Semi-Quantum Tokenized Signatures

Omri Shmueli[*]

Abstract

Quantum tokenized signature schemes (Ben-David and Sattath, QCrypt 2017) allow a sender to generate and distribute quantum unclonable states which grant their holder a one-time permission to sign in the name of the sender. Such schemes are a strengthening of public-key quantum money schemes, as they imply public-key quantum money where some channels of communication in the system can be made classical.

An even stronger primitive is semi-quantum tokenized signatures, where the sender is classical and can delegate the generation of the token to a (possibly malicious) quantum receiver. Semi-quantum tokenized signature schemes imply a powerful version of public-key quantum money satisfying two key features:

- The bank is classical and the scheme can execute on a completely classical communication network. In addition, the bank is *stateless* and after the creation of a banknote, does not hold any information nor trapdoors except the balance of accounts in the system. Such quantum money scheme solves the main open problem presented by Radian and Sattath (AFT 2019).
- Furthermore, the classical-communication transactions between users in the system are *direct* and do not need to go through the bank. This enables the transactions to be both classical and private.

While fully-quantum tokenized signatures (where the sender is quantum and generates the token by itself) are known based on quantum-secure indistinguishability obfuscation and injective one-way functions, the semi-quantum version is not known under any computational assumption. In this work we construct a semi-quantum tokenized signature scheme based on quantum-secure indistinguishability obfuscation and the sub-exponential hardness of the Learning with Errors problem. In the process, we show new properties of quantum coset states and a new hardness result on indistinguishability obfuscation of classical subspace membership circuits.

[*] Tel Aviv University, `omrishmueli@mail.tau.ac.il`. Supported by ISF grants 18/484 and 19/2137, by Len Blavatnik and the Blavatnik Family Foundation, by the European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482), and by the Clore Israel Foundation.

Contents

1 Introduction

Quantum money schemes are one of the basis pillars in quantum cryptography, allowing a bank to distribute quantum unclonable states in a system of users, who can trade the states as currency. The gold standard of quantum money requires the scheme to be *public-key* [AC12], including two quantum algorithms, Bank and QV, with the following syntax: Bank samples a quantum token $(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}}) \leftarrow \mathsf{Bank}$, where $|\mathsf{qt}\rangle_{\mathsf{pk}}$ is a quantum state and pk is a classical public verification key. pk can be distributed in the user network and the quantum part $|\mathsf{qt}\rangle_{\mathsf{pk}}$ can be sent to some specific user. The copy of $|\mathsf{qt}\rangle_{\mathsf{pk}}$ can then be passed around between users in the system, and be publicly verified with QV using the key pk. The core security guarantee assures that tokens are unclonable by anyone but the bank, or even more tightly, no user can generate two states that both pass the quantum verification $\mathsf{QV}(\cdot, \mathsf{pk})$.

By combining intrinsic properties of quantum information with cryptographic techniques, public-key quantum money holds great promise for the future of information technology. Such quantum cryptographic schemes implement functionalities that are *known to be impossible* in a world where only classical computation exists and also create a basis of techniques towards even more advanced primitives, like quantum lightning [Zha19] and quantum copy-protection of programs [Aar09]. Notably, public-key quantum money gives a solution to the problem of privacy in a currency system, where we want a system that is both, secure (a banknote keeps its value and cannot be counterfeited) and private (transaction's information can be kept only to the two parties involved, in particular, the bank does not have to know).

Unfortunately, by the standard definition, to execute a quantum money scheme we need quantum computation to generate and verify tokens, and quantum communication to transfer tokens between devices[1]. Ideally, however, we would like to minimize the required model, and use quantum computation and only *classical* communication - more precisely, making the communication classical while keeping the key advantages of quantum money (e.g. privacy of transactions) is a central open problem in quantum cryptography. Besides the intriguing theoretical question and the fact that there is a fundamental difference between classical and quantum communication[2], practical differences include (1) the fact that a classical communication network can be based on *information broadcasting* (which uses information cloning to execute), which in particular enables communication between mobile devices, and (2) that transactions based on classical communication has the potential to provide *proof of payment*, as the clonable classical transcript can serve as a proof.

Looking more closely on the classical communication problem, there are three directions of communication in a token system: (1) from the bank to a user, (2) from a user to another user, and (3) from a user to the bank. It is a known fact that the classical communication problem can be partially solved, by getting stronger no-cloning guarantees. Specifically, there are three known levels of no-cloning security for the quantum tokens. These levels enable increased classical communication, as we will later see.

1. **No Cloning:** The most basic security level of a quantum token is unclonability. No cloning says that a quantum polynomial-time malicious receiver Rec^* that obtains a single token $(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$ cannot output two quantum states $|\mathsf{qt}_1\rangle$, $|\mathsf{qt}_2\rangle$, such that both pass the public quantum verification $\mathsf{QV}(\cdot, \mathsf{pk})$.

2. **Classically Certifiable Destruction:** The next, stronger guarantee is classically certifiable destruction (CCD). In this version, along with Bank, QV, there are two additional algorithms; a quantum algorithm $\mathsf{GenCert}$ and a classical algorithm CV. While QV allows to publicly verify quantum tokens as before, $\mathsf{GenCert}$ allows to destroy the quantum token and output crt, a classical

[1] Note that quantum teleportation is a known technique to transfer quantum information using classical communication channels. However, assuming no available quantum channel, physical contact is required to distribute the entangled EPR pairs that are used for teleporting the quantum data.

[2] e.g. classical information is more stable and classical communication is likely to be more efficient, as a consequence of the better algorithmic efficiency and lower rate of classical error correcting codes, compared to their quantum counterparts.

certificate of destruction for it. This certificate can later be verified by the classical verification algorithm CV using the public key pk.

CCD security says that no adversary Rec^* can get a single token $(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$ and output both, a quantum token $|\mathsf{qt}\rangle'$ that passes the verification of $\mathsf{QV}(\cdot, \mathsf{pk})$ and crt a classical certificate for its destruction that passes the classical verification of $\mathsf{CV}(\cdot, \mathsf{pk})$. Note that this guarantee is at least as strong as the previous no-cloning, because as part of the correctness of schemes with CCD, for any quantum token $|\mathsf{qt}\rangle'$ that passes the verification $\mathsf{QV}(\cdot, \mathsf{pk})$, a valid classical certificate of destruction crt that passes $\mathsf{CV}(\cdot, \mathsf{pk})$ can be generated (thus two copies of the quantum token imply one quantum token and one classical certificate of destruction for it).

3. **Tokenized Signing:** The third and strongest known level of no-cloning security is tokenized signing. In such scheme like before we have Bank, QV, $\mathsf{GenCert}$, CV, except that now $\mathsf{GenCert}$ gets not only the quantum token $(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$, but also a bit $b \in \{0, 1\}$. The bit b acts as a target for the destruction process. Specifically, given $(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$ and $b \in \{0, 1\}$, the algorithm generates $\mathsf{crt}_b \leftarrow \mathsf{GenCert}(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}}, b)$, a "certificate of destruction with respect to the bit b". The classical verification algorithm then gets, additionally to the classical certificate crt and the public key pk, a bit b, and verifies that indeed crt is a valid certificate for the bit b.

   The tokenized signatures security guarantee says that no Rec^* can get a single token $(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$ and generate two classical certificates crt_0, crt_1 that pass the classical verification with the two different bits, that is, crt_0 passes for $b = 0$ and crt_1 passes for $b = 1$. This guarantee is at least as strong as the previous CCD. To see this, assume there is an adversary Rec^* that outputs a quantum token $|\mathsf{qt}\rangle'$ that passes quantum verification and a classical receipt crt that passes classical verification. crt passes classical verification which means it passes it for some bit $b \in \{0, 1\}$ - we can find out what the bit b is by executing classical verification on crt with input target 0 and input target 1, and then use $|\mathsf{qt}\rangle'$ to generate a targeted classical certificate of destruction for $\neg b$. In this process we obtain crt_b, $\mathsf{crt}_{\neg b}$. The targeted destruction mechanism allows us to think of $(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$ as a one-time signature token to sign in the name of the bank on a single bit, and in particular, we can think of the certificate generation algorithm as a quantum signing algorithm $\mathsf{crt}_b \leftarrow \mathsf{Sign}(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}}, b)$, hence the name signature tokens.

User-to-bank classical communication from CCD tokens. When we move from standard unclonable tokens to CCD tokens, any user can effectively "send" tokens to the bank, using only classical communication: by destroying the token $\mathsf{crt} \leftarrow \mathsf{GenCert}(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$ and sending the classical crt to the bank, the user proves to the bank that it cannot spend the money of that token anymore in the network, and the bank can reimburse the balance of that user. Still, CCD tokens do not solve any of the other two directions of communication: from the bank to a user, and from one user to another user.

1.1 The Advantages of Quantum Signature Tokens

Having the strongest no-cloning guarantee, the power behind signature tokens emerges when the tokens are used in a sequence: We can take λ i.i.d. signature tokens $(\mathsf{pk}_1, |\mathsf{qt}\rangle_{\mathsf{pk}_1}), (\mathsf{pk}_2, |\mathsf{qt}\rangle_{\mathsf{pk}_2}), \cdots, (\mathsf{pk}_\lambda, |\mathsf{qt}\rangle_{\mathsf{pk}_\lambda})$ as a single "string signature token" unit that can sign on any length-λ string. Along with the sequence of tokens, the bank decides on a token value $x \in \mathbb{N} \cup \{0\}$ (in the context of quantum money, this is how much money the bank assigns to that token), samples a unique (with high probability) identifier which is a random serial number $s \leftarrow \{0, 1\}^\lambda$, and a classical signature $\sigma := \sigma_{(\mathsf{pk}_1, \cdots, \mathsf{pk}_\lambda, x, s)}$ for the entire classical part of the token. The signature token is then

$$\mathsf{pk} = (\mathsf{pk}_1, \cdots, \mathsf{pk}_\lambda, x, s, \sigma), \ |\mathsf{qt}\rangle_{\mathsf{pk}} = \left(|\mathsf{qt}\rangle_{\mathsf{pk}_1}, |\mathsf{qt}\rangle_{\mathsf{pk}_2}, \cdots, |\mathsf{qt}\rangle_{\mathsf{pk}_\lambda}\right) \quad .$$

Note that σ is a signature for the entire sequence together, thus one cannot mix and match signatures of two different strings s_1, s_2 produced from two different tokens, in order to get a signature for a third string s_3. Tokens of value $x = 0$ can be regarded as ”dummy tokens” - we next show how they can be used.

User-to-user classical communication from signature tokens. Like CCD tokens, string signature tokens enable the previous classical communication from user to bank (as they are only a strengthening of CCD tokens), but moreover, they enable an additional direction of classical communication, from one user to another. More elaborately, one user Rec_1 holding a token $(\mathsf{pk}_1, |\mathsf{qt}\rangle_{\mathsf{pk}_1})$ of value x_1, can transfer the value x_1 to another user Rec_2 holding a token $(\mathsf{pk}_2, |\mathsf{qt}\rangle_{\mathsf{pk}_2})$ of value of 0, by using $|\mathsf{qt}\rangle_{\mathsf{pk}_1}$ to sign on s_2, the serial number of the token $(\mathsf{pk}_2, |\mathsf{qt}\rangle_{\mathsf{pk}_2})$. After the produced signature is sent to Rec_2, the token $(\mathsf{pk}_2, |\mathsf{qt}\rangle_{\mathsf{pk}_2})$ can be considered to have the value x_1.

Additionally to enabling user-to-user classical communication, two derived abilities of string signature tokens are as follows:

- **Online token destruction:** When the bank wants a certificate of destruction for any token, it samples a random string $d \leftarrow \{0,1\}^\lambda$ and asks the user to sign on d with the signature token.
- **Token value split:** To split the value x of the token $(\mathsf{pk}_1, |\mathsf{qt}\rangle_{\mathsf{pk}_1})$ between two tokens $(\mathsf{pk}_2, |\mathsf{qt}\rangle_{\mathsf{pk}_2})$, $(\mathsf{pk}_3, |\mathsf{qt}\rangle_{\mathsf{pk}_3})$ into $u_2, u_3 \in \mathbb{N} \cup \{0\}$ such that $u_2 + u_3 = x$ (i.e. the value of $(\mathsf{pk}_2, |\mathsf{qt}\rangle_{\mathsf{pk}_2})$ is added u_2 and the value of $(\mathsf{pk}_3, |\mathsf{qt}\rangle_{\mathsf{pk}_3})$ is added u_3), we can hash the serial numbers s_2, s_3 of the two target tokens along with the partition u_2, u_3 of x to a length-λ string, $H(s_2, s_3, u_2, u_3) = y$ for a collision resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^\lambda$, and then use $(\mathsf{pk}_1, |\mathsf{qt}\rangle_{\mathsf{pk}_1})$ to sign on y. This effectively gives a classical proof for the new values of the tokens $(\mathsf{pk}_2, |\mathsf{qt}\rangle_{\mathsf{pk}_2})$, $(\mathsf{pk}_3, |\mathsf{qt}\rangle_{\mathsf{pk}_3})$.

More advantages of signature tokens for quantum money. Aside from direct classical transactions, we get additional unique characteristics to a public-key quantum money system that is based on string signature tokens: **(1) No token database:** When a user wants to return a token to the bank and get its bank account balance reimbursed (using only classical communication), the user and bank can execute the online destruction mechanism. In contrast, in a quantum money system based on CCD tokens, where the token return mechanism is the user simply generating a classical certificate of destruction by itself and sending it to the bank, the bank needs to maintain a database of all previously-destroyed tokens, so malicious users cannot illegally re-use the mechanism and send the same classical certificate of destruction multiple times, for the same token. **(2) Dynamic payment amounts:** The value split mechanism gives one the ability for granular payment amounts, where a user can dynamically choose the amount it wants to pay (unlike in the CCD-based scheme where the value x of a token is fixed during its creation by the bank). **(3) Provable payments:** When one user sends a direct payment to a second user, by signing on the serial number of a dummy token which the second users holds, this signature on the serial number is also a proof of payment, which we do not have in the CCD tokens setting (without going through the bank). **(4) Private classical payments:** While in a scheme based on tokenized signatures, classical user-to-user transactions are direct and thus private, the bank can still obtain information when the user returns a banknote. The online destruction mechanism enables that when the user returns the signature for d using a token that was worth x, if it wishes to hide the token's information (i.e. all information of that token except its worth) and maintain privacy, it can encrypt the classical signature for d and send the encryption together with a zero-knowledge proof that the content of the encryption is a signature for d, and the token that signed on it has a value of x. This mechanism is still secure for the bank, as with high probability, it will never sample a repeating test string d.

1.2 Semi-Quantum Tokenized Signatures

We know how to construct public-key quantum money with signature tokens based on quantum-secure indistinguishability obfuscation and injective one-way functions, from a combination of the work of

Ben-David and Sattath [BDS16] with the work of Coladangelo, Liu, Liu, and Zhandry [CLLZ21]. While such quantum money scheme can cover two out of three directions of communication classically (i.e. from users to the bank and from users to other users), the direction from the bank to users still needs to be quantum.

A strengthening of public-key quantum money is public-key *semi-quantum* money, where everything is the same as before (i.e. same syntax and hierarchy of no-cloning levels of the tokens), but the bank is a classical algorithm, which in particular makes the interaction from bank to users classical. More precisely, the generation of a token is by an interactive protocol between the classical bank Bank and a possibly malicious, quantum receiver Rec: $(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}}) \leftarrow \langle\mathsf{Bank}, \mathsf{Rec}\rangle_{(\mathsf{OUT}_{\mathsf{Bank}}, \mathsf{OUT}_{\mathsf{Rec}})}$, i.e. the output of the bank is pk (this is the public key which the bank can now distribute), and the output of the receiver is the quantum state $|\mathsf{qt}\rangle_{\mathsf{pk}}$. Similarly to before, no-cloning guarantees (i.e. standard no-cloning, CCD or tokenized signing) apply for the state $|\mathsf{qt}\rangle_{\mathsf{pk}}$, but crucially, these guarantees now need to hold even given the fact the actual generator of the state is a possibly malicious receiver Rec^*. Radian and Sattath [Rad19] introduced the notion of semi-quantum money, showed a construction of private-key semi-quantum money, and left open the question of constructing any form of public-key semi-quantum money.

