\mathbf{x}_{\ell,j}\mathbf{A}''_\ell + \mathbf{e}'_{i,0} + g_{\ell,j} \cdot \mathbf{c} = [(\mathbf{x}_{\ell,j}(\mathbf{A}''_{\ell,1,j} + \mathbf{c}_1) + \sum_{i\neq j} \mathbf{x}_{\ell,j}(\mathbf{A}''_{\ell,1,i} + 0), \cdots, (\mathbf{x}_{\ell,j}(\mathbf{A}''_{\ell,m,j} + \mathbf{c}_m) + \sum_{i\neq i=j} \mathbf{x}_{\ell,j}(\mathbf{A}''_{\ell,m,i} + 0)] + \mathbf{e}'_{i,0} = \mathbf{x}_{\ell,j}\mathbf{A}''_\ell + \mathbf{e}'_{i,0}$, where $\mathbf{A}''_{\ell,x,y}$ denotes the entry in x-th column and y-th row of $\mathbf{A}''_\ell$.

□

Claim 11.2. *When the guess $g_{\ell,j} \neq \mathbf{x}_{\ell,j}$, the distributions $\mathcal{D}_{2.k}, \mathcal{D}_{\mathsf{unif}}$ are statistically close by distance $\eta_1 = \mathsf{poly}(k,m) \cdot (\frac{1}{q^n} + \frac{B_P}{B_{P'}})$.*

Proof. the two distributions are the same except on how they sample ct_0. ct_0 in $\mathcal{D}_{\mathsf{unif}}$ is uniformly sampled from $\mathbb{Z}_q^{(nk+k+1)\times m}$. It remains to show that ct_0 in $\mathcal{D}(g_{\ell,j})$ is close to this distribution.

Similar to Claim 11.1, we first apply noise flooding and then LHL to ct_0. Now let $\mathbf{A}''_\ell = \mathbf{A}'_\ell + \mathbf{C}, \mathbf{A}'_\ell \leftarrow \mathbb{Z}_q^{n\times m}$. $\mathbf{A}''_\ell$ is uniormly random because the only change is adding the uniform random vector $\mathbf{c}$ in its j-th row. Now we observe that when $g_{\ell,j} \neq \mathbf{x}_{\ell,j}$. We can ignore the term $\sum_i b_i \cdot \mathbf{u}'_i$ in the last row from our distribution, because b_i is known to the adversary and they are the same in both distributions. We consider the vector $\mathbf{w} = \mathbf{x}_{\ell,j}^\top \mathbf{A}''_\ell + \mathbf{e}'_{i,0} + g_{\ell,j} \cdot \mathbf{c} = [(g_{\ell,j} \cdot \mathbf{c}_1 + \sum_i \mathbf{x}_{\ell,i} \mathbf{A}''_{\ell,1,i}, \cdots, g_{\ell,j} \cdot \mathbf{c}_m + \sum_i \mathbf{x}_{\ell,i} \mathbf{A}''_{\ell,m,i}] + \mathbf{e}'_{i,0}$. Since $\mathbf{c}$ is uniformly random, the entire $\mathbf{w}$ now becomes uniformly random. Since the last row of ct_0 in $\mathcal{D}(g_{\ell,j})$ is $\sum_{i \neq \ell}(\mathbf{x}_{i,b_i}\mathbf{A}'_i + \mathbf{e}'_{i,0} + b_i \cdot \mathbf{u}'_i) + \mathbf{w}$, it is masked by $\mathbf{w}$ and becomes uniformly random.

□

Let us denote the probability of the measurement **outputting 1** on ρ by $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D},1/2+\gamma}\,\rho]$. Accordingly $1 - \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D},1/2+\gamma}\,\rho] := \Pr[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D},1/2+\gamma}\,\rho \to 0]$.

Corollary 11.3. *We make the following two claims. For any inverse polynomial ϵ and exponentially small δ, there exists exponentially small δ' such that:*

1. *If $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_{2.k},1/2+\gamma}\,\rho] = 1 - \delta$, then $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}(g_{\ell,j})),1/2+\gamma-\epsilon}\,\rho] \geq 1 - \delta' - \eta'_0$.*
2. *If $1 - \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_{\mathsf{unif}},1/2+\gamma}\,\rho] = 1 - \delta$, then $1 - \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}(g_{\ell,j})),1/2+\gamma-\epsilon}\,\rho] \geq 1 - \delta' - \eta'_1$.*

where $\eta'_0 = O(\eta_0 \cdot \mathsf{poly}(\lambda))$, $\eta'_1 = O(\eta_1 \cdot \mathsf{poly}(\lambda))$ and η_0, η_1 are the statistical distances between $(\mathcal{D}_{2.k}, \mathcal{D}(g_{\ell,j})$ and between $(\mathcal{D}_{\mathsf{unif}}, \mathcal{D}(g_{\ell,j})$ respectively.

This follows directly from Claim 11.1, Claim 11.2 and Theorem 5.10.

Claim 11.4. *For all $\ell \in [k], j \in [n]$, When the guess $g_{\ell,j} \neq \mathbf{x}_{\ell,j}$ in the above mixture of projections $(\mathcal{P}_{\mathcal{D}_{g_i}}, \mathbf{I} - \mathcal{P}_{\mathcal{D}_{g_i}})$, for any noticeable γ, and any quantum distinguisher ρ, there exists some function $\eta'_1 = O(\eta_1 \cdot \mathsf{poly}(\lambda))$ such that $1 - Tr[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_{\ell,j}),1/2+\gamma-\epsilon}\,\rho] = 1 - \eta'_1(\lambda)$, where η_1 is the statistical distance between $(\mathcal{D}(g_{\ell,j}), \mathcal{D}_{\mathsf{unif}})$.*

Proof. We first consider the perfect projective threshold implementation for the distribution $\mathcal{D}_{\mathsf{unif}}$ $\mathsf{TI}_{1/2+\gamma}(\mathcal{D}_{\mathsf{unif}})$. Since in this distribution, $\mathsf{ct}_0, \mathsf{ct}_1$ are sampled from identical distributions. no algorithm can have a noticeable advantage in distinguishing them. That is, all possible states will be projected onto the result 0, when one applies the projective implemention $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}_{\mathsf{unif}}})$ for any noticeable γ: $\Pr[\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathcal{D}_{\mathsf{unif}}})\rho \to 0] = 1$.

