cryptography

Article

Quantum Security of a Compact Multi-Signature

Shaoquan Jiang

School of Computer Science, University of Windsor, Windsor, ON N9B 3P4, Canada; jiangshq@uwindsor.ca

Abstract: With the rapid advances in quantum computing, quantum security is now an indispensable property for any cryptographic system. In this paper, we study how to prove the security of a complex cryptographic system in the quantum random oracle model. We first give a variant of Zhandry's compressed random oracle (**CStO**), called a compressed quantum random oracle with adaptive special points (**CStO**$_s$). Then, we extend the on-line extraction technique of Don et al. (EUROCRYPT'22) from **CStO** to **CStO**$_s$. We also extend the random experiment technique of Liu and Zhandry (CRYPTO'19) for extracting the **CStO** query that witnesses the future adversarial output. With these preparations, a systematic security proof in the quantum random oracle model can start with a random **CStO** experiment (that extracts the witness for the future adversarial output) and then converts this game to one involving **CStO**$_s$. Next, the online extraction technique for **CStO**$_s$ can be applied to extract the witness for any online commitment. With this strategy, we give a security proof of our recent compact multi-signature framework that is converted from any weakly secure linear ID scheme. We also prove the quantum security of our recent lattice realization of this linear ID scheme by iteratively applying the weakly collapsing protocol technique of Liu and Zhandry (CRYPTO 2019). Combining these two results, we obtain the first quantum security proof for a compact multi-signature.

Keywords: compressed quantum random oracle; ring-LWE; multi-signature; identification scheme

1. Introduction

A multi-signature scheme allows a group of signers to jointly generate a signature while any subset of them cannot represent the group. This mechanism was introduced by Itakura and Nakamura [1] with the motivation to reduce the signature size. In the blockchain application [2], it is also demanded that the aggregated public key that represents the group should also have a small size, as it will be part of the transaction and the network storage. The blockchain has no control over a user, and hence, one should be able to freely decide his public keys. Accordingly, we must make sure that it is secure against a rogue key attack: the attacker might choose his public key after seeing other signers' public keys. In a poorly designed scheme, an attacker could manage to decide the secret key of the aggregated public-key. In addition, with the advances of quantum computing, the quantum attack places a major threat to any cryptographic system. Especially, the RSA based multi-signature (such as [3]) is no longer secure [4]. In this paper, we investigate the multi-signature security in the quantum random oracle model, where the attacker has an internal quantum computer and also can access to the quantum random oracle. We aim to develop quantum random oracle techniques that enable a security proof of a complex cryptographic system. We then apply it to prove the security of our recent compact multi-signature.

1.1. Related Works

A multi-signature scheme [1] is a special case of aggregate signature [5], where each signer of the latter can sign a possibly different message. Since it was introduced by Itakura and Nakamura [1], it has been intensively studied in the literature [3,6–14]. However, most of schemes are based on some variants of a discrete logarithm assumption, which does not

check for updates

Citation: Jiang, S. Quantum Security of a Compact Multi-Signature. *Cryptography* **2024**, *8*, 50. https://doi.org/10.3390/cryptography8040050

Received: 28 August 2024
Revised: 21 October 2024
Accepted: 23 October 2024
Published: 28 October 2024

hold under a quantum attack [4]. There are multi-signatures that are based on quantum mechanics only (i.e., without a computational hardness assumption) [15,16]. However, their schemes are certainly not what is understood in the crypto community: (1) signers need to share a private key with a trusted party; (2) the verification is completely done by the trusted party; and (3) the signer has no public key.

Constructions from lattice assumptions such as (ring-)LWE are potentially the solutions for the quantum-secure multi-signature problem. However, there are currently only very few schemes [17–22] from this. In addition, some schemes [19,20] are known to be insecure [21,23]. Schemes [17,18,22–26] did not consider a quantum attacker. Fukumitsu and Hasegawa [27] is the only previous scheme that considered the quantum security. Their construction is based on Dilithium signature [28]. However, their scheme only allows a constant number of signers, and the verification requires all signers' public keys. Their proof technique (also that of Dilithium [28]) seems to rely on the statistical lossy property of the underlying ID scheme, and it is unclear if it can be generally usable in other security analysis. In this paper, we investigate general quantum random oracle techniques that are useful in proving a wide class of random oracle-based systems. With this, we prove the quantum security of our recent multi-signature framework [23].