Shmueli [Shm21] later constructs a public-key semi-quantum money scheme with CCD tokens, based on quantum-secure indistinguishability obfuscation and the sub-exponential quantum hardness of the Learning With Errors problem. This means that based on these computational assumptions, we know how to construct a public-key quantum money scheme that covers two directions of communication classically: from the bank to users (because the scheme is semi-quantum and a user can execute the receiver in the token generation protocol) and from a user to the bank (because the tokens are CCD tokens, and as we have seen earlier, such tokens enable returning tokens to the bank by destroying them and sending the receipt to the bank)[3]. So, looking on what we saw until now,

- Public-key fully-quantum money with signature tokens is missing the classical direction from the bank to users, and,
- Public-key semi-quantum money with CCD tokens is missing the classical direction from one user to another.

It remains an open question to classically cover *all three directions of communication at once*. We don't know how to construct such primitive under any computational assumption.

A construction of public-key semi-quantum money with *signature tokens*, or in short, a semi-quantum tokenized signature scheme, solves the above problem. Such scheme has a classical bank like the scheme from [Shm21], but unlike the previous scheme, it has the stronger no-cloning guarantee of tokenized signing. More formally, Radian and Sattath [Rad19] leave two open problems in their work: The first open problem is to construct what's called a *memory-dependent* public-key semi-quantum money, and the second (and the main) open problem, which subsumes the first one, is to construct a *memoryless* public-key semi-quantum money (both notions are defined in their work). The public-key semi-quantum money with CCD tokens of Shmueli [Shm21] solves the construction of a memory-dependent scheme, while constructing a semi-quantum tokenized signature scheme will resolve the main question of constructing a memoryless scheme.

Our focus in this work is to construct a semi-quantum tokenized signature scheme. On the technical side of things, such scheme will show for the first time that it is possible for a classical computer to securely delegate the generation of quantum states that maintain the tokenized signing property.

[3] A nice property of a semi-quantum CCD tokens scheme is *in-direct* classical-communication transactions from user to user: A user can return a token to the bank, and then the bank can classically send a newly-generated token with the same value to the recipient user of that transaction. Observe, however, that such in-direct transactions are always known by the bank and thus are not private, which is one of the fundamental problems that quantum money is intended to solve.

1.3 Results

We resolve the open question and construct a semi-quantum tokenized signature scheme, based on the existence of indistinguishability obfuscation (iO) for classical circuits secure against quantum polynomial-time attacks, and on that the Learning With Errors [Reg09] problem has sub-exponential indistinguishability against quantum computers, that is, there exists some constant $\delta \in (0,1)$ such that for every quantum polynomial-time algorithm, Decisional LWE cannot be solved with advantage greater than $2^{-\lambda^{\delta}}$, where $\lambda \in \mathbb{N}$ is the security parameter of LWE[4].

Formally, we prove the following main Theorem.

Theorem 1.1. *Assume that Decisional LWE has sub-exponential quantum indistinguishability and that indistinguishability obfuscation for classical circuits exists with security against quantum polynomial time distinguishers. Then, there is a semi-quantum tokenized signature scheme (as in Definition 3.3).*

The remaining of the paper is as follows. In Section 2 we explain the main ideas in our construction. The Preliminaries are given in Section 3. In Section 4 we present our construction of semi-quantum tokenized signatures with correctness proof and proof for security against sabotage. In Section 5 we give the security proof of the scheme against signature counterfeiting.

2 Technical Overview

In this section we explain the main technical ideas in our construction and the structure of the overview is as follows. In Section 2.1 we review the previous works related to our goal of constructing semi-quantum tokenized signatures, and explain why a straightforward extension of these works does not work to obtain our goal. In Section 2.2 we describe our construction and the reasoning behind it, with no security proof. In Section 2.3 we explain how the security of the entire scheme is reduced to proving a new hardness property of indistinguishability obfuscation, which is captured by our main technical Lemma 5.1.

2.1 Semi-quantum CCD Tokens and Fully-quantum Signature Tokens

Starting off based on previous work, there is a single protocol [Shm21] where a classical Bank can delegate to a quantum Rec the generation of quantum unclonable and publicly verifiable tokens - this scheme lets the bank and receiver sample together by interaction $(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}}) \leftarrow \langle \mathsf{Bank}, \mathsf{Rec} \rangle_{(\mathsf{OUT}_{\mathsf{Bank}}, \mathsf{OUT}_{\mathsf{Rec}})}$ a token for the receiver (the public key is the output of the bank, which the bank can then share with anyone, in particular the receiver). More precisely, the tokens in the scheme are CCD tokens. As mentioned in the introduction, the scheme also includes public quantum verification $\left(b \in \{0,1\}, |\mathsf{qt}\rangle'_{\mathsf{pk}}\right) \leftarrow \mathsf{QV}(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$, certificate generation $\mathsf{crt} \leftarrow \mathsf{GenCert}(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$, and public classical verification $\mathsf{CV}(\mathsf{pk}, \mathsf{crt}) \in \{0,1\}$.

Our direction in this overview will be to upgrade the construction to be able to generate not only CCD, but signature tokens. This means to have a signing procedure $\sigma_b \leftarrow \mathsf{Sign}(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}}, b)$ instead of the certificate generation $\mathsf{crt} \leftarrow \mathsf{GenCert}(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$, and the classical verification will become a classical signature verification $\mathsf{CV}(\mathsf{pk}, \sigma_b, b) \in \{0,1\}$. Looking at another previous work [BDS16, CLLZ21] which uses a quantum bank but manages to build the stronger signature tokens, it makes sense to try and combine the techniques of the two works. These two works are even more so inviting to be fused, as it is the case that in both works, the tokens are *coset states* - states of the form $|S\rangle^{x,z} := \sum_{u \in S}(-1)^{\langle z,u \rangle}|x+u\rangle$ for a subspace $S \subseteq \{0,1\}^{\lambda}$ and two strings $x, z \in \{0,1\}^{\lambda}$. Let us recall the high-order bits in the two works, and then examine their possible joining.

Recap: Coset states as fully-quantum signature tokens. The fully-quantum tokenized signature scheme of [BDS16, CLLZ21] is as follows: The bank samples a random $\frac{\lambda}{2}$-dimensional subspace

[4] Note that this assumption is weaker than assuming that Decisional LWE is hard for sub-exponential-time quantum algorithms, which is considered a standard cryptographic assumption.

$S \subseteq \{0,1\}^\lambda$, random strings $x, z \in \{0,1\}^\lambda$ and generates $|\mathsf{qt}\rangle_{\mathsf{pk}} := |S\rangle^{x,z}$ i.e. $\sum_{u \in S}(-1)^{\langle z,u\rangle}|x+u\rangle$. The public verification key of the state is $\mathsf{pk} = \left(\mathsf{O}_{S+x}, \mathsf{O}_{S^\perp+z}\right)$, for $\mathsf{O}_{S+x} \leftarrow \mathsf{iO}(C_{S+x}), \mathsf{O}_{S^\perp+z} \leftarrow \mathsf{iO}(C_{S^\perp+z})$, where iO is a quantum-secure indistinguishability obfuscator for classical circuits and $C_{S+x}, C_{S^\perp+z}$ are circuits that check membership in the corresponding cosets $S+x$, $S^\perp+z$. The entire token $(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$ is sent to the receiver.

Public quantum verification QV of the scheme is the standard procedure to verify a coset state [AC12]: Given input a quantum λ-qubit register QT, (1) Check that the output qubit of $\mathsf{O}_{S+x}(\mathsf{QT})$ is 1, then (2) perform Quantum Fourier Transform (QFT) in base 2 i.e. $H^{\otimes\lambda}$ on QT, then (3) Check that the output qubit of $\mathsf{O}_{S^\perp+z}(\mathsf{QT})$ is 1. It is a known fact in the literature that a successful verification in such procedure projects the state to be exactly $|\mathsf{qt}\rangle_{\mathsf{pk}} = |S\rangle^{x,z}$. Finally, regarding the signing algorithm $\mathsf{Sign}(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}}, b)$, to sign on $b=0$ just measure $|\mathsf{qt}\rangle_{\mathsf{pk}}$, and to sign on $b=1$ measure in the Hadamard basis i.e. perform $H^{\otimes\lambda}$ and then measure. Accordingly, a valid signature for $b=0$ is any string in $S+x$, which can be publicly verified using O_{S+x}, and a valid signature for $b=1$ is any string in $S^\perp+z$, which can be publicly verified using $\mathsf{O}_{S^\perp+z}$.

The main technical part of the works [BDS16, CLLZ21] is to show that it is computationally impossible, given $\left(\left(\mathsf{O}_{S+x}, \mathsf{O}_{S^\perp+z}\right), |S\rangle^{x,z}\right)$, to output both $s \in (S+x)$ and $s^\perp \in \left(S^\perp+z\right)$.

Recap: Coset states as semi-quantum CCD tokens. Moving to the semi-quantum setting, the scheme of [Shm21] includes a 3-message coset state generation protocol, as follows:

1. The classical Bank samples a random $\frac{\lambda}{2}$-dimensional subspace $S \subseteq \{0,1\}^\lambda$ (represented by a matrix $\mathbf{M}_S \in \{0,1\}^{\frac{\lambda}{2}\times\lambda}$), and sends to the receiver $(\mathbf{M}_S^x, \mathsf{ct}_x)$, an encryption of the matrix $\mathbf{M}_S$ under hybrid quantum fully-homomorphic encryption (QFHE)[5].

2. The quantum receiver Rec homomorphically evaluates the circuit C_{ssg}, which is a quantum circuit that gets as input the classical description of a subspace $S \subseteq \{0,1\}^\lambda$ e.g. by a matrix, and generates a uniform superposition over S. Thus, the receiver obtains a quantum, homomorphically evaluated ciphertext,
$$\left(|S\rangle^{x',z'}, \mathsf{ct}_{(x',z')}\right) \leftarrow \mathsf{QHE.Eval}\left((\mathbf{M}_S^x, \mathsf{ct}_x), C_{\mathsf{ssg}}\right) \quad ,$$
and sends to Bank the classical part $\mathsf{ct}_{(x',z')}$.

3. Bank decrypts $(x',z') = \mathsf{QHE.Dec}(\mathsf{ct}_{(x',z')})$ and sends obfuscations $\mathsf{O}_{S+x'} \leftarrow \mathsf{iO}(C_{S+x'})$, $\mathsf{O}_{S^\perp+z'} \leftarrow \mathsf{iO}(C_{S^\perp+z'})$ as the public verification key pk.

The coset state $|S\rangle^{x',z'}$ which the receiver holds is the quantum part $|\mathsf{qt}\rangle_{\mathsf{pk}}$ of the token. Accordingly, public quantum verification QV is identical to that of [BDS16, CLLZ21], the certificate generation $\mathsf{crt} \leftarrow \mathsf{GenCert}(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$ is simply a standard basis measurement and the classical certificate verification is just verifying $\mathsf{CV}(\mathsf{pk}, \mathsf{crt}) := \mathsf{O}_{S+x'}(\mathsf{crt})$.

In the security argument of [Shm21] it is shown that it is computationally impossible to output both, the quantum state $|\mathsf{qt}\rangle'$ that passes the verification $\mathsf{QV}(\mathsf{pk}, \cdot)$ and a certificate of destruction for it i.e. any string $s \in (S+x')$. The work does not claim that the generated coset state maintains the tokenized signing property, in fact, it is not even defined what it means that a token signs on 0 or 1.

Attacking the combined scheme. As we said in the beginning of the overview, we should first try to combine the schemes. Since both schemes have the same token structure (i.e., a coset state) and public key (i.e., obfuscations of the membership functions for the primal and dual cosets), to combine the schemes, all we need to do is to take the token generation protocol of [Shm21] and define a signature for $b=0$ to be any $s \in (S+x')$ and a signature for $b=1$ to be any $s^\perp \in (S^\perp+z')$. To argue that the combined scheme maintains the tokenized signing property, it is required to prove that for any

[5] A hybrid QFHE scheme is one where every encryption of a quantum state $|\psi\rangle$ is of the form $\left(|\psi\rangle^{x,z}, \mathsf{ct}_{(x,z)}\right)$, where $|\psi\rangle^{x,z}$ is a quantum OTP encryption of $|\psi\rangle$ with keys $x, z \in \{0,1\}^\lambda$, and $\mathsf{ct}_{(x,z)}$ is a classical FHE encryption of the keys.

quantum polynomial-time receiver Rec^* that interacts with the classical Bank during the token generation protocol, it is impossible to output $(s, s^{\perp})$.

As it turns out, there is a simple way for an adversary to break the tokenized signing security of the combined protocol. More elaborately, consider the following attacker Rec^* that interacts with Bank in the protocol of [Shm21] (described in the previous paragraph):

1. Rec^* obtains $(\mathbf{M}_S^x, \mathsf{ct}_x)$, the first message from Bank.

2. Rec^* samples a random $r \in \{0,1\}^{\frac{\lambda}{2}}$ and homomorphically evaluates the following *classical* circuit $C_{r,1}$: The circuit $C_{r,1}$ takes as input the matrix $\mathbf{M}_S \in \{0,1\}^{\frac{\lambda}{2} \times \lambda}$ and outputs $s := r^T \cdot \mathbf{M}_S$, a vector in the row span. The receiver gets the ciphertext $(\mathsf{ct}_{x'}, s \oplus x')$.

3. Rec^* samples a random $r^{\perp} \in \{0,1\}^{\frac{\lambda}{2}}$ and homomorphically evaluates the following *classical* circuit $C_{r^{\perp},2}$: The circuit $C_{r^{\perp},2}$ takes as input the matrix $\mathbf{M}_S \in \{0,1\}^{\frac{\lambda}{2} \times \lambda}$, computes a basis for $S^{\perp}$ in the form of a matrix $\mathbf{M}_{S^{\perp}} \in \{0,1\}^{\frac{\lambda}{2} \times \lambda}$ and outputs $s^{\perp} := \left(r^{\perp}\right)^T \cdot \mathbf{M}_{S^{\perp}}$, a vector in the row span. The receiver gets the ciphertext $\left(\mathsf{ct}_{x''}, s^{\perp} \oplus x''\right)$.

Assume that in the QFHE, the classical FHE scheme that encrypts the classical QOTP keys x, z, is a bit encryption scheme (this assumption is in many cases w.l.o.g., because in many QFHE schemes, the classical FHE is a bit-encryption scheme). This means in particular that the ciphertext $\mathsf{ct}_{x',z'}$ which the receiver sends in the second message of the protocol is the concatenation of two ciphertexts, $\mathsf{ct}_{x'}$, $\mathsf{ct}_{z'}$.

Going back to our attack, the malicious receiver Rec^* can send $(\mathsf{ct}_{x'}, \mathsf{ct}_{x''})$ as the second message in the protocol (which was originally $\mathsf{ct}_{x',z'}$) to Bank, which decrypts to get x', x'', and sends the obfuscations accordingly: $\mathsf{O}_{S+x'}$, $\mathsf{O}_{S^{\perp}+x''}$ in the third message of the protocol. Finally, note that the receiver still holds $(s \oplus x') \in (S + x')$ and thus a signature for $b = 0$, and also holds $(s^{\perp} \oplus x'') \in (S^{\perp} + x'')$ and thus a signature for $b = 1$.