Then we move on to use $\mathsf{ATI}_{1/2+\gamma,\mathcal{P},\mathcal{D}_{\mathsf{unif}}}$ and we have $1 - \mathrm{Tr}[\mathsf{ATI}_{1/2+\gamma-\epsilon,\mathcal{P},\mathcal{D}_{\mathsf{unif}}}\rho] \geq 1 - \delta$ for some exponenially small δ. By Corollary 11.3 we have $1 - \mathrm{Tr}[\mathsf{ATI}_{1/2+\gamma-2\epsilon,\mathcal{P},\mathcal{D}(g_{\ell,j})}\rho] \geq 1 - \eta'_1$ where $\eta'_1 = O(\eta_1^2/\mathsf{poly}(\lambda))$ and η_1 is the statistical distance between $(\mathcal{D}(g_{\ell,j}), \mathcal{D}_{\mathsf{unif}})$.

□

Lemma 11.5 (Invariant Through Measurements)**.** *Suppose we given at the beginning of the algorithm that $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_{2,k}),1/2+\gamma}\,\rho$ **outputs 1** for some inverse polynomial ϵ and exponentially small δ, then we have for all $\ell \in [k], j \in [n]$ and each $g_{\ell,j} \in [-B_x, B_x]$, and let ρ be the state of the distinguisher before the measurement $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_{\ell,j}),1/2+\gamma-4\epsilon}$ in the above algorithm, the following holds:*

- *when the guess $g_{\ell,j} = \mathbf{x}_{\ell,j}$, then there exists some function $\eta'_0(\cdot)$ such that*

$$\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}(g_{\ell,j}),1/2+\gamma-2\epsilon}\,\rho] = 1 - \eta'_0(\lambda)$$

.

- *when the guess $g_{\ell,j} \neq \mathbf{x}_{\ell,j}$, then there exists some function $\eta_1'(\cdot)$ such that*

$$1 - \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},g_{\ell,j},1/2+\gamma-2\epsilon}\,\rho] = 1 - \eta_1'(\lambda)$$

.

Proof. We are given that the distinguisher ρ_{Delete} (let us call it ρ now for brevity) satisfies $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_{2.k},1/2+\gamma}\rho \to 1$, for some exponentially small δ and inverse polynomial ϵ of our own choice (e.g. $\epsilon = \frac{\gamma}{100}$), according to our assumption in Theorem 10.6.

Note that since we have obtained this outcome, we must have already applied once $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_{2.k},1/2+\gamma}$ on ρ and obtained some remaining state ρ'. Now the remaining state ρ' will satisfy the condition that $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_{2.k},1/2+\gamma-3\epsilon}\rho] \geq 1 - 3\delta$ by item 2 and 3 in Lemma 5.7.

Consider the first time we apply $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_{\ell,j}),1/2+\gamma-4\epsilon}$ (when $\ell, j = 1$ and $g_{\ell,j} = -B_x$) in the above algorithm, suppose we have guessed $g_{\ell,j}$ correctly at this point:

By Corollary 11.3, have that $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma-4\epsilon}\,\rho] \geq 1 - 3\delta - \delta' - \eta_0'$, where δ, δ' are exponentially small and $\eta_0' = O(\eta_0) = O(\mathsf{poly}(n,k) \cdot (q^{-n} + B_P/B_{P'}))$, according to Claim 11.1. Since q^{-n} is exponentially small and B_P/B_P is inverse superpolynomial, the η_0' is negligible. We can then apply the gentle measurement lemma Lemma 4.1 and have the post-measurement state recovered to some ρ'' that satisfies $\|\rho' - \rho''\|_{\mathrm{Tr}} \leq \sqrt{\eta_0'}$. Since δ, δ' are exponentially small, we put them inside η_0' from now on.

Suppose the first time we apply $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_{\ell,j}),1/2+\gamma-4\epsilon}$, the guess $g_{\ell,j}$ is incorrect, as shown in Claim 11.4, we have $1 - \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_{\ell,j}),1/2+\gamma-4\epsilon}\,\rho'] \geq 1 - 3\delta - \delta' - \eta_1'$. The magnitude of η_1' is the same as η_0'. Since we obtain outcome 0 with probability $(1 - O(\eta_1'))$, we can therefore also apply the gentle measurement lemma and recover the post-measurement state to trace distance in $O(\sqrt{\eta_1'})$ to the pre-measurement state.

Since in our case, $\eta_0' = \eta_1'$, we use η' to denote both of them from now on.

We can then perform induction: assume the statements hold after the L-th measurement, then the state after the L-th steps in the loop, ρ_L, is $L \cdot O(\sqrt{\eta'})$-close to the distinguisher's state ρ' at the beginning of the algorithm. When the $L+1$-th measurement uses a correct $g_{\ell,j}$: by the fact that $|\mathrm{Tr}(\mathcal{P}\rho) - \mathrm{Tr}(\mathcal{P}\rho_L)| \leq \|\rho' - \rho_L\|_{\mathrm{Tr}}$ for all POVM measurements $\mathcal{P}$, we have $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_{\ell,j}),1/2+\gamma-4\epsilon}\,\rho_L] \geq 1 - (L+1) \cdot O(\sqrt{\eta'})$. Similarly, when the $L+1$-th measurement uses an incorrect $g_{\ell,j}$, we have $1 - \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_{\ell,j}),1/2+\gamma-4\epsilon}\,\rho_L] \geq 1 - (L+1) \cdot O(\sqrt{\eta'})$.

Note that in our case, the total number of loops $L = B_x \cdot nk$. Since we have that $B_{P'}/(B_P \cdot B_x^2)$ is superpolynomial and that $O(\sqrt{\eta'}) = O(\sqrt{B_P/B_{P'}})$, the disturbance on the state ρ_L from origina ρ' stays negligible throughout the entire algorithm.

□

Conclusion By Lemma E.5, we know that for every correct guess $g_{\ell,j} = \mathbf{x}_{\ell,j}$, we will get result 1 after our measurement $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_{\ell,j}),1/2+\gamma-4\epsilon}$ on the current distinguisher's state with probability $1 - \mathsf{negl}(\lambda)$; for every incorrect guess $g_{\ell,j} \neq \mathbf{x}_{\ell,j}$, we will get result 0 after our measurement with probability $1 - \mathsf{negl}(\lambda)$. Therefore, we get to know the value of every $\mathbf{x}_{\ell,j}$ with probability $1 - \mathsf{negl}(\lambda)$ for all $\ell \in [k], j \in [n]$. By the union bound, we can obtain the entire $\{\mathbf{x}_1, \cdots, \mathbf{x}_k\}$ with probability $1 - \mathsf{negl}(\lambda)$.