The random oracle basically models a hash function as a completely random function. It was first proposed by Bellare and Rogaway [29]. This methodology has a heuristic assumption: when the random oracle is replaced by a cryptographic hash function, the security will be preserved. This generally is not true [30]. However, the counter example does not seem realistic. So, the crypto community still widely believes that this methodology is practically meaningful. Furthermore, it greatly simplifies the construction of many cryptographic systems and the proof in the classical random oracle model is usually amazingly simple. However, it is not true in the quantum world. The great advantage of a classical random oracle is that the simulator can easily record the attacker's query history. In the quantum setting, this is difficult as an attacker can query a superposition. If the simulator makes a measurement on the query, it will destroy the quantum state. Zhandry [31] proposed new techniques to record the oracle query, which is called a compressed random oracle ($CStO$). Essentially, if the oracle is only queried q times, then the oracle can be compactly represented as a superposition of a database, with the basis record only containing at most q non-trivial values. Don et al. [32] showed a simulation that can extract an oracle query of a (classic) commitment on the fly. The impact of this feature is that if an adversary outputs a commitment value, we can immediately extract his query input that matches this commitment. This will not destroy the quantum state essentially because when an attacker outputs his classical commitment, he must have already made the measurement. Hence, this gives us a very useful tool, especially when a simulator needs to know the query in order to continue the simulation. However, this is not enough in some proofs. For example, in our multi-signature scheme, the adversary will receive a honest user's public key pk_1 and then generate two public keys pk_2, pk_2. At the end, he will try to forge a signature with respect to a combined public key $F(pk_1, pk_2, pk_3)$ that is computed from $H(pk_i|pk_1|pk_2|pk_3)$ for $i = 1, 2, 3$, and H is the random oracle. The problem is that pk_2, pk_3 will be revealed only at the end of the game. If the simulator wishes to know it in advance, it is impossible using the techniques in [32]. Liu and Zhandry [33] presented a measurement technique to extract pk_2, pk_3 during the game involving $CStO$. Essentially, it chooses a random query and measures it. Then, the outcome is $pk_i|pk_1|pk_2|pk_3$ for some i with a good probability. Furthermore, the adversary success probability for the forgery will be degraded only by a polynomial fraction. For technique reasons, it is desired that the simulator can set the random oracle value of the measure outcome $pk_i|pk_1|pk_2|pk_3$ (called a special point) to a value of his favorite. To take the advantage of both extraction techniques, one might consider the simulation of [32] with the measurement techniques in [33]. However, there are two issues. First, some verification measurements in [33] will be done on the random oracle database, and hence, the extraction theorems in [32] will no longer hold. Second, the special input measurement [33] is operated only once. This sometimes is insufficient to produce a witness for the final adversary output. Our work in this paper is to propose an improved *CStO* that addresses the two issues and then apply the improved random oracle techniques to prove the security of our recent compact multi-signature scheme [23].

1.2. Contribution

In this paper, we study how to improve *CStO* so that it still has a simulator (similar to [32]) that allows us to extract a query input of any given commitment on the fly but additionally also allows us to adaptively specify a small number of special points and set their random oracle values according to our own choices. The improved random oracle is called a compressed random oracle with adaptive special points ($\mathbf{CStO}_s$). We generalize the simulator and extraction theorem in [32] to the $\mathbf{CStO}_s$ setting. We also generalize the experiment sampling technique in [33] to allow samplings for several times. This allows us to extract the witness of the final adversary output, where this witness might depend on several random oracle queries (that are measured during the game). This random experiment can be easily converted to an interaction with $\mathbf{CStO}_s$ oracle, and hence, the foregoing online extraction technique can be applied. With this improved random oracle technique, we show that our recent multi-signature framework (which is converted from any weakly secure linear identification scheme) is provably secure in the quantum random oracle model. The proof strategy is to use the sequence of the game technique. It starts the adversary with a standard quantum random oracle and then continues with the compressed quantum random oracle (**CStO**) while preserving the same adversary success probability. It next applies the random experiment sampling techniques, which degrades the adversary success only by a polynomial fraction, but it can extract the witness for the final adversary output. Then, we convert the random experiment (with **CStO**) to one involving $\mathbf{CStO}_s$. Finally, the online extraction technique is used to simulate the interaction without the knowledge of the secret of an ID scheme. This allows us to reduce the adversary success to the security of the ID scheme. We also prove the quantum security of the JAK ID scheme in [23]. The main tool to achieve this is to use the collapsing sigma protocol technique in [33] that was originally proposed by Unruh [34]. Essentially, our security proof is to formulate the JAK ID security game into two public-coin protocols, each of which uses the collapsing property to guarantee the non-negligibility of the adversary success probability. This two-step analysis allows us to reduce the adversary success probability in attacking the JAK ID scheme to break the underlying ring-SIS assumption.