2.2 Signing Coset States by Splitting

With accordance to the above attack, if we wish to stay with the classical generation protocol of [Shm21], we need to move to a different signing procedure - this will be our first new technique. Formally, we would like to reduce the task of breaking the security of QFHE, to the task of breaking the security of the tokenized signature scheme. Note that S is a random subspace of dimension $\frac{\lambda}{2}$ and thus takes a tiny fraction of $\frac{2^{\frac{\lambda}{2}}}{2^{\lambda}} = 2^{-\frac{\lambda}{2}}$ inside the set of all length-λ strings $\{0,1\}^{\lambda}$. This means that by the security of the QFHE, it should be computationally hard to get $(\mathbf{M}_S^x, \mathsf{ct}_x)$ the classical QFHE encryption of a basis for S, and find a non-zero vector in S. Thus, what we aim for as a very first step is a *definition* of valid signatures for $b = 0$ and $b = 1$ such that given σ_0, σ_1, two signatures for 0 and 1, it is possible to efficiently derive a vector $s \in (S \setminus \{0\})$.

We suggest the following signature definitions for a bit $b \in \{0,1\}$: At the beginning of the protocol, additionally to choosing S at random, the bank randomly splits S (which has $\frac{\lambda}{2}$ dimensions) into S_0, a $\left(\frac{\lambda}{2} - 1\right)$-dimensional subspace of S, and the coset $S_0 + w$, for $w \in (S \setminus S_0)$. Note that these two parts are exactly two disjoint halves of S. If we define a signature for b to be any string in $S_0 + b \cdot w + x'$, then one can verify that the sum of any pair of signatures $\sigma_0 \in (S + x')$, $\sigma_1 \in (S + w + x')$ is a non-zero vector inside S. The above only opens the way for the solution, as we did not yet solve the two main technical parts:

- **Signing:** Given the generated coset state $|S\rangle^{x',z'}$, how can the honest Rec always succeed in signing on b? Simply measuring $|S\rangle^{x',z'}$ will yield the wanted signature only with probability $1/2$.

- **Security:** Given our mechanism for signing (which we did not describe yet), how can we prove security for the new scheme? This part is presented in Section 2.3 of the overview.

Projecting on half the coset with overwhelming probability. We put the security of the scheme aside for the rest of Section 2.2 and focus on proving correctness, that is, explaining how to sign. We show how to transform $|S\rangle^{x',z'}$ into $|S_0 + b \cdot w\rangle^{x',z'}$ given $b \in \{0,1\}$, which will suffice, as a signature can be obtained at that point with probability 1, by measurement. To enable the transformation, the first change in the protocol is that in the third (and last) message of the protocol, where the bank usually sends the public key $\mathsf{pk} := \left(\mathsf{O}_{S+x'}, \mathsf{O}_{S^\perp+z'}\right)$, it now sends an expanded key: $\mathsf{pk}' := \left(\mathsf{O}_{S_0+x'}, \mathsf{O}_{S_0+w+x'}, \mathsf{O}_{S^\perp+z'}\right)$.

Given the state $|S\rangle^{x',z'}$ and $\mathsf{pk}' := \left(\mathsf{O}_{S_0+x'}, \mathsf{O}_{S_0+w+x'}, \mathsf{O}_{S^\perp+z'}\right)$, we explain how to sign on $b = 0$ (the procedure for $b = 1$ is symmetric) by getting the state $|S_0\rangle^{x',z'}$. By measuring the output bit of $\mathsf{O}_{S_0+x'}(|S^{x',z'}\rangle)$, if we succeed (which happens with probability $1/2$) we are done, and if we fail we have $|S_0 + w\rangle^{x',z'}$. It will be enough for the procedure to make the correction and go from the faulty state $|S_0 + w\rangle^{x',z'}$ back to the original state $|S\rangle^{x',z'}$ - since the original state re-enables the experiment of obtaining the correct state $|S_0\rangle^{x',z'}$ with probability $1/2$, we can make λ consecutive iterations of trying to project $|S\rangle^{x',z'}$ to $|S_0\rangle^{x',z'}$ (and correct otherwise), and thus fail with an overall probability of $1 - 2^{-\lambda}$.

Correction of a faulty coset state. The correction procedure from $|S_0 + w\rangle^{x',z'}$ to $|S\rangle^{x',z'}$ is as follows: We start with performing QFT (i.e. $H^{\otimes\lambda}$) on $|S_0 + w\rangle^{x',z'}$ which gives us

$$\sum_{u \in S_0^\perp} (-1)^{\langle x'+w, u\rangle} |z' + u\rangle \ .$$

We can write the above state as

$$\sum_{\left(u \in S_0^\perp\right) \wedge (\langle u,w\rangle = 0)} (-1)^{\langle x'+w,u\rangle} |z' + u\rangle + \sum_{\left(u \in S_0^\perp\right) \wedge (\langle u,w\rangle = 1)} (-1)^{\langle x'+w,u\rangle} |z' + u\rangle$$

$$= \sum_{\left(u \in S_0^\perp\right) \wedge (\langle u,w\rangle = 0)} (-1)^{\langle x',u\rangle} |z' + u\rangle - \sum_{\left(u \in S_0^\perp\right) \wedge (\langle u,w\rangle = 1)} (-1)^{\langle x',u\rangle} |z' + u\rangle \ .$$

Notice that $u \in S^\perp$ if and only if $\left(u \in S_0^\perp\right) \wedge (\langle u, w\rangle = 0)$, also, the set of vectors u' such that $\left(u' \in S_0^\perp\right) \wedge (\langle u', w\rangle = 1)$ is exactly $S^\perp + v$, for any v such that $\left(v \in S_0^\perp\right) \wedge (\langle v, w\rangle = 1)$. We thus write the above sum as

$$\sum_{u \in S^\perp} (-1)^{\langle x',u\rangle} |z' + u\rangle - \sum_{u \in S^\perp} (-1)^{\langle x',u+v\rangle} |z' + u + v\rangle \ .$$

The left sum in the above state is exactly $|S^\perp\rangle^{z',x'}$, which means that if we project the above state with measuring the output bit of $\mathsf{O}_{S^\perp+z'}(\cdot)$ and get 1, we have $|S^\perp\rangle^{z',x'}$ and by executing QFT we go back to $|S\rangle^{x',z'}$, as required.

In case we get 0 then we have $\sum_{u\in S^\perp} (-1)^{\langle x',u+v\rangle} |z' + u + v\rangle$ and we go for the last part of the correction: We can clear the global phase,

$$\sum_{u \in S^\perp} (-1)^{\langle x',u+v\rangle} |z' + u + v\rangle = (-1)^{\langle x',v\rangle} \sum_{u \in S^\perp} (-1)^{\langle x',u\rangle} |z' + u + v\rangle$$

$$\equiv \sum_{u \in S^\perp} (-1)^{\langle x',u\rangle} |z' + u + v\rangle \ ,$$

and execute QFT to get

$$\sum_{u \in S} (-1)^{\langle z'+v,u\rangle} |x' + u\rangle \ .$$

We can write the above state by splitting the sum to S_0 and $S_0 + w$,

$$\sum_{u \in S_0} (-1)^{\langle z'+v,u \rangle} |x' + u\rangle + \sum_{u \in S_0} (-1)^{\langle z'+v,u+w \rangle} |x' + u + w\rangle \ ,$$

and the advantage in that is, because $\left(v \in S_0^{\perp}\right) \wedge (\langle v, w\rangle = 1)$, the above state can be written as

$$\sum_{u \in S_0} (-1)^{\langle z',u \rangle} |x' + u\rangle - \sum_{u \in S_0} (-1)^{\langle z',u+w \rangle} |x' + u + w\rangle$$

$$= |S_0\rangle^{x',z'} - |S_0 + w\rangle^{x',z'} \ .$$

Finally, although we can correct the above state to be $|S\rangle^{x',z'} := |S_0\rangle^{x',z'} + |S_0 + w\rangle^{x',z'}$ (by a phase flip conditioned on the acceptance bit of the circuit $\mathsf{O}_{S_0+w+x'}$), there is no need. This follows because the above state is again a state that enables projecting it on $|S_0\rangle^{x',z'}$ with success probability of $1/2$, and if we fail we get $-|S_0 + w\rangle^{x',z'} \equiv |S_0 + w\rangle^{x',z'}$, which were exactly the properties we needed from $|S\rangle^{x',z'}$.

2.3 Proving CCD Security Versus Proving Tokenized Signing Security

To quickly touch base on where we currently stand, our new generation protocol for signature tokens is the same as the CCD token generation from [Shm21] (which is described in Section 2.1), with two differences:

- The last message from Bank to Rec in the new protocol is $\mathsf{pk}' := \left(\mathsf{O}_{S_0+x'}, \mathsf{O}_{S_0+w+x'}, \mathsf{O}_{S^{\perp}+z'}\right)$ rather than $\mathsf{pk} = \left(\mathsf{O}_{S+x'}, \mathsf{O}_{S^{\perp}+z'}\right)$ from the previous.
- Instead of the certificate generation $\mathsf{crt} \leftarrow \mathsf{GenCert}(\mathsf{pk}, |S\rangle^{x',z'})$ of the previous work which just makes a measurement to the coset state (and does not really use pk), we now have a bit-signing procedure $\sigma_b \leftarrow \mathsf{Sign}(\mathsf{pk}', |S\rangle^{x',z'}, b)$, described in Section 2.2.

Until now we did not cover any of the security aspects of our construction, only the correctness. This following part of the overview, which explains the security argument in high-level, is constructed as follows: We recall the security arguments from previous work [Shm21] that are still relevant for our new construction, until we arrive at the key point of difference between the current work and the previous. Next, we explain why the previous techniques do not cover this difference. Finally, we explain how our main technical Lemma 5.1 covers this gap and enables us to prove that the new scheme produces signature tokens.

Previous techniques and our security argument outline. In our reduction setting, given a malicious Rec^* that breaks the security of the semi-quantum tokenized signature scheme, we construct an adversary $\mathcal{A}_{\textbf{QHE}}$ against the QFHE scheme, in the following manner:

1. $\mathcal{A}_{\textbf{QHE}}$ gets the ciphertext $(\mathbf{M}_S^x, \mathsf{ct}_x)$ as input (for a random S with dimension $\frac{\lambda}{2}$) and passes it directly to Rec^* as the first message of the bank in the protocol.
2. Rec^* returns ct^* as the second message in the protocol.
3. $\mathcal{A}_{\textbf{QHE}}$ computes $(\mathsf{O}_1, \mathsf{O}_2, \mathsf{O}_3)$ as the third message in the protocol and sends to Rec^*.
4. Rec^* outputs two signatures σ_0, σ_1. These signatures are used by $\mathcal{A}_{\textbf{QHE}}$, which outputs the sum $\sigma_0 + \sigma_1$ as an attempt for a non-zero vector in S. The reason why this sum is indeed a non-zero vector in S, at least when the messages of the bank are honestly generated, was explained earlier, in the beginning of Section 2.2.

Note that the third message $(\mathsf{O}_1, \mathsf{O}_2, \mathsf{O}_3)$ of $\mathcal{A}_{\mathbf{QHE}}$ needs to be computationally indistinguishable from $\left(\mathsf{O}_{S_0+x'}, \mathsf{O}_{S_0+w+x'}, \mathsf{O}_{S^\perp+z'}\right)$, the third message in the original protocol. Crucially, in the original protocol, the secret key fhek of the QFHE is used to generate this third message. Specifically, the bank obtains (x', z') by decryption. Having fhek is clearly not possible for the QFHE adversary $\mathcal{A}_{\mathbf{QHE}}$, and the reduction needs to overcome this difficulty.

We prove the reduction by a hybrid argument, and use three previously known tools in the process.

Subspace-hiding obfuscation: We use the well-known subspace-hiding [Zha19] property of indistinguishability obfuscation, which says that (as long as quantum-secure injective one-way functions exist) the obfuscation $\mathsf{O}_{S+x} \leftarrow \mathsf{iO}(C_{S+x})$ is indistinguishable from an obfuscation $\mathsf{O}_{T+x} \leftarrow \mathsf{iO}(C_{T+x})$, for a random superspace $S \subseteq T$ - as long as the dimension of T is not too large[6], even if S is known to the attempting distinguisher (see the formal statement in Lemma 3.1).

Sub-exponential security of QFHE: Another aid we use is the assumption that the QFHE has sub-exponential security[7], which in turn implies that it should not be possible to get a non-zero vector in S with probability greater than $\approx 2^{-\lambda^{\delta'}}$. Note that since we can pick δ the parameter indicating the dimension of the subspaces T_0, T_1 to be any constant, we can take it as a function of δ', in particular, $\delta := \frac{\delta'}{2}$. Such choice of parameters implies $2^{-\lambda^{\delta}} >> 2^{-\lambda^{\delta'}}$.

Blind sampling of obfuscations: As part of the security argument in [Shm21] it is shown that given any fixed pair T_0, T_1 of subspaces with dimension $\lambda - \lambda^{\delta}$ each, even if we do not know x', z', we can successfully sample from a distribution indistinguishable from $(\mathsf{O}_{T_0+x'}, \mathsf{O}_{T_0+w+x'}, \mathsf{O}_{T_1+z'})$ with probability $\approx 2^{-\lambda^{\delta}}$.

Together, the above seemingly paves the way to finish the proof by a hybrid argument:

- Hyb_0 : In the first hybrid $\mathcal{A}_{\mathbf{QHE}}$ acts exactly like the bank and computes the third message $\left(\mathsf{O}_{S_0+x'}, \mathsf{O}_{S_0+w+x'}, \mathsf{O}_{S^\perp+z'}\right)$ using the secret QFHE key fhek. As we know, two valid signatures σ_0, σ_1 in this setting indeed imply that $\sigma_0 + \sigma_1$ is a non-zero vector in S.

- Hyb_1 : In the next hybrid $\mathcal{A}_{\mathbf{QHE}}$ still holds fhek, but sends $(\mathsf{O}_{T_0+x'}, \mathsf{O}_{T_0+w+x'}, \mathsf{O}_{T_1+z'})$ instead. This is indistinguishable from the previous hybrid by the subspace hiding property of the iO. Recall the sub-exponential security of the QFHE where the exponent constant is $\delta' \in (0, 1]$. We take the dimension of the random superspaces $S_0 \subseteq T_0$, $S^\perp \subseteq T_1$ to be both $\lambda - \lambda^{\delta}$, for $\delta := \frac{\delta'}{2}$.

- Hyb_2 : In the next hybrid $\mathcal{A}_{\mathbf{QHE}}$ still holds fhek, but the subspaces T_0, T_1 are fixed by an averaging argument, to be the pair of subspaces that maximize the probability for a successful attack i.e. $\sigma_0 + \sigma_1 \in (S \setminus \{0\})$. Note that S is a random subspace of dimension $\frac{\lambda}{2}$ subjected to $S_0 \subseteq T_0$, $T_1^\perp \subseteq S$. By the sub-exponential security of the QFHE and by the fact that this restriction on S still leaves it enough entropy, it is still computationally impossible to find a non-zero vector in S with probability $>> 2^{-\lambda^{\delta'}}$.

- Hyb_3 : In this experiment $\mathcal{A}_{\mathbf{QHE}}$ does not hold fhek, and given the fixed subspaces T_0, T_1 samples from $(\mathsf{O}_{T_0+x'}, \mathsf{O}_{T_0+w+x'}, \mathsf{O}_{T_1+z'})$ and still succeeds with probability $\approx 2^{-\lambda^{\delta}}$, by blind sampling of the obfuscated circuits.

All hybrids from Hyb_0 to Hyb_2 are indistinguishable, thus in Hyb_2 we still have $\sigma_0 + \sigma_1 \in (S \setminus \{0\})$, but the secret QFHE key fhek is still needed. Hyb_3 then successfully samples from the same output distribution of Hyb_2, without holding fhek and with probability $\approx 2^{-\lambda^{\delta}} >> 2^{-\lambda^{\delta'}}$, which finishes the

[6] For any constant $\delta \in (0, 1]$, the indistinguishability holds for dimension bounded by $\lambda - \lambda^{\delta}$.