Running Time By Lemma 5.7, the running time of each $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_{\ell,j})),1/2+\gamma}$ is $T \cdot \mathsf{poly}(1/\epsilon, 1/\log\delta)$ where T is th running of of quantum algorithm ρ, so the entire algorithm is $Tnk \cdot B_x \cdot \mathsf{poly}(1/\epsilon, 1/\log\delta)$, where B_x is quasipolynomial, ϵ is an inverse polynomial and δ can be an inverse exponential function. Since we rely on a subexponential hardness assumption, such running time is acceptable.

12 References

[Aar04] Scott Aaronson. Limitations of quantum advice and one-way communication. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 320–332. IEEE, 2004.

[Aar09] Scott Aaronson. Quantum copy-protection and quantum money. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 229–242. IEEE, 2009.

[Aar18] Scott Aaronson. Shadow tomography of quantum states. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 325–338, 2018.

[AF16] Gorjan Alagic and Bill Fefferman. On quantum obfuscation. *arXiv preprint arXiv:1602.01771*, 2016.

[AK21] Prabhanjan Ananth and Fatih Kaleoglu. Unclonable encryption, revisited. In *Theory of Cryptography Conference*, pages 299–329. Springer, 2021.

[AKL+22] Prabhanjan Ananth, Fatih Kaleoglu, Xingjian Li, Qipeng Liu, and Mark Zhandry. On the feasibility of unclonable encryption, and more. In *Annual International Cryptology Conference*, pages 212–241. Springer, 2022.

[AKN+23] Shweta Agrawal, Fuyuki Kitagawa, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Public key encryption with secure key leasing. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 581–610. Springer, 2023.

[ALL+20] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection, 2020.

[AP21] Prabhanjan Ananth and Rolando L. La Placa. Secure software leasing. EUROCRYPT, 2021.

[APV23] Prabhanjan Ananth, Alexander Poremba, and Vinod Vaikuntanathan. Revocable cryptography from learning with errors. Cryptology ePrint Archive, Paper 2023/325, 2023. `https://eprint.iacr.org/2023/325`.

[BBK22] Nir Bitansky, Zvika Brakerski, and Yael Tauman Kalai. Constructive post-quantum reductions. In *Annual International Cryptology Conference*, pages 654–683. Springer, 2022.

[BCM+21] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. *Journal of the ACM (JACM)*, 68(5):1–47, 2021.

[BGG+23] James Bartusek, Sanjam Garg, Vipul Goyal, Dakshita Khurana, Giulio Malavolta, Justin Raizes, and Bhaskar Roberts. Obfuscation and outsourced computation with certified deletion. *Cryptology ePrint Archive*, 2023.

[BI20] Anne Broadbent and Rabib Islam. Quantum encryption with certified deletion. In *Theory of Cryptography: 18th International Conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings, Part III 18*, pages 92–122. Springer, 2020.

[BJL+21] Anne Broadbent, Stacey Jeffery, Sebastien Lord, Supartha Podder, and Aarthi Sundaram. Secure software leasing without assumptions. In *Theory of Cryptography Conference*, pages 90–120. Springer, 2021.

[BK23] James Bartusek and Dakshita Khurana. Cryptography with certified deletion. In *Annual International Cryptology Conference*, pages 192–223. Springer, 2023.

[BL19] Anne Broadbent and Sébastien Lord. Uncloneable quantum encryption via random oracles. *IACR Cryptology ePrint Archive*, 2019:257, 2019.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011.

[CCY20] Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. In *Theory of Cryptography Conference*, pages 181–206. Springer, 2020.

[ÇGLR23] Alper Çakan, Vipul Goyal, Chen-Da Liu-Zhang, and João Ribeiro. Unbounded leakage-resilience and leakage-detection in a quantum world. *IACR Cryptol. ePrint Arch.*, page 410, 2023.

[CHV23] Céline Chevalier, Paul Hermouet, and Quoc-Huy Vu. Semi-quantum copy-protection and more. *Cryptology ePrint Archive*, 2023.

[CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part I 41*, pages 556–584. Springer, 2021.

[CMP20] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model, 2020.

[CMSZ22] Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. Post-quantum succinct arguments: breaking the quantum rewinding barrier. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 49–58. IEEE, 2022.

[DGTK+10] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *Theory of Cryptography: 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings 7*, pages 361–381. Springer, 2010.

[GC01] Daniel Gottesman and Isaac Chuang. Quantum digital signatures. *arXiv preprint quant-ph/0105032*, 2001.

[Gen10] Craig Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 116–137. Springer, Heidelberg, August 2010.

[GPV07] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. Cryptology ePrint Archive, Report 2007/432, 2007. `https://eprint.iacr.org/2007/432`.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Heidelberg, August 2013.

[HMNY21] Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Quantum encryption with certified deletion, revisited: Public key, attribute-based, and classical communication. In *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part I 27*, pages 606–636. Springer, 2021.

[KNY21] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Secure software leasing from standard assumptions. In *Theory of Cryptography Conference*, pages 31–61. Springer, 2021.

[LLQZ22] Jiahui Liu, Qipeng Liu, Luowen Qian, and Mark Zhandry. Collusion resistant copy-protection for watermarkable functionalities. In *Theory of Cryptography Conference*, pages 294–323. Springer, 2022.

[Mah18] Urmila Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267. IEEE, 2018.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Heidelberg, April 2012.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with small parameters. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 21–39. Springer, Heidelberg, August 2013.

[MW05] Chris Marriott and John Watrous. Quantum arthur–merlin games. *computational complexity*, 14(2):122–152, 2005.

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

[Por22] Alexander Poremba. Quantum proofs of deletion for learning with errors. *arXiv preprint arXiv:2203.01610*, 2022.

[Reg05a] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[Reg05b] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[RS19] Roy Radian and Or Sattath. Semi-quantum money. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, AFT '19, page 132–146. Association for Computing Machinery, 2019.

[Shm22] Omri Shmueli. Public-key quantum money with a classical bank. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 790–803, 2022.

[STHY23] Kyohei Sudo, Masayuki Tezuka, Keisuke Hara, and Yusuke Yoshida. Quantum search-to-decision reduction for the lwe problem. *Cryptology ePrint Archive*, 2023.

[Vai20] Vinod Vaikunthananthan. Cs 294 lecture 4: Worst-case to average-case reductions for lwe. Class at UC Berkeley, 2020. `http://people.csail.mit.edu/vinodv/CS294/lecture6.pdf`.

[Wie83] Stephen Wiesner. Conjugate coding. *ACM Sigact News*, 15(1):78–88, 1983.

[Yue13] Henry Yuen. A quantum lower bound for distinguishing random functions from random permutations. *arXiv preprint arXiv:1310.2885*, 2013.

[Zha12] Mark Zhandry. How to construct quantum random functions. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 679–687. IEEE, 2012.

[Zha20] Mark Zhandry. Schrödinger's pirate: How to trace a quantum decoder. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography*, pages 61–91, Cham, 2020. Springer International Publishing.

A Remarks on Definitions and Strong SKL Security Implies IND-SKL Security

A.1 Additional Remarks for Our Definitions

Remark A.1. *We can additionally add the following verification algorithm on the public key generated, but not necessary for our construction:*

- $\mathsf{VerPK}(\mathsf{mpk}, \mathsf{td}, \mathsf{pk})$: *on input a public key* pk*, a master public key* mpk *and trapdoor* td*, output* Valid *or* $\mathsf{Invalid}$.

This algorithm allows the lessor to check if a public key generated by the lessee is malformed. This will render the key generation protocol as two-message.

However, in our construction, the verification of deletion inherently prevents attacks by generating malicious public keys. For correctness it only has to hold for honest key generations. We thus omit it.

But having such an algorithm in the definition can open the possibility for other SKL constructions with classical communication for future work.

Remark A.2. *We can also have an additional decryption algorithm in which one can use the classical trapdoor* td *to decrypt. We omit it here.*

Remark A.3. *The extension to encrypting multi-bit message follows naturally from a parallel repetition of scheme above. The security is also naturally extended, see [APV23] for discussions.*

A.2 Proof for Claim 8.8

In this section, we prove Claim 8.8.

Claim A.4. *If a construction satisfies Definition 8.7 holds for any noticeable* γ*, then it satisfies Definition 8.2.*

Proof. Suppose there's an adversary $\mathcal{A}$ for Definition 8.2: note that in Definition 8.2, in order for $\mathcal{A}$ to win the game, it must give a valid certificate with some inverse polynomial probability ϵ_1 *and* the following holds:

$$|\Pr\left[\text{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b = 0) = 0\right] - \Pr\left[\text{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b = 1) = 0\right]| \geq \epsilon_2(\lambda)$$

for some inverse polynomial $\epsilon_1(\lambda)$. Note that the above inequality is implicitly conditioned on the fact that $\mathsf{VerDel}(\mathsf{pk}, \mathsf{td}, \mathsf{cert}) = \mathsf{Valid}$. Without loss of generality, we consider the case that $\mathcal{A}$ guesses the correct bit with higher probability. The other case holds symmetrically (by defining the γ-good decryptor to win with probability less than $1/2 - \gamma$).

We can then deduce the following:

$$\begin{aligned}
&\Pr[\text{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b) \to b' : b' = b] \\
&= \frac{1}{2} \cdot \Pr\left[\text{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b = 0) \to 0\right] + \frac{1}{2} \cdot \Pr\left[\text{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b = 1) \to 1\right] \\
&= \frac{1}{2} \cdot \Pr\left[\text{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b = 0) \to 0\right] + \frac{1}{2} \cdot (1 - \Pr\left[\text{IND-PKE-SKL}(\mathcal{A}, 1^\lambda, b = 1) \to 0\right]) \\
&\geq \frac{1}{2} + \frac{1}{2}\epsilon_2(\lambda)
\end{aligned}$$

Therefore, $\mathcal{A}$ (after passting the VerDel-test, should be able to distinguish encryption of μ from a random value in the ciphertext space with probability $1/2 + 1/2 \cdot \epsilon_2$ for some noticeable ϵ_2. Clearly, the adversary $\mathcal{A}$ after handing in the deletion certificate is a $1/2 \cdot \epsilon_2$-good decryptor with noticeable probability. Combining these two conditions above, $\mathcal{A}$ would win in the $1/2 \cdot \epsilon_2$-$\mathsf{StrongSKL}$ game. □

B Secure Key Leasing for FHE Security

B.1 FHE with Secure Key Leasing

To achieve FHE with key leasing, we further define a classical Eval algorithm:

$\mathsf{Eval}(\mathsf{ct}_0, \mathsf{ct}_1) \to \mathsf{ct}'$ on input ciphertexts $(\mathsf{ct}_0, \mathsf{ct}_1)$, output a ciphertext ct'

Remark B.1. *(Levelled FHE with Secure-Key Leasing) We could similar define (levelled) homomorphic encryption supporting secret key leasing analogusly, denoted as FHE-SKL. In such a scheme, the* Setup *algorithm would also take as input a depth parameter d (that's arbitrary polynomial of the security parameter) and the scheme would support homomorphically evaluating ciphertexts of polynomial sized circuits of depth d. The scheme will additionally have an algorithm* Eval *that takes as input a circuit $C : \{0,1\}^{\ell} \to \{0,1\}$ of depth d and polynomial size along with ℓ ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_\ell$ encrypting bits $\mu_1, \ldots, \mu_\ell$. It should produce an evaluated ciphertext $\hat{\mathsf{ct}}$ that encrypts $C(\mu_1, \ldots, \mu_\ell)$. We additionally require the scheme to be compact, namely the size of the ciphertexts and evaluated ciphertexts should be polynomial in λ and d indepdendent of the circuit size. The security requirements for such a scheme remain the same like an PKE-SKL scheme.*

Remark B.2. *(Unlevelled FHE with Secure-Key Leasing) We note that the bootstrapping techinque due to Gentry [Gen10] to lift a levelled FHE to unlevelled FHE assuming circular security does not work for FHE with SKL. Observe that if one released encryption of the secret key, then, in the security experiment the adversary could decrypt the secret-key from such an encryption using the state ρ_{sk} without disturbing it and then use it later win in the security game. Therefore, it seems intuitively impossible to achieve.*

We leave the question of giving some meaningful relaxation in defining unleveled FHE-SKL security for future work.