[7] The sub-exponential security says that there exists some constant $\delta' \in (0, 1]$ such that it is impossible for any quantum polynomial-time attacker to distinguish encryptions of differing plaintexts with advantage greater than $2^{-\lambda^{\delta'}}$.

proof as with this same probability we get a non-zero vector in S, in contradiction to the sub-exponential security of the QFHE.

Key point of difference - quantumness in the reduction. We inserted one small, but fatal inaccuracy to the above hybrid argument: When we use subspace-hiding techniques to hide S, it becomes no longer correct that getting *any* vector $s \in (S \setminus \{0\})$ is sufficient to break the QFHE security. More precisely, in hybrid Hyb_2 and on, the subspaces T_0, T_1 are fixed and moreover, $T_1^{\perp} \subseteq S$. This makes getting $s \in (S \setminus \{0\})$ not only possible, but trivial: any $s \in (T_1^{\perp} \setminus \{0\})$ will do. In order to break the QFHE we will need $s \in (S \setminus T_1^{\perp})$.

To understand why needing $s \in (S \setminus T_1^{\perp})$ rather than only $s \in (S \setminus \{0\})$ tears apart the above security proof sketch for signature tokens, let us first understand why the above argument actually holds when we want to prove that the tokens in the scheme maintain the weaker, CCD security guarantee. In a nutshell, the key difference is that in the CCD security reduction we are able to use the *quantumness* of the output of the adversary Rec^*.

A successful adversary Rec^* against CCD security manages to output not only two classical strings as signatures, σ_0, σ_1, but one certificate $\mathsf{crt} \in (S + x')$ along with the quantum state $|S\rangle^{x',z'} := \sum_{u \in S}(-1)^{\langle z',u \rangle}|x' + u\rangle$. The use of such output in the reduction is by adding crt to the superposition $|S\rangle^{x',z'}$; this only cancels the x'-pad and gets us $|S\rangle^{0^{\lambda},z'}$. Now, the quantum state $|S\rangle^{0^{\lambda},z'}$ does not give us just an arbitrary non-zero vector in S, but measuring it gives us a *uniform sample* from S. In particular, it is easy to get $s \in (S \setminus T_1^{\perp})$ from such measurement, because the fraction of $T_1^{\perp}$ in S is negligible, which means that with overwhelming probability, the random sample lands outside $T_1^{\perp}$.

Technically, the above hybrid argument fails to prove tokenized signing already in Hyb_1; Even though the hybrids Hyb_0, Hyb_1 are indeed indistinguishable, and even though in both of them we can know S_0, w, x', z' and check whether the output of Rec^* still maintains $\sigma_0 \in (S_0 + x')$, $\sigma_1 \in (S_0 + w + x')$, it can still be the case that $\sigma_0 + \sigma_1 \in T_1^{\perp}$. Then, this fact that $\sigma_0 + \sigma_1 \in T_1^{\perp}$ is dragged for the remaining hybrids, which invalidates the proof - the reduction does not find a vector in $(S \setminus T_1^{\perp})$, and thus QFHE security is unbroken.

Avoiding the dual subspace to prove tokenized signing security. It seems that we need a property of the indistinguishability obfuscator that is of different nature from the subspace-hiding property. We want to claim that given an obfuscation O_{T_1} of a random superspace T_1 of $S^{\perp}$, it is computationally hard to find a vector in the dual subspace $T_1^{\perp}$. Note that such hardness property will finish our proof: We can use it after moving from the above Hyb_0 to Hyb_1, claiming that in Hyb_1, the adversary cannot find vectors in $T_1^{\perp}$. Finally, since the adversary does find vectors in S, we know that the vector in S we found $\sigma_0 + \sigma_1$ is in $\left(S \setminus T_1^{\perp}\right)$. This property can then be carried for the rest of the hybrid experiments, to break the security of the QFHE in the end.

Ideally we indeed would like to prove such strong hardness property, but we do not manage to do so, in fact, it isn't even true that it is always hard: If the dimension of $T_1^{\perp}$, the subspace of S, is big enough (which means that the randomly sampled primal superspace T_1 is not that much bigger than $S^{\perp}$), just by outputting a vector in S, we must be able to land inside $T_1^{\perp}$ with good probability.

What we do manage to show in our main technical Lemma 5.1 is a *dual subspace anti-concentration* property, that says that while it may be possible to hit the dual subspace $T_1^{\perp}$ after getting an obfuscation $\mathsf{O}_{T_1} \leftarrow \mathsf{iO}(C_{T_1})$ (for a random high-dimensional superspace of $S^{\perp}$), it is hard to concentrate there exclusively. In other words, such adversary will always have to make a *near miss into* S, i.e., as long as the adversary manages to hit $T_1^{\perp}$ with a noticeable probability, it has to accidentally hit the background subspace S sometimes, also with a noticeable probability.

3 Preliminaries

We rely on standard notions of classical Turing machines and Boolean circuits:

- A PPT algorithm is a probabilistic polynomial-time Turing machine.
- For a PPT algorithm M, we denote by $M(x;r)$ the output of M on input x and random coins r. For such an algorithm and any input x, we write $m \in M(x)$ to denote the fact that m is in the support of $M(x;\cdot)$.

We follow standard notions from quantum computation.

- A QPT algorithm is a quantum polynomial-time Turing machine.
- An interactive algorithm M, in a two-party setting, has input divided into two registers and output divided into two registers. For the input, one register I_m is for an input message from the other party, and a second register I_a is an auxiliary input that acts as an inner state of the party. For the output, one register O_m is for a message to be sent to the other party, and another register O_a is again for auxiliary output that acts again as an inner state. For a quantum interactive algorithm M, both input and output registers are quantum.

The Adversarial Model. Throughout, efficient adversaries are modeled as quantum circuits with non-uniform quantum advice (i.e. quantum auxiliary input). Formally, *a polynomial-size adversary* $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, consists of a polynomial-size non-uniform sequence of quantum circuits $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, and a sequence of polynomial-size mixed quantum states $\{\rho_\lambda\}_{\lambda\in\mathbb{N}}$.

For an interactive quantum adversary in a classical protocol, it can be assumed without loss of generality that its output message register is always measured in the computational basis at the end of computation. This assumption is indeed without the loss of generality, because whenever a quantum state is sent through a classical channel then qubits decohere and are effectively measured in the computational basis.

Indistinguishability and other Standard Notations.

- For $n \in \mathbb{N}$, define $[n] := \{1, 2, 3, \cdots, n\}$.
- For an n-qubit state $|\psi\rangle$, for classical strings $x, z \in \{0,1\}^n$, the state $|\psi\rangle^{x,z}$ is the Quantum One-Time Pad of $|\psi\rangle$ with pads x, z and is defined to be $\left(\otimes_{i\in[n]} X^{x_i}\right) \cdot \left(\otimes_{i\in[n]} Z^{z_i}\right) \cdot |\psi\rangle$.
- Let $f : \mathbb{N} \to [0,1]$ be a function.
  - f is negligible if for every constant $c \in \mathbb{N}$ there exists $N \in \mathbb{N}$ such that for all $n > N$, $f(n) < n^{-c}$.
  - Accordingly, f is non-negligible if there exists a constant $c \in \mathbb{N}$ such that for infinitely many values of $n \in \mathbb{N}$, $f(n) > n^{-c}$.
  - f is noticeable if there exists $c \in \mathbb{N}, N \in \mathbb{N}$ such that for every $n \geq N$, $f(n) \geq n^{-c}$.
  - f is overwhelming if it is of the form $1 - \mu(n)$, for a negligible function μ.
- For a register of n qubits QT and a classical Boolean function $f : \{0,1\}^n \to \{0,1\}$, the quantum computation $f(\mathsf{QT})$ is computing the classical function f in superposition, that is, applying the unitary transformation $U_f : \forall x \in \{0,1\}^n, b \in \{0,1\}, |x, b\rangle \to |x, b \oplus f(x)\rangle$. The outputs of such computation is a quantum register of $n+1$ qubits: The first n qubits of the output register is the register QT and the last, single-qubit register is denoted by OUT.
- We may consider random variables over bit strings or over quantum states. This will be clear from the context.

- For two random variables X and Y supported on quantum states, quantum distinguisher circuit D with, quantum auxiliary input ρ, and $\mu \in [0,1]$, we write $X \approx_{\mathsf{D},\rho,\mu} Y$ if

$$|\Pr[\mathsf{D}(X;\rho)=1] - \Pr[\mathsf{D}(Y;\rho)=1]| \leq \mu.$$

- Two ensembles of random variables $\mathcal{X} = \{X_i\}_{\lambda \in \mathbb{N}, i \in I_\lambda}$, $\mathcal{Y} = \{Y_i\}_{\lambda \in \mathbb{N}, i \in I_\lambda}$ over the same set of indices $I = \cup_{\lambda \in \mathbb{N}} I_\lambda$ are said to be *computationally indistinguishable*, denoted by $\mathcal{X} \approx_c \mathcal{Y}$, if for every polynomial-size quantum distinguisher $\mathsf{D} = \{\mathsf{D}_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ there exists a negligible function $\mu(\cdot)$ such that for all $\lambda \in \mathbb{N}, i \in I_\lambda$,

$$X_i \approx_{\mathsf{D}_\lambda,\rho_\lambda,\mu(\lambda)} Y_i \ .$$

- The trace distance between two distributions X, Y supported over quantum states, denoted $\mathrm{TD}(X,Y)$, is a generalization of statistical distance to the quantum setting and represents the maximal distinguishing advantage between two distributions supported over quantum states, by unbounded quantum algorithms. We thus say that ensembles $\mathcal{X} = \{X_i\}_{\lambda \in \mathbb{N}, i \in I_\lambda}$, $\mathcal{Y} = \{Y_i\}_{\lambda \in \mathbb{N}, i \in I_\lambda}$, supported over quantum states, are statistically indistinguishable (and write $\mathcal{X} \approx_s \mathcal{Y}$), if there exists a negligible function $\mu(\cdot)$ such that for all $\lambda \in \mathbb{N}, i \in I_\lambda$,

$$\mathrm{TD}\left(X_i, Y_i\right) \leq \mu(\lambda) \ .$$

In what follows, we introduce the cryptographic tools used in this work.

3.1 Indistinguishability Obfuscation

We use indistinguishability obfuscators for classical circuits, that are secure against quantum polynomial-time adversaries.

Definition 3.1. *An indistinguishability obfuscation scheme* iO *is a PPT algorithm that gets as input a security parameter* $\lambda \in \mathbb{N}$ *and a classical circuit* C*, and outputs a classical circuit. It has the following guarantees.*

- ***Correctness:*** *For every classical circuit* C *and security parameter* $\lambda \in \mathbb{N}$*, the programs* $\mathsf{iO}(1^\lambda, C)$ *and* C *are functionally equivalent.*

- ***Indistinguishability:*** *For every polynomial* $\mathrm{poly}(\cdot)$*:*

$$\{\mathsf{iO}(1^\lambda, C_0)\}_{\lambda, C_0, C_1} \approx_c \{\mathsf{iO}(1^\lambda, C_1)\}_{\lambda, C_0, C_1} \ ,$$

where $\lambda \in \mathbb{N}$*,* C_0, C_1 *are two* $\mathrm{poly}(\lambda)$*-size classical circuits with the same functionality.*

In [Zha19], it is shown that indistinguishability obfuscation schemes have the property of *subspace-hiding*. This is proven in Theorem 6.3 in [Zha19]. Lemma 3.1 in [Shm21] extends the parameters in [Zha19] to get the following strengthened statement.

Lemma 3.1 (Lemma 3.1 in [Shm21]). *Let* $S = \{S_\lambda\}_{\lambda \in \mathbb{N}}$ *a subspace* $S \subseteq \{0,1\}^\lambda$ *such that there is a constant* $\delta' \in (0,1]$ *with* $\forall \lambda \in \mathbb{N} : \dim(S_\lambda) \leq \lambda - \lambda^{\delta'}$.

- *Let* iO *an indistinguishability obfuscation scheme, and assume that injective one-way functions exist.*

- *For a subspace* V*, denote by* C_V *a classical circuit that checks membership in* V*.*

Then, for every constants $\delta \in (0, \delta']$, $c \in \mathbb{N}$, we have the following indistinguishability,

$$\left\{\left(\mathsf{O}_{S_\lambda}^{(1)}, \cdots, \mathsf{O}_{S_\lambda}^{(\lambda^c)}\right) \;\middle|\; \left(\mathsf{O}_{S_\lambda}^{(1)}, \cdots, \mathsf{O}_{S_\lambda}^{(\lambda^c)}\right) \leftarrow \mathsf{iO}(C_{S_\lambda})\right\}_{\lambda\in\mathbb{N}}$$

$$\approx_c \left\{\left(\mathsf{O}_{T}^{(1)}, \cdots, \mathsf{O}_{T}^{(\lambda^c)}\right) \;\middle|\; T \leftarrow \mathcal{S}_\lambda^{\subseteq}, \left(\mathsf{O}_{T}^{(1)}, \cdots, \mathsf{O}_{T}^{(\lambda^c)}\right) \leftarrow \mathsf{iO}(C_T)\right\}_{\lambda\in\mathbb{N}} ,$$

where $\mathcal{S}_\lambda^{\subseteq}$ is the uniform distribution over all superspaces of S_λ with dimension $\lambda - \lambda^\delta$.

Instantiations. Indistinguishability Obfuscation for classical circuits that has security against quantum polynomial-time attacks follows from the recent line of works on lattice-inspired iO candidates [BDGM20a, GP21, BDGM20b, DQV+21].

3.2 Leveled Hybrid Quantum Fully Homomorphic Encryption

We rely on quantum fully homomorphic encryption of a specific structure. The formal definition follows.

Definition 3.2 (Leveled Hybrid Quantum Fully-Homomorphic Encryption)**.** *A hybrid leveled quantum fully homomorphic encryption scheme is given by six algorithms* ($\mathsf{QHE.Gen}$, $\mathsf{QHE.Enc}$, $\mathsf{QHE.OTP}$, $\mathsf{QHE.Dec}$, $\mathsf{QHE.QOTP}$, $\mathsf{QHE.Eval}$) *with the following syntax:*

- $\mathsf{fhek} \leftarrow \mathsf{QHE.Gen}(1^\lambda, 1^\ell)$ *: A PPT algorithm that given a security parameter $\lambda \in \mathbb{N}$ and target circuit bound $\ell \in \mathbb{N}$, samples a classical secret key* fhek.
- $m \oplus x = \mathsf{QHE.OTP}_x(m)$ *: A classical polynomial-time deterministic algorithm that takes as input a classical pad $x \in \{0,1\}^*$ and message m such that $|m| = |x|$, and outputs $m \oplus x$.*
- $\mathsf{ct}_b \leftarrow \mathsf{QHE.Enc}_{\mathsf{fhek}}(b)$ *: A PPT algorithm that takes as input a classical bit b and the secret key* fhek *and outputs a classical ciphertext* ct_b. *To encrypt a multi-bit string $x \in \{0,1\}^*$, the algorithm executes on each bit independently.*
- $x = \mathsf{QHE.Dec}_{\mathsf{fhek}}(\mathsf{ct})$ *: A classical polynomial-time deterministic algorithm that takes as input a classical ciphertext* ct *and the secret key* fhek *and outputs a string x.*
- $|\psi\rangle^{x,z} = \mathsf{QHE.QOTP}_{(x,z)}(|\psi\rangle)$ *: A QPT algorithm that takes as input an n-qubit quantum state $|\psi\rangle$ and classical strings as quantum one-time pads $x, z \in \{0,1\}^n$ and outputs the QOTP transformation of the state $|\psi\rangle^{x,z} := \left(\otimes_{i\in[n]} X^{x_i}\right) \cdot \left(\otimes_{i\in[n]} Z^{z_i}\right) \cdot |\psi\rangle$.*
- $\left(|\phi\rangle^{x',z'}, \mathsf{ct}_{(x',z')}\right) \leftarrow \mathsf{QHE.Eval}\left((|\psi\rangle^{x,z}, \mathsf{ct}_{(x,z)}), C\right)$ *: A QPT algorithm that takes as input a general quantum circuit C, a quantum one-time-pad encrypted state $|\psi\rangle^{x,z}$ and a classical ciphertext* $\mathsf{ct}_{(x,z)}$ *of the pads. The evaluation outputs a QOTP encryption of some quantum state $|\phi\rangle$ encrypted under new keys (x', z') and a classical ciphertext* $\mathsf{ct}_{(x',z')}$.