B.2 FHE Eval Algorithm

To achieve (leveled) fully homomorphic encryption, we add the following algorithm to construction Section 9.2, which ollows from [GSW13]:

- $\mathsf{Eval}(\mathsf{ct}_0, \mathsf{ct}_1)$: the algrithm evaluates an NAND gate on two ciphertexts $(\mathsf{ct}_0, \mathsf{ct}_1)$. It outputs the following:

$$\mathbf{G} - \mathsf{ct}_0 \cdot \mathbf{G}^{-1}(\mathsf{ct}_1)$$

Observe that if $\mathsf{ct}_i = \mathbf{B}\mathbf{R}_i + \mathbf{E}_i + \mu_i \mathbf{G}$ for $i \in \{0,1\}$, the evaluated ciphertext is of the form $\mathbf{B}\mathbf{R} + \mathbf{E} + (1 - \mu_1\mu_2)\mathbf{G}$.

$$\mathbf{R} = -\mathbf{R}_0 \mathbf{G}^{-1}(\mathsf{ct}_1) - \mu_0 \mathbf{R}_1$$
$$\mathbf{E} = -\mathbf{E}_0 \mathbf{G}^{-1}(\mathsf{ct})_1 - \mu_0 \mathbf{E}_1$$

Observe that at each NAND operation the norm of the randomness $\mathbf{R}$ and $\mathbf{E}$ terms could increase by a factor of $((n+1)k+1)\log q$ (the bigger dimension of $\mathbf{B}$) as $\mathbf{G}^{-1}(\mathsf{ct}_1)$ is a binary operation. Assuming subexponential noise to modulus ratio, this means that we can support apriori bounded polynomial depth. In other words, given a depth d, assuming subexponential modulus to ratio LWE holds we can set dimensions and the modulus size to be polynomial in d to ensure a supported depth of d.

C Additional Missing Proofs

C.1 Proof for Corollary 10.7

Proof. Let us denote $\mathcal{D}_0$ as the distribution used in ATI of Game 0 and $\mathcal{D}_{2.k}$ as the distribution used in ATI of Game $2.k$. These two distributions are computationally indistinguishable by the security of LWE (see Claim 10.3, Claim 10.4).

By the property of computationally indistinguishable ATI (Corollary 5.9), we have:

$$\Pr\left[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_{2.k},1/2+\gamma-3\epsilon}(\rho_{\mathsf{Delete}}) = 1 | \mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_0,1/2+\gamma}(\rho_{\mathsf{Delete}}) = 1\right] \geq 1 - \mathsf{negl}(\lambda)$$

By Theorem 10.6 we have:

$$\Pr\left[\begin{array}{c}\mathsf{Ext}(\rho_{\mathsf{Delete}}, \mathsf{pk}) \to (\mathbf{x}_1, \cdots \mathbf{x}_k) : \\ \mathbf{x}_i \text{ is the secret in } \mathbf{x}_i\mathbf{A}_i + \mathbf{e}'_i\end{array} | \mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_{2.k},1/2+\gamma}(\rho_{\mathsf{Delete}}) = 1\right] \geq 1 - \mathsf{negl}(\lambda)$$

Since Theorem 10.6 holds for any inverse polynomial γ and $(\gamma - 3\epsilon)$ is also inverse-polynomial ($\epsilon = \gamma/100$), we have in Game 0, for some negligible $\mathsf{negl}(\cdot)$:

$$\Pr\left[\begin{array}{c}\mathsf{Ext}(\rho_{\mathsf{Delete}}, \mathsf{pk}) \to (\mathbf{x}_1, \cdots \mathbf{x}_k) : \\ \mathbf{x}_i \text{ is the secret in } \mathbf{x}_i\mathbf{A}_i + \mathbf{e}'_i\end{array} | \mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}_0,1/2+\gamma}(\rho_{\mathsf{Delete}}) = 1\right] \geq 1 - \mathsf{negl}(\lambda)$$

We can deduce the above from the following probability relations:

Suppose $\Pr[B|A] \geq 1 - \mathsf{negl}$ and $\Pr[C|B] \geq 1 - \mathsf{negl}$ and all events A, B, C happen with noticeable probability, then we have $\Pr[B] \geq \Pr[B \cap A] \geq \Pr[A] - \mathsf{negl}$ and $\Pr[B \cap C] \geq \Pr[B] - \mathsf{negl} \geq \Pr[A] - \mathsf{negl}$. Thus

$$\Pr[A \cap C] \geq \Pr[A \cap B \cap C] = \Pr[A \cap B] - \Pr[A \cap B \cap \overline{C}] \geq \Pr[A] - \Pr[\overline{C} \cap B] - \mathsf{negl} \geq \Pr[A] - \mathsf{negl}$$

. Finally, by conditional probability and the fact that $\Pr[A]$ is noticeable, we have $\Pr[C|A] \geq 1 - \mathsf{negl}$.

□

C.2 Proof for Theorem 5.10

We follow the proof in [Zha20] Theorem 6.5 with some modifications.

The difficulty is that ATI makes queries on a superposition of exponentially-many samples from the respective $\mathcal{D}_b$ distribution, whose indistinguishability does not follow from the indistinguishability of single samples. We tackle with this using the small range distribution technique in [Zha12].

D_0, D_1 are distributions over $\mathcal{R}$, by distance η.

Hybrid 0 : This is the original world using ATI with respect to D_0 we denote the probability that our output is 1 as $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_0,\gamma}]$.

Hybrid 1 : we choose a random permutation Π on $\mathcal{R}$. Let $D_0^{\Pi}(r) = D_0(\Pi(r))$. Since D_0 and D_0^{Π} are identical distributions, the measurements $\mathcal{P}_{D_0}$ and $\mathcal{P}_{C^{\Pi}}$ are identical, and therefore so are their projective/threshold implementations. Thus, $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_0,\gamma}]$ and $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_0^{\Pi},\gamma}]$ are identically distributed, for any $\gamma \in [0, 1]$.

Hybrid 2 Now we change Π to be the small-range functions $\Sigma = G \circ F$ of [Zha12], where $F : \mathcal{R} \to [s]$ and $G : [s] \to \mathcal{R}$ are random functions, and s is a parameter we will decide later.

By Theorem 6.6, Theorem 6.7 in [Zha20] (eprint version; originally in [Zha12, Yue13]), we have:

$$\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_0^{\Sigma},\gamma}] \geq \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_0^{\Pi},\gamma}] - O(Q^3/s) - O(Q^3/|\mathcal{R}|)$$

By taking $\mathcal{R}$ to be exponentially large and Q is the number of queries the efficient ATI algorithm will make to the function Σ, Π and thus polynomial (in our case, $Q = \mathsf{poly}(1/\epsilon, 1/\log\delta) \cdot T$ for some polynomial T).

Hybrid 3 This is the same as Hybrid 3, except that we change the random function F to be a $2Q$-wise independent function E. By Theorem 6.8 in [Zha20](eprint version, originally [Zha12]), we have that the distributions in Hybrid 2 and 3 are identical.

Hybrid 4 Now we use the distribution D_0, with everything else staying the same. That is, we use distribution $D_1^{G \circ E}$ on any input r is $D_1(G(E(r)))$. We can view $D_b(G(\cdot))$ as a list of s samples from D_b, which the input r is mapped to an index $i \in [s]$ and i selects which sample to use. Since $\mathcal{D}_0$ and $\mathcal{D}_1$ are statistically indistinguishable by distance η, which requires $\Omega(1/\eta^2)$ samples to distinguish. Then setting $s = 1/\eta$ samples will not help distinguish them. Therefore by combining hybrid 0-4 we have: $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_0,\gamma}] \geq \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D_1^{G\circ E},\gamma}] - O(Q^3 \cdot \eta) - O(Q^3/|\mathcal{R}|)$

Hybrid 5-8 These hybrids will be the reverse of hybrid 0-4, with distribution $\mathcal{D}_1$, so we simply multiply the error terms by 2 and conclude the statement.

C.3 Proof for the Probability Relations in Section 10.3

Claim C.1. *Suppose there are three events A, B, C which all happen with inverse polynomial probability, and suppose $\Pr[A|B] \geq 1 - \mathsf{negl}(\lambda)$ and $\Pr[B \cap C] \geq 1/p$ for some polynomial p, we will have $\Pr[B \cap C] \geq 1/p - \mathsf{negl}'(\lambda)$ for some negligible $\mathsf{negl}'(\cdot)$.*

Proof. Since $\Pr[A|B] \geq 1 - \mathsf{negl}(\lambda)$, we have $\Pr[A \cap B] = \Pr[B]\Pr[A|B] \geq \Pr[B] - \mathsf{negl}'(\lambda)$ for some negligible $\mathsf{negl}'(\cdot)$. Therefore, we have $\Pr[B \cap \overline{A}] \leq \mathsf{negl}_1(\lambda)$. Then we can deduce:

$$\begin{aligned} \Pr[A \cap B \cap C] &= \Pr[B \cap C] - \Pr[B \cap C \cap \overline{A}] \\ &\geq 1/p - \Pr[B \cap \overline{A}] = 1/p - \mathsf{negl}'(\lambda) \end{aligned}$$

□

D Secure Key Leasing with Classical Lessor: Protocol

Our one-message key generation protocol and key revocation/deletion protocol come naturally with our security definition Section 8 and construction in Section 9.2.

- **Key Generation**:

- the lessor runs Setup and sends the classical $\mathsf{mpk} = \{\mathbf{A}_i, \mathbf{s}_i\mathbf{A}_i + \mathbf{e}_i\}_{i\in[k]}$ to the lessee. It keeps the trapdoor td private.
- The lessee runs $\mathsf{KeyGen}(\mathsf{mpk})$ to obtain $\rho_{\mathsf{sk}} = \frac{1}{\sqrt{2^k}} \bigotimes_i^k |0, \mathbf{x}_{i,0}\rangle + |1, \mathbf{x}_{i,1}\rangle$ and $\{\mathbf{y}_i\}_{i\in[k]}$. It publishes the public key as $\mathsf{pk} = (\mathsf{mpk}, \{\mathbf{y}_i\}_{i\in[k]})$.

- **Revocation/Deletion:**
  - The lessor sends a message to the lessee asking it to delete.
  - The lessee runs $\mathsf{Delete}(\rho_{\mathsf{sk}})$ to obtain $\mathsf{cert} = (c_1, \mathbf{d}_1, \cdots, c_k, \mathbf{d}_k)$. It sends to the lessor.
  - Lessor runs $\mathsf{VerDel}(\mathsf{td}, \mathsf{pk}, \mathsf{cert})$ and outputs Valid or Invalid.

E Quantum Search-to-Decision LWE with Almost-Perfect Extraction(Clean Version)

In this section, we provide a means to achieve perfect extraction of the LWE secret in a post-quantum LWE search-to-decision reduction, where the quantum LWE distinguisher may come with an auxiliary quantum state.

The algorithm will be analogous to what we have in Section 11 but the analysis will be simler and parameters will be cleaner, because we will feed the algorithm with actual LWE instances versus random instances.

We nevertheless provide it here in an independent section because we believe that the techniques involved will be of independent interest and the statement here may also be used in a black-box way for future works.

γ-good LWE distinguisher Firstly, we define the notion of an γ-good quantum distinguisher for decisional $\mathsf{LWE}_{n,m,q,\sigma}$ with some auxiliary quantum input.

The definition is analogous to the γ-good decryptor in Definition 8.4, but with LWE sample/uniform sample as challenges.

Definition E.1 (γ-good LWE Distinguisher)**.** *A quantum distinguisher can be described by a quantum circuit U and an auxiliary state ρ. W.l.o.g, we denote such a distinguisher simply as ρ to represent a quantum state that also contains a classical description of the circuit.*

- *For some arbitrary secret $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$*[8]*.*
- *Let $\mathcal{P}_{\mathsf{LWE}} = (P_{\mathsf{LWE}}, I - P_{\mathsf{LWE}})$ be the following mixture of projective measurements (in the sense of Definition 5.5) acting on a quantum state ρ. We denote the following distribution we sample from as $\mathcal{D}(\mathsf{LWE})$.*
  - *Sample $\mathbf{u}_0 = \mathbf{s}\mathbf{A} + \mathbf{e}$, where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$.*
  - *Compute $\mathbf{u}_1 \leftarrow \mathbb{Z}_q^m$, a uniform random vector.*
  - *Sample a uniform $b \leftarrow \{0, 1\}$.*
  - *Run the quantum distinguisher ρ on input $(\mathbf{A}, \mathbf{u}_b)$. Check whether the outcome is b. If so, output 1, otherwise output 0.*

[8] For our reduction, we only need such a weak distinguisher that works with respect to a presampled secret $\mathbf{s}$ and error $\mathbf{e}$.