The scheme satisfies the following.

- **Encryption Security:** *For every polynomials $m(\cdot)$, $\ell(\cdot)$, and quantum polynomial-time algorithm $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\})_{\lambda\in\mathbb{N}}$ there exists a negligible function $\mathrm{negl}_{\mathcal{A}}(\cdot)$ such that*

$$\left\{(m_0 \oplus x, \mathsf{ct}_x) \;\middle|\; \begin{array}{l} x \leftarrow \{0,1\}^{m(\lambda)}, \mathsf{fhek} \leftarrow \mathsf{QHE.Gen}(1^\lambda, 1^{\ell(\lambda)}), \\ \mathsf{ct}_x \leftarrow \mathsf{QHE.Enc}_{\mathsf{fhek}}(x), \end{array}\right\}_{\lambda, m_0, m_1}$$

$$\approx_{\mathcal{A}_\lambda, \rho_\lambda, \mathrm{negl}_{\mathcal{A}}(\lambda)}$$

$$\left\{(m_1 \oplus x, \mathsf{ct}_x) \;\middle|\; \begin{array}{l} x \leftarrow \{0,1\}^{m(\lambda)}, \mathsf{fhek} \leftarrow \mathsf{QHE.Gen}(1^\lambda, 1^{\ell(\lambda)}), \\ \mathsf{ct}_x \leftarrow \mathsf{QHE.Enc}_{\mathsf{fhek}}(x), \end{array}\right\}_{\lambda, m_0, m_1} ,$$

where $\lambda \in \mathbb{N}$, $m_0, m_1 \in \{0,1\}^{m(\lambda)}$.

- *If there exists a constant $\delta \in (0,1]$ such that, for every (quantum polynomial-time) adversary $\mathcal{A}$, $\forall \lambda \in \mathbb{N}$, $\mathrm{negl}_{\mathcal{A}}(\lambda) \leq 2^{-\lambda^{\delta}}$, we say that the QFHE scheme has sub-exponential advantage security.*

- **Homomorphism:** *For every polynomial $\ell = \{\ell_\lambda\}_{\lambda \in \mathbb{N}}$ there is a negligible function $\mathrm{negl}(\cdot)$ such that the following holds. Let $\mathsf{fhek} \in \mathsf{QHE.Gen}(1^\lambda, 1^\ell)$, let x, z equal-length strings, let $\mathsf{ct}_{(x,z)} \in \mathsf{QHE.Enc}_{\mathsf{fhek}}(x, z)$, let C a quantum circuit of size $\leq \ell$, let $|\psi\rangle$ a $|x|$-qubit state input for C. Then, $\mathrm{TD}(D_0, D_1) \leq \mathrm{negl}(\lambda)$, where D_0, D_1 are defined as follows.*
  - *D_0 : The output state $|\psi'\rangle \leftarrow C(|\psi\rangle)$.*
  - *D_1 : The state generated by first evaluating $\left(|\phi\rangle^{x',z'}, \mathsf{ct}_{(x',z')}\right) \leftarrow \mathsf{QHE.Eval}\left((|\psi\rangle^{x,z}, \mathsf{ct}_{(x,z)}), C\right)$, and then decrypting $(\tilde{x}, \tilde{z}) = \mathsf{QHE.Dec}_{\mathsf{fhek}}(\mathsf{ct}_{(x',z')})$, $|\phi\rangle = \mathsf{QHE.QOTP}_{(\tilde{x},\tilde{z})}(|\phi\rangle^{x',z'})$.*

Instantiations. Quantum Leveled Fully-Homomorphic encryption with the hybrid structure follows from the work of Mahadev [Mah20], and can be based on the hardness of Learning with Errors. Brakerski [Bra18] shows how to increase the security of QFHE using a weaker LWE assumption. Consequently, constructing QFHE that has hybrid structure, leveled, and has sub-exponential advantage security can be based on assuming Decisional LWE for quantum computers, with sub-exponential indistinguishability.

3.3 Semi-Quantum Tokenized Signatures

In this work we construct a semi-quantum tokenized signature scheme based on cryptographic assumptions. Before describing our construction in Section 4, we give a definition of a semi-quantum tokenized signature scheme. Note that in the below definition, and also in the rest of the technical sections of the paper, we use the general terminology of a sender (instead of a party called the bank) and a receiver. The rest of the names of the algorithms (quantum verification, signature generation and signature verification) stay the same.

Definition 3.3 (Semi-quantum tokenized signature)**.** *A semi-quantum tokenized signature scheme consists of algorithms* (Sen, Rec, QV, Sign, CV) *with the following syntax.*

- $(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}}) \leftarrow \langle \mathsf{Sen}, \mathsf{Rec} \rangle_{(\mathsf{OUT}_{\mathsf{Sen}}, \mathsf{OUT}_{\mathsf{Rec}})}$: *a classical-communication protocol between a PPT algorithm* Sen *and a QPT algorithm* Rec*. At the end of interaction the sender outputs a classical public key* pk *and the receiver outputs a quantum state* $|\mathsf{qt}\rangle_{\mathsf{pk}}$.

- $\left(b, |\mathsf{qt}\rangle'_{\mathsf{pk}}\right) \leftarrow \mathsf{QV}(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}})$ *: A QPT algorithm that gets as input the public key and a candidate token* $|\mathsf{qt}\rangle_{\mathsf{pk}}$ *and outputs a token* $|\mathsf{qt}\rangle'_{\mathsf{pk}}$ *along with a bit $b \in \{0,1\}$.*

- $\sigma_b \leftarrow \mathsf{Sign}(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}}, b)$ *: A QPT algorithm that gets as input the public key* pk*, a candidate token* $|\mathsf{qt}\rangle_{\mathsf{pk}}$ *and a bit $b \in \{0,1\}$ and outputs a classical string σ_b.*

- $\mathsf{CV}(\mathsf{pk}, \sigma_b, b) \in \{0,1\}$ *: A classical polynomial-time deterministic algorithm that takes as input the public key* pk*, a classical string σ_b and a bit $b \in \{0,1\}$, and outputs a bit.*

The scheme satisfies the following guarantees.

- **Statistical Correctness:** *There exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for every $\lambda \in \mathbb{N}$,*

$$\Pr_{(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}}) \leftarrow \langle \mathsf{Sen}, \mathsf{Rec} \rangle_{(\mathsf{OUT}_{\mathsf{Sen}}, \mathsf{OUT}_{\mathsf{Rec}})}} \left[(1, |\mathsf{qt}\rangle'_{\mathsf{pk}}) \leftarrow \mathsf{QV}(\mathsf{pk}, |\mathsf{qt}\rangle_{\mathsf{pk}}) \right] \geq 1 - \mathrm{negl}(\lambda) \ .$$

- **Security:** *For every* $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ *a quantum polynomial-time algorithm there exists a negligible function* $\mathrm{negl}(\cdot)$*, such that for* QT *sampled by interaction with the sender,*

$$(\mathsf{pk}, \mathsf{QT}) \leftarrow \langle \mathsf{Sen}, \mathcal{A} \rangle_{(\mathsf{OUT}_{\mathsf{Sen}}, \mathsf{OUT}_{\mathcal{A}})} \enspace ,$$

for every $\lambda \in \mathbb{N}$*, for each of the below events, the probability for it to occur is* $\leq \mathrm{negl}(\lambda)$*:*

  - **Signature Counterfeiting:** $\mathsf{QT} = (\sigma_0, \sigma_1)$*, such that* $\mathsf{CV}(\mathsf{pk}, \sigma_0, 0) = \mathsf{CV}(\mathsf{pk}, \sigma_1, 1) = 1$.
  - **Quantum Sabotage:** $\mathsf{QT} = |\mathsf{qt}\rangle^{(1)}_{\mathsf{pk}}$ *such that* $(1, |\mathsf{qt}\rangle^{(2)}_{\mathsf{pk}}) \leftarrow \mathsf{QV}(\mathsf{pk}, |\mathsf{qt}\rangle^{(1)}_{\mathsf{pk}})$ *on first execution of* QV*, and then* $(0, |\mathsf{qt}\rangle^{(3)}_{\mathsf{pk}}) \leftarrow \mathsf{QV}(\mathsf{pk}, |\mathsf{qt}\rangle^{(2)}_{\mathsf{pk}})$.
  - **Classical Sabotage:** $\mathsf{QT} = |\mathsf{qt}\rangle^{(1)}_{\mathsf{pk}}$ *such that* $(1, |\mathsf{qt}\rangle^{(2)}_{\mathsf{pk}}) \leftarrow \mathsf{QV}(\mathsf{pk}, |\mathsf{qt}\rangle^{(1)}_{\mathsf{pk}})$ *on first execution of* QV*, and then* $\sigma_b \leftarrow \mathsf{Sign}(\mathsf{pk}, |\mathsf{qt}\rangle^{(2)}, b)$*,* $\mathsf{CV}(\mathsf{pk}, \sigma_b, b) = 0$.

The above definition is relatively succinct compared to the number of protections it guarantees. We go over these derived guarantees here.

Security against sabotage. Security against quantum and classical sabotage protects users in the system i.e. token holders. Security against quantum sabotage basically says that when a user is given a quantum token and it passed the public quantum verification $\mathsf{QV}(\mathsf{pk}, \cdot)$ once, it will pass all further quantum verifications with overwhelming probability. Security against classical sabotage further adds that at the end of this process we can destroy the token to sign on any bit $b \in \{0, 1\}$, $\sigma_b \leftarrow \mathsf{Sign}(\mathsf{pk}, \cdot, b)$. This signature σ_b will pass the public classical verification of $\mathsf{CV}(\mathsf{pk}, \cdot, b)$.

Security against signature counterfeiting is intended to protect the sender. The guarantee says that an adversary cannot output more than a single signature for the single token it got.

Correctness. The formal correctness guarantee says that when the protocol is executed honestly, then the generated token $|\mathsf{qt}\rangle_{\mathsf{pk}}$ passes quantum verification with overwhelming probability. When combined with security against classical sabotage, this means that the token which passed the a quantum verification will successfully generate a classical signature σ_b for any chosen $b \in \{0, 1\}$, that passes the classical verification $\mathsf{CV}(\mathsf{pk}, \cdot, b)$. So, when the protocols are executed honestly the token both passes quantum verification and classical signature generation and verification.

Multi-session Security. As explained in Section 1.1 of the introduction, there is a straightforward transformation to turn single-bit, single-use quantum signature tokens (i.e. the above Definition 3.3) to reusable tokens that can sign on length-λ strings. This transformation is enabled by assuming classical digital signatures with security against quantum polynomial-time attackers.

4 Semi-Quantum Tokenized Signatures Construction

In this section we present our construction of a semi-quantum tokenized signatures (SQTS) scheme, proof of correctness and proof of security against quantum and classical sabotage (all of these are in Definition 3.3).

Ingredients and notation:

- A quantum hybrid fully homomorphic encryption scheme ($\mathsf{QHE.Gen}$, $\mathsf{QHE.Enc}$, $\mathsf{QHE.OTP}$, $\mathsf{QHE.Dec}$, $\mathsf{QHE.QOTP}$, $\mathsf{QHE.Eval}$), with sub-exponential advantage security (Definition 3.2).

- An indistinguishability obfuscation scheme iO (Definition 3.1).

In Figure 1 we describe the token generation protocol and token quantum verification procedures. In Figure 2 we describe the quantum signing algorithm and the classical signature verification procedures.

Protocol 1

Token Generation Protocol: Sen is classical and Rec is quantum. The joint input is the security parameter $\lambda \in \mathbb{N}$.

1. Sen samples a random $\frac{\lambda}{2}$-dimensional subspace $S \subseteq \{0,1\}^{\lambda}$, described by a matrix $\mathbf{M}_S \in \{0,1\}^{\frac{\lambda}{2} \times \lambda}$. Samples OTP key $p_x \leftarrow \{0,1\}^{\frac{\lambda^2}{2}}$ to encrypt $\mathbf{M}_S^{(p_x)} = \mathsf{QHE.OTP}_{p_x}(\mathbf{M}_S)$, and then $\mathsf{fhek} \leftarrow \mathsf{QHE.Gen}(1^{\lambda}, 1^{\ell(\lambda)})$ for some polynomial $\ell(\cdot)$, $\mathsf{ct}_{p_x} \leftarrow \mathsf{QHE.Enc}_{\mathsf{fhek}}(p_x)$. Sen sends the encryption $(\mathbf{M}_S^{(p_x)}, \mathsf{ct}_{p_x})$ to Rec.

2. Let C the quantum circuit that for an input matrix $\mathbf{M} \in \{0,1\}^{\frac{\lambda}{2} \times \lambda}$, outputs a uniform superposition of its row span. The receiver Rec homomorphically evaluates C: $(|S\rangle^{x,z}, \mathsf{ct}_{x,z}) \leftarrow \mathsf{QHE.Eval}\left((\mathbf{M}_S^{(p_x)}, \mathsf{ct}_{p_x}), C\right)$, saves the quantum part $|S\rangle^{x,z}$ and sends the classical part $\mathsf{ct}_{x,z}$ to Sen.

3. Sen decrypts $(x,z) = \mathsf{QHE.Dec}_{\mathsf{fhek}}(\mathsf{ct}_{x,z})$. If $x \in S$, the interaction is terminated. Let $\mathbf{M}_{S^{\perp}} \in \{0,1\}^{\frac{\lambda}{2} \times \lambda}$ a basis for $S^{\perp}$ (as a matrix), let w the first row in $\mathbf{M}_S$ and let $\mathbf{M}_{S_0} \in \{0,1\}^{(\frac{\lambda}{2}-1) \times \lambda}$ the rest of the matrix $\mathbf{M}_S$, without w.

   Sen computes indistinguishability obfuscations $\mathsf{O}_{S_0+x} \leftarrow \mathsf{iO}(\mathbf{M}_{S_0}, x)$, $\mathsf{O}_{S_0+w+x} \leftarrow \mathsf{iO}(\mathbf{M}_{S_0}, w+x)$, $\mathsf{O}_{S^{\perp}+z} \leftarrow \mathsf{iO}(\mathbf{M}_{S^{\perp}}, z)$, all with padding $\mathrm{poly}'(\lambda)$ for some polynomial poly'.

   The output of Sen is $\mathsf{pk} := \left(\mathsf{O}_{S_0+x}, \mathsf{O}_{S_0+w+x}, \mathsf{O}_{S^{\perp}+z}\right)$, the output of Rec is $|\mathsf{qt}\rangle_{\mathsf{pk}} := |S\rangle^{x,z}$.

Quantum Token Verification:

- $\mathsf{QV}\left(\left(\mathsf{O}_{S_0+x}, \mathsf{O}_{S_0+w+x}, \mathsf{O}_{S^{\perp}+z}\right), \mathsf{QT}\right)$: Given a public key and a λ-qubit quantum register QT, the verifier checks two things:
  - Checks that the output qubit of $(\mathsf{O}_{S_0+x} \vee \mathsf{O}_{S_0+w+x})$ (QT) is 1.
  - Executes Hadamard transform $H^{\otimes\lambda}$ on QT and then checks that the output qubit of $\mathsf{O}_{S^{\perp}+z}(\mathsf{QT})$ is 1.

  If both checks passed, the verifier executes $H^{\otimes\lambda}$ again on QT and accepts the signature token.

Figure 1: Token generation protocol between the classical sender and quantum receiver, and quantum token verification procedure of our semi-quantum tokenized signature scheme.

4.1 Correctness and Security Against Sabotage

We first prove that our scheme is correct, which includes two steps: (1) If the scheme's algorithms are ran honestly then the protocol ends successfully, with the output of the honest receiver having negligible trace distance to $|S\rangle^{x,z}$. (2) We recall that $|S\rangle^{x,z}$ passes the quantum verification with probability 1, which overall means that the probability to pass the quantum verification is $1 - \mathrm{negl}(\lambda)$.

Claim 4.1. *If the token generation protocol is executed honestly, the quantum token $|\mathsf{qt}\rangle_{\mathsf{pk}}$ has negligible trace distance from the state $|S\rangle^{x,z} := \sum_{u \in S}(-1)^{\langle z,u\rangle}|x+u\rangle$ (the output of the protocol is defined to be $\perp$ in case the honest sender aborted the interaction), where x, z are the values obtained by the decryption executed by the sender in step 3 of the protocol.*

Protocol 2

Quantum Signing Algorithm:

- $\mathsf{Sign}\left(\left(\mathsf{O}_{S_0+x}, \mathsf{O}_{S_0+w+x}, \mathsf{O}_{S^\perp+z}\right), \mathsf{QT}, b\right)$: Given a public key, a λ-qubit quantum register QT and $b \in \{0,1\}$, the signing algorithm repeats the following procedure λ times and if the loop did not terminate in the middle, it outputs $\perp$.
  1. Measure the output qubit of $\mathsf{O}_{S_0+b\cdot w+x}(\mathsf{QT})$, let $m \in \{0,1\}$ the measurement result.
     (a) If $m = 1$, measure the register QT to get measurement σ_b, output σ_b and terminate.
     (b) If $m = 0$, execute $H^{\otimes\lambda}$ on QT, measure the output qubit of $\mathsf{O}_{S^\perp+z}(\mathsf{QT})$, and execute $H^{\otimes\lambda}$ on QT once again. Restart the loop.

Classical Signature Verification:

- $\mathsf{CV}\left(\left(\mathsf{O}_{S_0+x}, \mathsf{O}_{S_0+w+x}, \mathsf{O}_{S^\perp+z}\right), \sigma_b, b\right)$: To verify a classical signature candidate σ_b for the bit b, the verifier outputs the bit $\mathsf{O}_{S_0+b\cdot w+x}(\sigma_b)$.

Figure 2: The quantum signature algorithm and the classical signature verification procedure of our semi-quantum tokenized signature scheme.

Proof. By the statistical correctness of the QFHE, at the end of step 2 of the generation protocol, the quantum state that the honest Rec holds in its quantum-evaluated register has negligible trace distance to $|S\rangle^{x,z}$, that is, this negligible distance holds with probability 1 over the first two messages of the protocol.

Now, we claim that the probability for such honest Rec to have $x \in S$ is negligible. So, assume towards contradiction it was noticeable. Because the probability for $x \in S$ is noticeable, it has to be the case that with a noticeable probability, when we execute the honest protocol, at the end of step 2 the receiver holds a state with negligible trace distance to $|S\rangle^{x,z}$ for $x \in S$. Now, for any $x \in S$ it follows that $|S\rangle^{x,z} = |S\rangle^{0^\lambda,z}$. This means that by measuring the receiver's state we get a non-zero vector in S with overwhelming probability, and overall, with a noticeable probability we can get a non-zero vector in S without even knowing the QFHE secret key.

Getting a non-zero vector in S violates the security of the QFHE, due to the fact that S is chosen at random and it covers only a negligible fraction out of $\{0,1\}^\lambda$. So, the honest execution of the protocol terminates on with a negligible probability.

Overall, with probability $1-\text{negl}(\lambda)$, we have $x \notin S$, the protocol ends successfully and the receiver holds a quantum state with negligible trace distance to $|S\rangle^{x,z}$. □

We explain how Claim 4.1 implies the statistical correctness of our scheme.

Proposition 4.1. *The scheme presented in Protocol 1 has statistical correctness (Definition 3.3).*

Proof. In Claim 4.1 we saw that with probability $1 - \text{negl}(\lambda)$, the honest receiver Rec holds a quantum state with negligible trace distance to $|S\rangle^{x,z}$.

Finally, our public quantum verification QV is the standard QFT-based verification procedure of a coset state, and a well-known fact in the literature that a successful verification of such procedure is a projection of the verified state onto the subspace spanned only by the coset state [AC12, BDS16]. Because the trace distance of $|\mathsf{qt}\rangle_{\mathsf{pk}}$ from $|S\rangle^{x,z}$ is negligible, the probability for the state to be verified is overwhelming.

Overall, with probability $1-\text{negl}(\lambda)$ over the execution of the honest protocol, the receiver's quantum state passes the quantum verification $\mathsf{QV}(\mathsf{pk}, \cdot)$. □

Security against quantum sabotage. From the fact that the quantum verification $\mathsf{QV}(\mathsf{pk}, \cdot)$ is a projector on the coset state, it follows that after a single successful quantum verification, $|\mathsf{qt}\rangle_{\mathsf{pk}}$ is now $|S\rangle^{x,z}$, which passes the next quantum verification with probability 1.

It remains to prove the security of the scheme against classical sabotage.

Proposition 4.2. *The scheme presented in Protocol 1 has security against classical sabotage (Definition 3.3).*

Proof. The starting point of the algorithm is the state after passing successfully the verification $\mathsf{QV}(\mathsf{pk}, \cdot)$, which, as we stated above, means the state is exactly $|S\rangle^{x,z}$. We now consider what happens to the input state $|S\rangle^{x,z}$ during the execution of the quantum signing algorithm $\mathsf{Sign}(\mathsf{pk}, \cdot, b)$. After the first step 1 of an iteration, if $m = 1$ we are done, as we have $|S_0 + b \cdot w\rangle^{x,z}$ after the measurement, which means that by measuring we get $\sigma_b \in (S_0 + b \cdot w)$ with probability 1. If $m = 0$ we now have $|S_0 + (\neg b) \cdot w\rangle^{x,z}$, which we would like to correct.

Regarding the second step 1b, denote by $m' \in \{0, 1\}$ the measured output bit of $\mathsf{O}_{S^\perp + z}(\mathsf{QT})$, that is, in step 1b of the signing procedure we execute QFT on QT, then measure the output qubit of $\mathsf{O}_{S^\perp + z}(\mathsf{QT})$ (we denoted by m' the outcome of this 1-qubit measurement) and then execute QFT on QT again.

One can verify that if $m' = 1$ then we have $|S^\perp\rangle^{z,x}$ before the second QFT, and thus back to $|S\rangle^{x,z}$ after the second QFT. On the other hand, if $m' = 0$, after the second QFT we have $|S_0\rangle^{x,z} - |S_0 + w\rangle^{x,z}$.

In any case, regardless of the value m', at the end of step 1b of the signing procedure, the state (which is either $|S\rangle^{x,z}$ or $|S_0\rangle^{x,z} - |S_0 + w\rangle^{x,z}$) maintains the property that after measurement of the output qubit of $\mathsf{O}_{S_0 + b \cdot w}(\mathsf{QT})$ (which will come up in upcoming step 1 of the next iteration), it will be projected to be the correct $|S_0 + b \cdot w\rangle^{x,z}$ with probability $1/2$ and with the remaining probability $1/2$ it will be projected to $|S_0 + (\neg b) \cdot w\rangle^{x,z}$.

We deduce that at the beginning of each of the λ iterations we make, when we start with step 1, before the step is executed, we have a state that is projected to $|S_0 + b \cdot w\rangle^{x,z}$ with probability $1/2$ and to $|S_0 + (\neg b) \cdot w\rangle^{x,z}$ with probability $1/2$. The entire process will thus fail only if we fail consecutively λ times, where each experiment is independent from the rest and succeeds with probability $1/2$. Overall, this implies a failure probability of $1 - 2^{-\lambda}$. □

5 Security against Signature Counterfeiting

In this section we will argue that the scheme is secure against signature counterfeiting as in Definition 3.3, that is, under the security of our ingredient primitives, there is no quantum polynomial-time adversary that can get a single signature token (i.e. execute once the protocol with the classical sender, to get a single quantum token for signing) and sign on two different bits 0 and 1.

Proposition 5.1 (Security against Signature Counterfeiting)**.** *Let $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ a quantum polynomial-time adversary that interacts once with the honest classical sender* Sen *in the token generation protocol. Then, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for every $\lambda \in \mathbb{N}$,*

$$\Pr\left[\mathsf{CV}(\mathsf{pk}, \sigma_0, 0) = \mathsf{CV}(\mathsf{pk}, \sigma_1, 1) = 1\right] \le \mathrm{negl}(\lambda) \ ,$$

where the probability is over the random experiment,

$$(\mathsf{pk}, (\sigma_0, \sigma_1)) \leftarrow \langle \mathsf{Sen}, \mathcal{A} \rangle_{(\mathsf{OUT}_{\mathsf{Sen}}, \mathsf{OUT}_{\mathcal{A}})} \ .$$

Proof. Let $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ a quantum polynomial time adversary that succeeds in signing on two different bits with some non-negligible probability $\varepsilon = \{\varepsilon_\lambda\}_{\lambda \in \mathbb{N}}$. We will show how to use $\mathcal{A}$ in order to break the sub-exponential security of the QFHE.

We next describe a sequence of hybrid experiments, consequently arriving to a hybrid experiment that is directly useful for breaking the security of the QFHE.

- Hyb_0 : The original attack.

Defined to be exactly the experiment described above where $\mathcal{A}$ succeeds in signing on two different bits 0 and 1. Specifically, the output of Hyb_0 is the two signatures, σ_0, σ_1. The experiment is defined to be successful iff both signatures are accepted by the signature verification algorithm $\mathsf{CV}(\mathsf{pk}, \cdot, 0/1)$. By definition, the success probability of Hyb_0 is ε.

- Hyb_1 : Changing how we check signatures.

Identical to Hyb_0, only that the success of the experiment is defined to be $(\sigma_0 + \sigma_1) \in (S \setminus \{0^\lambda\})$, rather than before, where we checked $\mathsf{O}_{S+x}(\sigma_0) \wedge \mathsf{O}_{S+x+w}(\sigma_1)$.

By the correctness of the obfuscation scheme iO, it follows from $\mathsf{O}_{S_0+x}(\sigma_0) \wedge \mathsf{O}_{S_0+w+x}(\sigma_1)$ that $\sigma_0 = u_0 + x$ and $\sigma_1 = u_1 + w + x$ for some $u_0, u_1 \in S_0$. This implies exactly that $\sigma_0 + \sigma_1 = (u_0 + u_1) + w$ is inside S by the closure property of S (and by the fact that all three vectors u_0, u_1, w are inside S), but it cannot be zero, because $(u_0 + u_1) \in S_0$ due to closure of S_0, and w is not inside S_0 by definition, which makes the sum $(u_0 + u_1) + w$ non-zero. We get $\sigma_0 + \sigma_1 \in (S \setminus \{0^\lambda\})$ and the success probability of the experiment Hyb_1 is thus at least the success probability of Hyb_0 i.e. it is $\geq \varepsilon$.

- Hyb_2 : Synchronizing subspace membership circuits.

This hybrid is identical to Hyb_1, with the only difference is that all of the obfuscations O_{S_0+x}, O_{S_0+x+w}, $\mathsf{O}_{S^\perp+z}$ that Sen sends to $\mathcal{A}$ at step 3 of the token generation protocol, are changed as follows: The circuit $S_0 + x$ is changed to a circuit that subtracts (mod 2) x from the input and then applies a membership check in S_0, only that the membership check is executed by an obfuscated circuit O_{S_0}. The circuit $S_0 + x + w$ is changed in the analogous way where the subtraction is $x + w$ and the *same* obfuscated membership circuit O_{S_0} for S_0 is used. The circuit $S^\perp + z$ is changed to a circuit that subtracts (mod 2) z and checks membership in $S^\perp$ by the obfuscated circuit $\mathsf{O}_{S^\perp}$.

By the correctness of the obfuscations $\mathsf{iO}(S^\perp)$, $\mathsf{iO}(S_0)$, the functionality of the programs O_{S_0+x}, O_{S_0+x+w}, $\mathsf{O}_{S^\perp+z}$ did not change from Hyb_1 to Hyb_2. It follows that by the security of the indistinguishability obfuscation scheme (applied to the three circuits we are using to check memberships in Hyb_2), the success probability of Hyb_2 is negligibly close to Hyb_1, that is, $\geq \varepsilon - \mathrm{negl}(\lambda)$.

- Hyb_3 : Moving to larger superspaces using the subspace-hiding property of iO.

Let $\delta' \in (0, 1]$ the sub-exponential security level of the QFHE (that is, any quantum polynomial-time algorithm cannot break the security of the QFHE with advantage bigger than $2^{-\lambda^{\delta'}}$), and denote $\delta := \frac{\delta'}{2}$. This hybrid is identical to Hyb_2, with the following changes: When the process samples the *inner* obfuscations O_{S_0} (which is used inside both of the obfuscations O_{S_0+x} and O_{S_0+x+w}) and $\mathsf{O}_{S^\perp}$ (which is used inside the third obfuscation $\mathsf{O}_{S^\perp+z}$), it instead samples a random superspace $S_0 \subseteq T_0 \subseteq \{0,1\}^\lambda$ of dimension $\lambda - \lambda^\delta$, and another random superspace $S^\perp \subseteq T_1 \subseteq \{0,1\}^\lambda$ of dimension $\lambda - \lambda^\delta$, and uses O_{T_0} instead of O_{S_0}, and uses O_{T_1} instead of $\mathsf{O}_{S^\perp}$. As an updated notation in Hyb_3 and on, we can now think of the obfuscations sent by the sender in step 3 of the token generation protocol, as O_{T_0+x}, O_{T_0+x+w}, O_{T_1+z} instead of O_{S_0+x}, O_{S_0+x+w}, $\mathsf{O}_{S^\perp+z}$.

By the subspace hiding property (Lemma 3.1) of indistinguishability obfuscators, the hybrids are indistinguishable and the success probability of Hyb_3 is $\geq \varepsilon - \mathrm{negl}(\lambda)$.

- Hyb_4 : Switching to checking elements outside of $T_1^\perp$.

This hybrid is identical to the previous, with one change to the success definition of the experiment: instead of checking whether $(\sigma_0 + \sigma_1) \in (S \setminus \{0^\lambda\})$, we check whether $(\sigma_0 + \sigma_1) \in (S \setminus T_1^\perp)$.

Now, consider the statement of the anti-concentration Lemma 5.1, where for the subspace S in the Lemma's statement we take our dual subspace $S^{\perp}$, and for the random super-space of the S in the Lemma's statement, we accordingly take our T_1. Because $\varepsilon(\lambda) - \mathrm{negl}(\lambda)$ is a non-negligible function, by Lemma 5.1, it is necessarily the case that there is a non-negligible function $\varepsilon' = \{\varepsilon'_\lambda\}_{\lambda \in \mathbb{N}}$ such that for all $\lambda \in \mathbb{N}$, the probability for $\sigma_0 + \sigma_1 \in (S \setminus T_1^{\perp})$ is at least $\varepsilon'(\lambda)$.

- Hyb_5 : Lowering the dependency from fully knowing x, z, w to knowing only a leakage.