- *Let $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathsf{LWE}})$ be the threshold implementation of $\mathcal{P}_{\mathsf{LWE}}$ with threshold value $\frac{1}{2}+\gamma$, as defined in Definition 5.3. Run $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{\mathsf{LWE}})$ on ρ, and output the outcome. If the output is 1, we say that the quantum distinguisher is γ-good with respect to the secret $\mathbf{s}$ and error $\mathbf{e}$, otherwise not.*

Theorem E.2. *Let $\lambda \in \mathbb{N}$ be the security parameter. Let $n, m \in \mathbb{N}$ be integers and let q be a prime. If there is a γ-good quantum distinguisher for decisional $\mathsf{LWE}_{n,m,q,\sigma}$ with some auxiliary quantum input running in time T for some noticeable γ, there there exists a quantum algorithm that solves search $\mathsf{LWE}_{n,m,q,\sigma}$ with probability $(1-\mathsf{negl}(\lambda))$, running in time $T' = Tnp \cdot \mathsf{poly}(1/\epsilon, 1/\log\delta)$ where p is the size of the domain for each entry $\mathbf{s}_i$ in the secret $\mathbf{s}$.* [9] *and $\epsilon, \delta \in (0,1)$*[10].

The Search-to-Decision Algorithm On a search LWE challenge $(\mathbf{A}, \mathbf{u} = \mathbf{sA}+\mathbf{e})$ and a decision LWE distinguisher with state ρ which is γ-good. We perform the following algorithm:

- For coordinate $i \in 1, 2, \cdots, n$:
  - For $g_i \in k, k+1, \cdots, k+p-1$:
    1. Let ρ be the current state from the quantum distinguisher.
    2. Perform $\mathsf{ATI}^{\epsilon,\delta}_{,\mathcal{P},\mathcal{D}(g_i),1/2+\gamma}$ on the state ρ in Section 5.0.1 according to the following mixture of measurements $\mathcal{P}_{\mathcal{D}(g_i)}$ (we denote the following distribution we sample from as $\mathcal{D}(g_i)$.):
       * Sample a random $\mathbf{c} \leftarrow \mathbb{Z}_q^{1\times m}$, and let matrix $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$ be a matrix where the i-th row is $\mathbf{c}$ and the rest of rows are $\mathbf{0}$'s.
       * Let $\mathbf{A}' = \mathbf{A} + \mathbf{C}$ and let $\mathbf{u}'_0 = \mathbf{u} + g_i \cdot \mathbf{c}$.
       * Sample another uniformly random vector $\mathbf{u}'_1 \leftarrow \mathbb{Z}_q^{1\times m}$
       * Flip a bit $b \leftarrow \{0,1\}$.
       * Run the quantum distinguisher ρ on input $(\mathbf{A}', \mathbf{u}_b)$. Check whether the outcome is b. If so, output 1, otherwise output 0.
    3. If $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}(g_i),1/2+\gamma}$ outputs 1 (that ρ is an γ-good distinguisher), then set $\mathbf{s}_i = g_i$ and move on to next coodinate: $i := i+1$.
    4. If $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}(g_i),1/2+\gamma}$ outputs 0 (that ρ is not an γ-good distinguisher), then let $g_i := g_i + 1$ and go to step 1.

We make the following claims:

Claim E.3. *For all $i \in [n]$, when the guess g_i is the correct $\mathbf{s}_i$ in a search LWE challenge $(\mathbf{A}, \mathbf{u} = \mathbf{sA}+\mathbf{e})$, then $\Pr[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}\, \rho \to 1] = \Pr[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(\mathsf{LWE}),1/2+\gamma}\, \rho \to 1]$.*

Proof. When $g_i = \mathbf{s}_i$, we have $\mathbf{A}' = \mathbf{A} + \mathbf{C}$ where $\mathbf{C}$ is everywhere 0 except the i-th row being uniformly random $\mathbf{c}$. We also have $\mathbf{u}'_0 = \mathbf{sA} + \mathbf{e} + g_i \cdot \mathbf{c} = [(\mathbf{s}_i(\mathbf{A}_{1,i} + \mathbf{c}_1) + \sum_{j\neq i} \mathbf{s}_j(\mathbf{A}_{1,j} + 0), \cdots, (\mathbf{s}_i(\mathbf{A}_{m,i} + \mathbf{c}_m) + \sum_{j\neq i} \mathbf{s}_j(\mathbf{A}_{m,j} + 0)] + \mathbf{e}^\top = \mathbf{s}^\top\mathbf{A}' + \mathbf{e}$, where $\mathbf{A}_{x,y}$ denotes the entry in x-th column and y-th row.

Therefore, for all $i \in [n]$ and each $g_i \in [k, k+p-1]$, when the guess g_i is the correct $\mathbf{s}_i$ in the search LWE challenge $(\mathbf{A}, \mathbf{u} = \mathbf{sA}+\mathbf{e})$, then the mixture of measurements $(\mathcal{P}_{\mathcal{D}(g_i)}, \mathbf{I} - \mathcal{P}_{\mathcal{D}(g_i)})$ is the

[9] Our $\mathsf{LWE}_{n,m,q,\sigma}$ instance may be over a large modulus q, but we require the secret $\mathbf{s}$ to have entries in some smaller domain $[k, k+p) \in \mathbb{Z}_q$ of size p. Our algorithm will work for polynomial or superpolynomial but subexponential p.

[10] In our algorithm, we will set ϵ to be some inverse polynomial smaller than γ. We set δ to be exponentially small.

same as the measurements $(\mathcal{P}_{\mathsf{LWE}}, \mathbf{I} - \mathcal{P}_{\mathsf{LWE}})$ defined in Definition E.1, which test the distinguisher on LWE instances with respect to the same secret $\mathbf{s}$ and error $\mathbf{e}$. Since they are statistically the same measurements, their ATI will yield the same result. $\square$

Claim E.4. *For all $i \in [n]$, when the guess g_i is incorrect, in the above mixture of projections $(\mathcal{P}_{\mathcal{D}_{g_i}}, \mathbf{I} - \mathcal{P}_{\mathcal{D}_{g_i}})$, for any noticeable γ, and any quantum distinguisher ρ, there exists some negligible function $\mathsf{negl}(\cdot)$ such that $\Pr[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}\,\rho \to 0] = 1 - \mathsf{negl}(\lambda)$.*

Proof. We first consider the perfect projective threshold implementation $\mathsf{TI}_{1/2+\gamma}(\mathcal{D}(g_i))$ when $g_i \neq \mathbf{s}_i$. The vector $\mathbf{u}'_0 = \mathbf{s}\mathbf{A} + \mathbf{e} + g_i \cdot \mathbf{c} = [(g_i \cdot \mathbf{c}_1 + \sum_j \mathbf{s}_j \mathbf{A}_{1,j}, \cdots, g_i \cdot \mathbf{c}_m + \sum_j \mathbf{s}_j \mathbf{A}_{m,j}] + \mathbf{e}^\top$. Since $\mathbf{c}$ is uniformly random, the entire $\mathbf{u}'_0$ now becomes uniformly random and statistically the same as $\mathbf{u}'_1$. Therefore no algorithm can have a noticeble advantage in distinguishing them. That is, all possible states will be projected onto the result 0, when one applies the projective implemention $\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{cD(g_i)})$ for any noticeable γ: $\Pr[\mathsf{TI}_{1/2+\gamma}(\mathcal{P}_{cD(g_i)})\rho \to 0] = 1$.

Accordingly, by Lemma 5.7, there exists ϵ, δ such that $\Pr[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}\,\rho \to 0] = 1 - \delta(\lambda) \geq 1 - \mathsf{negl}(\lambda)$ since $\delta(\cdot)$ can be exponentially small. $\square$

Lemma E.5 (Invariant Through Measurements)**.** *For all $i \in [n]$ and each $g_i \in [k, k+p-1]$, and let ρ be the state of the distinguisher before the measurement $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}$ in the above algorithm, the following holds:*

- *when the guess g_i is the correct $\mathbf{s}_i$ in the search LWE challenge $(\mathbf{A}, \mathbf{u} = \mathbf{s}\mathbf{A} + \mathbf{e})$, then there exists some negligible function $\mathsf{negl}(\cdot)$ such that $\Pr[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}\,\rho \to 1] = 1 - \mathsf{negl}(\lambda)$.*
- *when the guess g_i is an incorrect for $\mathbf{s}_i$ in $(\mathbf{A}, \mathbf{u} = \mathbf{s}\mathbf{A} + \mathbf{e})$, then there exists some negligible function $\mathsf{negl}(\cdot)$ such that $\Pr[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}\,\rho \to 0] = 1 - \mathsf{negl}(\lambda)$.*

Proof. Let us denote the probability of the measurement **outputting 1** on ρ by $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}\,\rho]$. Accordingly $1 - \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}\,\rho] := \Pr[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}\,\rho \to 0]$.

We are given that the quantum LWE distinguisher ρ satisfies $\mathrm{Tr}[\mathsf{TI}_{\mathcal{P},D(\mathsf{LWE}),1/2+\gamma}\,\rho] = 1$ according to the definition for good LWE distinguisher Definition E.1.

Consider the first time we apply $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}$ (when $i = 1$ and $g_i = k$) in the above algorithm, suppose we have guessed g_i correctly at this point:

By Lemma 5.7 and claim E.3, have that $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}\,\rho] \geq 1 - \delta$, where δ is negligible (in fact exponentially small). We can then apply the gentle measurement lemma Lemma 4.1 and have the post-measurement state recovered to some ρ' that satisfies $\|\rho - \rho'\|_{\mathrm{Tr}} \leq \sqrt{\delta}$.

Suppose the first time we apply $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}$, the guess g_i is incorrect, as shown in Claim E.4, we have $1 - \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}\,\rho] \geq 1 - \delta$. We can therefore also apply the gentle measurement lemma and recover the post-measurement state to negligible trace distance to the pre-measurement state.

We can then perform induction: assume the statements hold after the k-th measurement, then the state after the k-th measurement, ρ_k, is negligibly close to the original distinguisher's state ρ. When the $k+1$-th measurement uses a correct g_i: by the fact that $|\mathrm{Tr}(\mathcal{P}\rho) - \mathrm{Tr}(\mathcal{P}\rho_k)| \leq \|\rho - \rho_k\|_{\mathrm{Tr}}$ for all POVM measurements $\mathcal{P}$, we have $\mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}\,\rho_k] \geq 1 - \mathsf{negl}(\lambda)$ for some negligible function $\mathsf{negl}(\cdot)$. Similarly, when the $k+1$-th measurement uses an incorrect g_i, we have

$1 - \mathrm{Tr}[\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}\,\rho_k] \geq 1 - \mathsf{negl}(\lambda)$. If $|p|$ is polynomial, then the algorithm will only go on for polynomially many steps, the loss will clearly stay negligible; if $|p|$ is superpolynomial but subexponential, since the loss $\sqrt{\delta}$ from the measurement $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}$ can be exponentially small, the loss will also stay negligible. □

Conclusion By Lemma E.5, we know that for every correct guess $g_i = \mathbf{s}_i$, we will get result 1 after our measurement $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}$ on the current distinguisher's state with probability $1 - \mathsf{negl}(\lambda)$; for every incorrect guess $g_i \neq \mathbf{s}_i$, we will get result 0 after our measurement with probability $1 - \mathsf{negl}(\lambda)$. Therefore, we get to know the value of every $\mathbf{s}_i$ with probability $1 - \mathsf{negl}(\lambda)$ for all $i \in [n]$. By a union bound, we can obtain the entire $\mathbf{s}$ with probability $1 - \mathsf{negl}(\lambda)$.

Running Time By Lemma 5.7, the running time of each $\mathsf{ATI}^{\epsilon,\delta}_{\mathcal{P},D(g_i),1/2+\gamma}$ is $T \cdot \mathsf{poly}(1/\epsilon, 1/\log\delta)$, so the entire algorithm is $Tnp \cdot \mathsf{poly}(1/\epsilon, 1/\log\delta)$, where ϵ is an inverse polynomial and δ can be an inverse exponential function.