The difference between this process and the previous is that when we send the obfuscations O_{T_0+x}, O_{T_0+x+w}, O_{T_1+z} at step 3 of the generation protocol, the way in which we check membership in each of the cosets is this: Let B_0 a basis for $T_0^{\perp}$, let B_1 a basis for $T_1^{\perp}$, let $B_0 \cdot x = y_x$, $B_0 \cdot w = y_w$, $B_1 \cdot z = y_z$, all strings of length λ^{δ}. O_{T_0+x} is changed to be an obfuscation of a circuit that for input $u \in \{0,1\}^{\lambda}$ checks whether $B_0 \cdot u = y_x$. O_{T_0+x+w} is changed to be an obfuscation of a circuit that for input $u \in \{0,1\}^{\lambda}$ checks whether $B_0 \cdot u = y_x + y_w$. O_{T_1+z} is changed to be an obfuscation of a circuit that for input $u \in \{0,1\}^{\lambda}$ checks whether $B_1 \cdot u = y_z$.

The functionality of the obfuscated circuits O_{T_0+x}, O_{T_0+x+w}, O_{T_1+z} did not change, and thus by the security of the indistinguishability obfuscation schemes, the distributions are indistinguishable and the success probability of Hyb_5 is $\geq \varepsilon' - \mathrm{negl}(\lambda)$.

- Hyb_6 : Changing the way we sample S_0, w, T_0 and $T_1^{\perp}$.

In the remainder of the reduction we would like to move, by a sequence of hybrids, to a final hybrid which allows us to break the security of the QFHE. Specifically, we will consider the subspace S_0 as hidden (more precisely, any vector in $S_0 \setminus T_1^{\perp}$). More specifically, we will get a classical QFHE encryption for a basis for a random S_0, and the information of T_0, $T_1^{\perp}$ and w will be fixed and public. To break the security of the QFHE, we will need to find any vector in $S_0 \setminus T_1^{\perp}$. To fix T_0, $T_1^{\perp}$, w, we will flip the order of sampling all four T_0, $T_1^{\perp}$, w and S_0. The goal of this hybrid is not to flip the order of sampling yet, but to prepare the ground for it.

Observe the following property of the subspace $T_1^{\perp}$ of S: It is either the case that $T_1^{\perp} \subseteq S_0$, or that S_0 contains exactly half of $T_1^{\perp}$. To elaborate, in the tiny-probability case where all λ^{δ} basis vectors of $T_1^{\perp}$ are sampled inside S_0 (for each, this happens with probability $1/2$, because S_0 takes half of S), we get $T_1^{\perp} \subseteq S_0$. This happens with probability exactly $2^{-\lambda^{\delta}}$. In the other case, it is sufficient that at least one of the basis vectors of $T_1^{\perp}$ is in $S \setminus S_0$. One can observe that at this case, the subspace $T_1^{\perp}$ can be written as the set union of two disjoint cosets $\tilde{T}_1^{\perp} \cup \left(\tilde{T}_1^{\perp} + w\right)$, for a random subspace $\tilde{T}_1^{\perp}$ inside S_0 of $\lambda^{\delta} - 1$ dimensions. The reason for the ability to write $T_1^{\perp}$ that way is that we can consider the basis vectors of $T_1^{\perp}$ when we sample them from inside S, then split them to vectors from either S_0 or $S_0 + w$: For every element inside $T_1^{\perp}$, consider its coordinates vector, and more specifically, the coordinates for the elements from $S_0 + w$. For every coordinates vector (recall that the coordinate vectors are binary, as we are dealing with the vectors over the field $\{0,1\}$) with an even number of 1's on the elements from $S_0 + w$, we get a vector inside S_0 (because the summed w's cancel out), and for every coordinates vector with an odd number of 1's on the elements from $S_0 + w$ we get a vector in $S_0 + w$. Since there are exactly half of each case, this splits our subspace $T_1^{\perp}$ into two disjoint cosets, as above.

This means that conditioned on that the event $T_1^{\perp} \subseteq S_0$ (which, as we explained above, happens only with probability $2^{-\lambda^{\delta}}$) does not happen, here is an equivalent way to sample our subspaces and vectors, identical to the previous hybrid: (1) Sample S_0 a random $\left(\frac{\lambda}{2} - 1\right)$-dimensional subspace of $\{0,1\}^{\lambda}$, (2) Sample a random $w \in \left(\{0,1\}^{\lambda} \setminus S_0\right)$ and set $S := S_0 \cup (S_0 + w)$, (3) Sample $\tilde{T}_1^{\perp}$ a random $\left(\lambda^{\delta} - 1\right)$-dimensional subspace of S_0 and set $T_1^{\perp} := \tilde{T}_1^{\perp} \cup \left(\tilde{T}_1^{\perp} + w\right)$, and finally (4) Sample T_0 a random $\left(\lambda - \lambda^{\delta}\right)$-dimensional subspace conditioned on $S \subseteq T_0 \subseteq \{0,1\}^{\lambda}$.

The current Hyb_6 only differs from the previous Hyb_5 for the case $T_1^{\perp} \subseteq S_0$ that happens only in Hyb_5 with probability $2^{-\lambda^{\delta}}$. It follows that the success probability of Hyb_6 is $\geq \varepsilon' - \mathrm{negl}(\lambda) - 2^{-\lambda^{\delta}} = \varepsilon' - \mathrm{negl}(\lambda)$.

- Hyb_7 : Changing the order we sample S_0, w, T_0 and $T_1^{\perp}$.

We are now ready to change the order of sampling our elements. In this hybrid the sampling order is as follows: (1) Sample w a random vector in $\{0,1\}^{\lambda}$. (2) Sample $\tilde{T}_0$ a random $\left(\lambda - \lambda^{\delta} - 1\right)$-dimensional subspace conditioned on $w \notin \tilde{T}_0$, set $T_0 := \tilde{T}_0 \cup \left(\tilde{T}_0 + w\right)$. (3) Sample $\tilde{T}_1^{\perp}$ a random $\left(\lambda^{\delta} - 1\right)$-dimensional subspace of $\tilde{T}_0$, set $T_1^{\perp} := \tilde{T}_1^{\perp} \cup \left(\tilde{T}_1^{\perp} + w\right)$. (4) Sample S_0 a random $\left(\frac{\lambda}{2} - 1\right)$-dimensional subspace of $\tilde{T}_0$. Overall, we sampled w, then $\tilde{T}_0$, then $\tilde{T}_1^{\perp}$, and (importantly) lastly S_0. The distributions over these samples is identical to the previous Hyb_6, and thus in particular the success probability in the current Hyb_7 is $\geq \varepsilon' - \mathrm{negl}(\lambda)$.

- Hyb_8 : Fixing w, $\tilde{T}_0$ and $\tilde{T}_1^{\perp}$.

We can take the sampling procedure of the subspaces described in Hyb_7 and perform an averaging argument on the sampling of w, $\tilde{T}_0$ and $\tilde{T}_1^{\perp}$, to take the three samples that maximize the success probability of Hyb_7. It is straightforward to make this averaging argument at this point, because w, $\tilde{T}_0$ and $\tilde{T}_1^{\perp}$ are sampled before everything else. While this process is clearly not indistinguishable from the previous (as we fix a lot of the entropy in the experiment), because we took the samples w, $\tilde{T}_0$ and $\tilde{T}_1^{\perp}$ for which the previous hybrid is successful with probability $\geq \varepsilon - \mathrm{negl}(\lambda)$, the success of the current Hyb_8 is also $\geq \varepsilon' - \mathrm{negl}(\lambda)$.

- Hyb_9 : Losing the QFHE secret key.

This experiment is identical to Hyb_8 with one change: In the third step 3 of the token generation protocol in Hyb_8, when the sender usually decrypts the QFHE classical part to get the QOTP keys x, z, the current process Hyb_9 does not decrypt to get x, z, and instead it samples uniformly random $y_x', y_w', y_z' \in \{0,1\}^{\lambda^{\delta}}$, and inserts these strings as y_x, y_w, y_z in the obfuscations O_{T_0+x}, O_{T_0+x+w}, O_{T_1+z}, respectively.

Observe that conditioned on the probabilistic event $y_x' = y_x$, $y_w' = y_w$, $y_z' = y_z$ (for which to happen, the probability is exactly $2^{-3\cdot\lambda^{\delta}}$), Hyb_9 and Hyb_8 distribute identically. It follows that the success probability in Hyb_9 is at least $2^{-3\cdot\lambda^{\delta}} \cdot (\varepsilon' - \mathrm{negl}(\lambda)) > 2^{-4\cdot\lambda^{\delta}}$.

- Hyb_{10} : Clearing all given knowledge on S (other than w, $\tilde{T}_0$ and $\tilde{T}_1^{\perp}$).

This hybrid is identical to the previous, with the exception that instead of the honest Sen sending the QFHE encryption $(\mathbf{M}_S^{(p_x)}, \mathsf{ct}_{p_x})$ in step 3 of the token generation protocol, the sender sends an encryption of (a matrix of) zeros $(\mathbf{M}_0^{(p_x)}, \mathsf{ct}_{p_x})$.

Note that in order to execute Hyb_{10} there is no need to know the secret key of the QFHE scheme, so it follows that we can invoke the security of the QFHE. Specifically, we use the sub-exponential-advantage security of the QFHE, so the success probability of Hyb_{10} is $> 2^{-4\cdot\lambda^{\delta}} - 2^{-\lambda^{\delta'}} > 2^{-4\cdot\lambda^{\delta}-1}$.

Wrapping up the proof. We can use the experiment Hyb_{10} to perform a task which is information-theoretically impossible. Specifically, from $S = S_0 \cup (S_0 + w)$ it follows that whenever the experiment Hyb_{10} is successful,

$$(\sigma_0 + \sigma_1) \in \left(S_0 \setminus T_1^{\perp}\right) \ \vee \ (\sigma_0 + \sigma_1) \in \left((S_0 + w) \setminus T_1^{\perp}\right) \ ,$$

which, according to our definition $T_1^{\perp} := \tilde{T}_1^{\perp} \cup \left(\tilde{T}_1^{\perp} + w\right)$, $\tilde{T}_1^{\perp} \subseteq S_0$, is equivalent to

$$(\sigma_0 + \sigma_1) \in \left(S_0 \setminus T_1^{\perp}\right) \ \vee \ (\sigma_0 + \sigma_1) \in \left((S_0 + w) \setminus \left(T_1^{\perp} + w\right)\right) \quad ,$$

which in turn is equivalent to

$$(\sigma_0 + \sigma_1) \in \left(S_0 \setminus \tilde{T}_1^{\perp}\right) \ \vee \ (\sigma_0 + \sigma_1 + w) \in \left(S_0 \setminus \tilde{T}_1^{\perp}\right) \quad .$$

This gives us our contradiction and finishes our proof. More elaborately, we can play the information-theoretic game from Claim 5.1, then execute Hyb_{10} to get $\sigma_0 + \sigma_1$ as the output of the process, and guess (assuming we are in the case where the experiment Hyb_{10} is successful) whether we have $(\sigma_0 + \sigma_1) \in \left(S_0 \setminus \tilde{T}_1^{\perp}\right)$ or $(\sigma_0 + \sigma_1 + w) \in \left(S_0 \setminus \tilde{T}_1^{\perp}\right)$, and output $\sigma_0 + \sigma_1 + b \cdot w$. By the success probability of Hyb_{10}, we can win the information-theoretic game of Claim 5.1 with probability $\geq \frac{1}{2} \cdot 2^{-4 \cdot \lambda^{\delta} - 1}$, in contradiction to Claim 5.1. □

Claim 5.1. *For any two subspaces $\tilde{T}_1^{\perp}$, $\tilde{T}_0$ such that $\tilde{T}_1^{\perp} \subseteq \tilde{T}_0$, $\dim\left(\tilde{T}_1^{\perp}\right) = \lambda^{\delta} - 1$, $\dim\left(\tilde{T}_0\right) = \lambda - \lambda^{\delta} - 1$, assume we sample a random subspace S_0 subject to $\tilde{T}_1^{\perp} \subseteq S_0 \subseteq \tilde{T}_0$, $\dim(S_0) = \frac{\lambda}{2} - 1$. Then for any (possibly unbounded algorithm) the probability to output $s \in (S_0 \setminus \tilde{T}_1^{\perp})$ is bounded by $2^{-\frac{\lambda}{2} + \lambda^{\delta} + 1}$.*

Proof. Let $\mathcal{A}$ any unbounded algorithm. As $\mathcal{A}$ got no input, we can make an averaging argument on the output s of $\mathcal{A}$ that maximizes the probability to guess s that hits the set $(S_0 \setminus \tilde{T}_1^{\perp})$. So, $\mathcal{A}$ always outputs some $s^* \in \{0,1\}^{\lambda}$, independently of the sampled S_0.

If $s^* \notin \left(\tilde{T}_0 \setminus \tilde{T}_1^{\perp}\right)$ then $s^* \notin \left(S_0 \setminus \tilde{T}_1^{\perp}\right)$ and the proof ends as the probability that $\mathcal{A}$ guesses correctly is 0. Since S_0 is a uniformly random $(\frac{\lambda}{2} - 1)$-dimensional subspace subject to $\tilde{T}_1^{\perp} \subseteq S_0 \subseteq \tilde{T}_0$, it follows that for *any* string $s^* \in (\tilde{T}_0 \setminus \tilde{T}_1^{\perp})$, the probability that $s^* \in \left(S_0 \setminus \tilde{T}_1^{\perp}\right)$ is the same, which is,

$$\frac{|S_0 \setminus \tilde{T}_1^{\perp}|}{|\tilde{T}_0 \setminus \tilde{T}_1^{\perp}|} = \frac{2^{\frac{\lambda}{2} - 1} - 2^{\lambda^{\delta} - 1}}{2^{\lambda - \lambda^{\delta} - 1} - 2^{\lambda^{\delta} - 1}} < \frac{2^{\frac{\lambda}{2} - 1}}{2^{\lambda - \lambda^{\delta} - 1} - 2^{\lambda^{\delta} - 1}}$$

$$< \frac{2^{\frac{\lambda}{2} - 1}}{2^{\lambda - \lambda^{\delta} - 1} - 2^{\lambda - \lambda^{\delta} - 2}} = \frac{2^{\frac{\lambda}{2} - 1}}{2^{\lambda - \lambda^{\delta} - 2}} = 2^{-\frac{\lambda}{2} + \lambda^{\delta} + 1} \quad .$$

□

5.1 Hardness of Concentration in the Dual of an Obfuscated Subspace

In this Section we prove the main technical lemma of this work, which intuitively says the following. Assume we have a subspace $S \subseteq \{0,1\}^{\lambda}$ of dimension d, and assume we sample a random subspace $T \subseteq \{0,1\}^{\lambda}$, $S \subseteq T$, such that the dimension of T is sufficiently far away from being the full dimension λ (formally, there needs to be at least some positive power function λ^{δ}, $\delta \in (0,1)$ such that the dimension is bounded by $\lambda - \lambda^{\delta}$). Then, for any quantum polynomial-time algorithm $\mathcal{A}$ that gets an indistinguishability obfuscation of membership check in the random T, it cannot be the case that $\mathcal{A}$, given the obfuscation, can hit $T^{\perp}$ with a noticeable probability but concentrate there, that is, make a ”near miss” only with a negligible probability.

By a near miss, we mean this: Since $S \subseteq T \subsetneq \{0,1\}^{\lambda}$, we have $T^{\perp} \subseteq S^{\perp} \subsetneq \{0,1\}^{\lambda}$, which means that the background of $T^{\perp}$ is $S^{\perp}$. A near miss is not just any miss $v \in \{0,1\}^{\lambda} \setminus T^{\perp}$. A near miss is missing $T^{\perp}$ by a little, and hitting the immediate background $S^{\perp} \setminus T^{\perp}$ and not just $\{0,1\}^{\lambda}$. The formal statement of the above is given in Lemma 5.1 below.

Lemma 5.1 (IO Dual Subspace Anti-Concentration). *Let $S = \{S_\lambda\}_{\lambda\in\mathbb{N}}$ a subspace $S_\lambda \subseteq \{0,1\}^\lambda$ of dimension $d = \{d_\lambda\}_{\lambda\in\mathbb{N}}$. Let $t = \{t_\lambda\}_{\lambda\in\mathbb{N}}$ such that $\forall\lambda \in \mathbb{N} : t_\lambda \leq \lambda$.*

- *Let* iO *a quantum-secure indistinguishability obfuscation scheme for classical circuits and assume that post-quantum injective one-way functions exist.*
- *For a subspace V, denote by $C_V : \{0,1\}^\lambda \to \{0,1\}$ some canonical circuit that checks membership in the subspace V (say, by Gaussian elimination for some basis for the subspace).*
- *Denote by $\mathcal{S}^{\subseteq} = \{\mathcal{S}^{\subseteq}_{\lambda-t_\lambda}\}_{\lambda\in\mathbb{N}}$ the uniform distribution over subspaces of dimension $\lambda - t_\lambda$ that contain S.*

Assume there is a quantum polynomial-time algorithm $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ such that there exists a constant $c \in \mathbb{N}$ and an infinite set $Q \subseteq \mathbb{N}$, such that,

$$\forall\lambda \in Q : \Pr\left[\mathcal{A}_\lambda(\rho_\lambda, \mathsf{O}_T) \in \left(T^\perp \setminus \{0\}\right) \,\middle|\, T \leftarrow \mathcal{S}^{\subseteq}_{\lambda-t}, \mathsf{O}_T \leftarrow \mathsf{iO}(C_T)\right] \geq \frac{1}{\lambda^c} \enspace .$$

Then, if there is some constant $\delta \in (0,1)$ with: $\forall\lambda \in \mathbb{N} : t_\lambda \geq \lambda^\delta,\ \lambda - (d_\lambda + 3 \cdot t_\lambda) \geq 6 + c \cdot \log_2(\lambda)$, then, there is no infinite subsequence $K \subseteq Q$ such that,

$$\forall\lambda \in K : \Pr\left[\mathcal{A}_\lambda(\rho_\lambda, \mathsf{O}_T) \in \left(S^\perp \setminus T^\perp\right) \,\middle|\, T \leftarrow \mathcal{S}^{\subseteq}_{\lambda-t}, \mathsf{O}_T \leftarrow \mathsf{iO}(C_T)\right] < \frac{1}{2 \cdot \lambda^{(c+2)\cdot 2}} \enspace ,$$

where the above obfuscation of C_T is padded with some sufficiently large polynomial $\mathrm{poly}(\lambda)$ *number of bits* $1^{\mathrm{poly}(\lambda)}$.

Proof. Assume toward contradiction that the claim is false and there is such quantum algorithm $\mathcal{A}$. Define the following probability,

$$p_\lambda := \Pr\left[\mathcal{A}_\lambda(\rho_\lambda, \mathsf{O}_T) \in \left(S^\perp \setminus T^\perp\right) \,\middle|\, T \leftarrow \mathcal{S}^{\subseteq}_{\lambda-t}, \mathsf{O}_T \leftarrow \mathsf{iO}(C_T)\right] \enspace .$$

Then, our assumption (towards contradiction) is that there is an infinite set $K \subseteq Q$ such that for every $\lambda \in K$ we have $p_\lambda < \frac{1}{2\cdot\lambda^{(c+2)\cdot 2}}$.

For short notations in the proof, we denote by $\mathsf{iO}(C_S)$ the random variable that samples an obfuscation of the circuit C_S, and by $\mathsf{iO}(T)$ the random variable that first samples a superspace $T \leftarrow \mathcal{S}^{\subseteq}$ and then outputs a sample obfuscation of C_T. Naturally, these random variables depend on the security parameter $\lambda \in \mathbb{N}$, which is also dropped from these notations. We'll use $\mathcal{A}$ to distinguish between the distributions $\{\mathsf{O}_S | \mathsf{O}_S \leftarrow \mathsf{iO}(C_S)\}$ and $\{\mathsf{O}_T | T \leftarrow \mathcal{S}^{\subseteq}, \mathsf{O}_T \leftarrow \mathsf{iO}(C_T)\}$, and due to the fact that the dimension of T is $\lambda - t$ and also $t \geq \lambda^\delta$ for some constant $\delta \in (0,1)$, such distinguisher is in contradiction to the subspace hiding property of indistinguishability obfuscation (Lemma 3.1). We first arrange some details and intuitions that are helpful for the proof.

Initial observations of the adversary's behavior on obfuscations of T. First, let us observe some basic probabilistic facts.

- By an averaging argument, there is at least a fraction of $\frac{1}{2\cdot\lambda^c}$ out of the possible choices for T, such that for such choices of T, with probability at least $\frac{1}{2\cdot\lambda^c}$ we have $\mathcal{A}_\lambda(\rho_\lambda, \mathsf{O}_T) \in \left(T^\perp \setminus \{0\}\right)$. This follows directly from the Lemma's assumptions (and not from our assumption towards contradiction).
- From our assumption towards contradiction that $p_\lambda < \frac{1}{2\cdot\lambda^{(c+2)\cdot 2}}$, by an additional different averaging argument, it follows that for at most a fraction of $\frac{1}{2\cdot\lambda^{(c+2)\cdot 2}} \cdot k$ of the choices of T, we have that the probability for $\mathcal{A}_\lambda(\rho_\lambda, \mathsf{O}_T) \in \left(S^\perp \setminus T^\perp\right)$ is at least $\frac{1}{k}$. This is in particular true for $k := 2 \cdot \lambda^{(c+2)}$. We get that for at most a fraction of $\frac{1}{2\cdot\lambda^{(c+2)}}$ of the choices of T, we have that the probability for $\mathcal{A}_\lambda(\rho_\lambda, \mathsf{O}_T) \in \left(S^\perp \setminus T^\perp\right)$ is at least $\frac{1}{2\cdot\lambda^{(c+2)}}$.

- A conclusion of the combination of the two items above, implies that for at least a fraction of $\frac{1}{4\cdot\lambda^c}$ from the choices of T,
  - The event $\mathcal{A}_\lambda(\rho_\lambda, \mathsf{O}_T) \in \left(T^\perp \setminus \{0\}\right)$ happens with at least probability $\frac{1}{2\cdot\lambda^c}$, and
  - The event $\mathcal{A}_\lambda(\rho_\lambda, \mathsf{O}_T) \in \left(S^\perp \setminus T^\perp\right)$ happens with at most probability $\frac{1}{2\cdot\lambda^{(c+2)}}$.

  We'll call the above set of choices for T (which we know happens with at least probability $\frac{1}{\lambda^c\cdot 4}$) the set of good T's.

Our reduction. Our reduction is the following, and it is used to distinguish between two distributions, where each of the two distributions contains $\ell := \lambda^c \cdot \lambda \cdot (t+1)$ samples of obfuscations $\left(\mathsf{O}^{(1)}, \cdots, \mathsf{O}^{(\ell)}\right)$ (we did not specify yet exactly obfuscations of what). Given the ℓ obfuscations, execute $\mathcal{A}(\rho, \cdot)$ on each of them and obtain ℓ vectors $\{u_1, \cdots, u_\ell\}$. Then, take only the vectors $\{v_1, \cdots, v_m\}$ that are inside $S^\perp$, and then compute the dimension of their span, $D := \dim\left(\text{span}\left(v_1, \cdots, v_m\right)\right)$.

The first distribution out of the two will simply be $\left(\mathsf{O}_S^{(1)}, \cdots, \mathsf{O}_S^{(\ell)}\right)$, that is, ℓ i.i.d. obfuscations of C_S. We'll call this distribution $\mathcal{D}_S$. Depending on the adversary's output behavior on this distribution, we will decide on the second distributions. Specifically, the second distribution will be one of the following two:

- $\mathcal{D}_1$: Sample T once, then sample ℓ i.i.d. obfuscations of it, $\mathsf{O}_T^{(1)}, \cdots, \mathsf{O}_T^{(\ell)}$.
- $\mathcal{D}_2$: Sample ℓ i.i.d superspaces $T_1, \cdots, T_\ell$, and for each of them, send a single obfuscations of it: $\mathsf{O}_{T,1}, \cdots, \mathsf{O}_{T,\ell}$.

Intuitively, we will see that whenever the reduction gets a sample from $\mathcal{D}_1$ then it should be the case that the output dimension D is bounded by t, and whenever the reduction gets a sample from $\mathcal{D}_2$, then D is at least $t+1$ with high probability. The point is, that $\mathcal{D}_S \approx_c \mathcal{D}_1 \approx_c \mathcal{D}_2$. That is, $\mathcal{D}_S \approx_c \mathcal{D}_1$ directly by Lemma 3.1, and $\mathcal{D}_S \approx_c \mathcal{D}_2$ by a hybrid argument of that same Lemma 3.1.

Our decision for what indistinguishability to use is derived from this: Whether given a sample from $\mathcal{D}_S$, the probability for $D \geq t+1$ is at least $1 - \frac{1}{16\cdot\lambda^c}$ or not.

First Case: The output of the adversary is scattered on obfuscations of C_S. Formally, when the reduction gets a sample from $\mathcal{D}_S$, then with probability at least $1 - \frac{1}{16\cdot\lambda^c}$ the output dimension is $D \geq t+1$. We will see a distinguisher between $\mathcal{D}_S$ and $\mathcal{D}_1$, by showing that when the input of the reduction is a sample from $\mathcal{D}_1$, then the probability for $D \geq t+1$ is bounded by $1 - \frac{1}{8\cdot\lambda^c}$.

To see this, on an input sample from $\mathcal{D}_1$, by union bound, the probability for $D \geq t+1$ is bounded by the sum of probabilities for such event for when T is inside and outside of the good set. The probability that the (single) sampled T is outside the good set is bounded by $1 - \frac{1}{4\cdot\lambda^c}$. The probability that T is inside the good set is at least $\frac{1}{4\cdot\lambda^c}$, and note that whenever T is inside the good set, the probability for $D \geq t+1$ is bounded by the probability that any of the ℓ executions of $\mathcal{A}$ produced a vector in $\left(S^\perp \setminus T^\perp\right)$, and by union bound, the aforementioned probability is bounded by $\lambda^{c+2} \cdot \frac{1}{2\cdot\lambda^{c+2}} = \frac{1}{2}$. In total, the probability for the reduction to get $D \geq t+1$ on a sample from $\mathcal{D}_1$ is bounded by

$$\left(1 - \frac{1}{4\cdot\lambda^c}\right)\cdot 1 + \frac{1}{4\cdot\lambda^c}\cdot\frac{1}{2} = 1 - \frac{1}{8\cdot\lambda^c} \quad .$$

Second Case: The output of the adversary is concentrated on obfuscations of C_S. Formally, when the reduction gets a sample from $\mathcal{D}_S$, then with probability less than $1 - \frac{1}{16\cdot\lambda^c}$ the output dimension is $D \geq t+1$. We will see a distinguisher between $\mathcal{D}_S$ and $\mathcal{D}_2$, by showing that when the input of the reduction is a sample from $\mathcal{D}_2$, then the probability for $D \geq t+1$ is at least $1 - \frac{1}{32\cdot\lambda^c}$.

Let us see what happens when the sample came from the distribution $\mathcal{D}_2$, that is, ℓ i.i.e. obfuscations of different super-spaces T_i. Consider the ℓ vectors $\{u_1, \cdots, u_\ell\}$ obtained by executing $\mathcal{A}$ on each of the input obfuscations. Recall that $\ell := \lambda^c \cdot \lambda \cdot (t+1)$ and consider a partition of the vectors into $t+1$ consecutive sequences (or buckets), each of length λ^{c+1}. For every bucket $j \in [t+1]$ it is the case with probability $1 - e^{\Omega(\lambda)}$, at least one u_i in that λ^{c+1}-sized set of samples satisfies that it is a non-zero vector inside $T_i^\perp \setminus \{0\}$ the dual subspace (of its obfuscated subspace T_i). It follows that with probability $\left(1 - e^{\Omega(\lambda)}\right)^{t+1}$ (which equals the overwhelming probability $1 - e^{\Omega(\lambda)}$ because recall there is some constant $\delta \in (0,1)$ such that $\forall \lambda \in \mathbb{N} : t_\lambda \geq \lambda^\delta$) there exists a subset of $\{u_1, \cdots, u_\ell\}$ containing $t+1$ vectors $\{w_1, \cdots, w_{t+1}\}$ such that for every $i \in [t+1]$, the vector w_i is inside $T_{j_{w_i}}^\perp \setminus \{0\}$ such that $T_{j_{w_i}}$ is the corresponding subspace to w_i, or more formally: $j_{w_i} \in \{(i-1)\lambda^{c+1}+1, \cdots, (i-1)\lambda^{c+1} + \lambda^{c+1}\}$ is the index such that w_i was the output of $\mathcal{A}\left(\rho, \mathsf{O}_{T_{j_{w_i}}}\right)$.

It remains to observe that for every $i \in [t+1]$, the probability that $w_i \in \text{span}(w_1, \cdots, w_{i-1})$ is bounded by the probability that the intersection between $T_{j_{w_i}}$ and $S_{i-1} := \text{span}(w_1, \cdots, w_{i-1})$ has non-zero vectors, which is in turn bounded by

$$\prod_{j \in [t]} \frac{|S_{i-1}| \cdot 2^{j-1}}{|S^\perp|} = \prod_{j \in [t]} \frac{2^{i-1} \cdot 2^{j-1}}{2^{\lambda - d}} = 2^{i-2-\lambda+d} \cdot \prod_{j \in [t]} 2^j$$

$$= 2^{i-2-\lambda+d} \cdot \left(2^{t+1} - 2\right) < 2^{i-2-\lambda+d+t+1} = 2^{d+t+i-1-\lambda} \quad .$$

It follows that the probability for $\dim(\text{span}(w_1, \cdots, w_{t+1})) = t+1$ is

$$\prod_{i \in [t+1]} \left(1 - 2^{d+t+i-1-\lambda}\right) \geq \prod_{i \in [t+1]} \left(1 - 2^{d+t+(t+1)-1-\lambda}\right) = \left(1 - 2^{d+2t-\lambda}\right)^{t+1} \quad .$$

Finally, we trivially have $\left(1 - 2^{d+2t-\lambda}\right)^{t+1} \geq 1 - 2^{t+1} \cdot 2^{d+2t-\lambda} = 1 - 2^{d+3t+1-\lambda}$. By the assumptions in our Lemma 5.1 statement we have $\lambda - (d_\lambda + 3 \cdot t_\lambda + 1) \geq 5 + c \cdot \log_2(\lambda)$, which makes the above probability at least $1 - \frac{1}{32 \cdot \lambda^c}$. □

Acknowledgements

We are grateful to Tamer Mour, for helpful discussions during the writing of this work.

References

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

[AC12] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 41–60, 2012.

[BDGM20a] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate io from homomorphic encryption schemes. 2020.

[BDGM20b] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for io: Circular-secure lwe suffices. *IACR Cryptol. ePrint Arch.*, 2020:1024, 2020.

[BDS16] Shalev Ben-David and Or Sattath. Quantum tokens for digital signatures. *arXiv preprint arXiv:1609.09047*, 2016.

[Bra18] Zvika Brakerski. Quantum fhe (almost) as secure as classical. In *Annual International Cryptology Conference*, pages 67–95. Springer, 2018.

[CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In *Annual International Cryptology Conference*, pages 556–584. Springer, 2021.

[DQV$^+$21] Lalita Devadas, Willy Quach, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Succinct lwe sampling, random polynomials, and obfuscation. *Cryptology ePrint Archive*, 2021.

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 736–749, 2021.

[Mah20] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. *SIAM Journal on Computing*, (0):FOCS18–189, 2020.

[Rad19] Roy Radian. Semi-quantum money. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 132–146, 2019.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[Shm21] Omri Shmueli. Public-key quantum money with a classical bank. *Cryptology ePrint Archive*, 2021.

[Zha19] Mark Zhandry. Quantum lightning never strikes the same state twice. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 408–438. Springer, 2019.