This paper is organized as follows. In Section 2, we present some essential notations and definitions that will be used in the paper. In Section 3, we present some basic properties in quantum computing that are useful in this work. In Section 4, we present **CStO** and our extension to $\mathbf{CStO}_s$. In Section 5, we show how to measure the record in $\mathbf{CStO}_s$ to see if a given relation R is satisfied or not. In Section 6, we show how to extract a query x in $\mathbf{CStO}_s$ that satisfies a given commitment $t = f(x, RO(x))$. In Section 7, we extend the query extraction technique of Liu and Zhandry [33] that witnesses the future adversarial output. In Section 8, we prove the quantum security of our previous multi-signature framework using the techniques in Sections 6 and 7. In Section 9, we prove the quantum security of the JAK ID scheme, which together with the multi-signature theorem, gives the first quantum security of a compact multi-signature scheme. The last section is the conclusion.

2. Preliminaries

Notations. We will use the following notations:

- $x \leftarrow S$ samples x uniformly random from a set S.
- For a randomized algorithm A, $u = A(x; r)$ denotes the output of A with input x and randomness r, while $u \leftarrow A(x)$ denotes the random output (with unspecified randomness).
- Min-entropy $H_\infty(X) = -\log(\max_x \log P_X(x))$. This is widely known as the worst uncertainty of X, while the well-known Shannon entropy $H(X)$ is its average uncertainty.
- A concatenating with B is denoted by $A|B$ and also by (A, B) (if the context is clear).

- A non-negative function $\mathsf{negl}(\lambda)$ is *negligible* if it vanishes faster than any polynomial fraction. That is, for any polynomial $poly(\lambda)$, there exists $N > 0$ so that when $\lambda > N$, it holds that $\mathsf{negl}(\lambda) < 1/poly(\lambda)$.
- $[\nu]$ denotes set $\{1, \cdots, \nu\}$.
- $\mathcal{Y}^{\mathcal{X}}$ denotes the set of vector $\mathbf{y} := \{y_x\}_{x \in \mathcal{X}}$. That is, each entry in $\mathbf{y}$ is indexed by $x \in \mathcal{X}$. We use $\mathbf{y}(x)$ to denote the entry y_x.

2.1. Ring and Module

In this section, we review math concepts: the commutative ring and module (see [35] for details). We start from the integer set $\mathbb{Z}$. It is clear that it has a multiplicative identity of 1 (so $1 \cdot z = z$ for any $z \in \mathbb{Z}$) and an additive identity of 0 (so $0 + z = z$ for any $z \in \mathbb{Z}$). It forms a group under operator +. But, it is not a group under multiplication as any integer other than -1, as 1 has no inverse in $\mathbb{Z}$. But, it is associative: $(ab)c = a(bc)$. It satisfies the distributive law: $a(b+c) = ab + ac$ and $(b+c)a = ba + ca$. Actually, $\mathbb{Z}$ is a special case of a general concept ring. In this work, we are only concerned with a *commutative ring*.

A **commutative ring** A is a set associated with multiplication and addition operators that is respectively written as a product and a sum, satisfying the following conditions for any $a, b, c \in A$:

- **R-0.** It has a unit $\mathbf{1}$ and is commutative under multiplication: $ab = ba$ and $\mathbf{1}a = a$.
- **R-1.** A is a commutative group under addition operator $+$ with identity element $\mathbf{0}$.
- **R-2.** A is associative under multiplication operator: $(ab)c = a(bc)$.
- **R-3.** It satisfies the distributive law: $a(b+c) = ab + ac$ and $(b+c)a = ba + ca$.

For simplicity, we use the term *ring* to represent a *commutative ring* in this paper. If A is a ring with $\mathbf{0} \neq \mathbf{1}$, and every non-zero element in A has an inverse, then A is a **field**. The rational number set $\mathbb{Q}$, the real number set $\mathbb{R}$, and the complex number set $\mathbb{C}$ are all examples of a field.

Another concept of our interest is the module. A module is actually a simple generalization of a *vector space*. Recall that a vector space is an additive group V that is associated with a coefficient field F. We can take $V = \mathbb{R}^n$ and $F = \mathbb{R}$ as an example. In this example, it is distributive: (1) for $\mathbf{v}_1, \mathbf{v}_2 \in V, r \in F$, it has $r(\mathbf{v}_1 + \mathbf{v}_2) = r\mathbf{v}_1 + r\mathbf{v}_2$; (2) for $r_1, r_2 \in F, \mathbf{v} \in V$, it has $(r_1 + r_2)\mathbf{v} = r_1\mathbf{v} + r_2\mathbf{v}$. It is also associative: for $r, s \in F$ and $\mathbf{v} \in V$, it has $(rs)\mathbf{v} = r(s\mathbf{v})$. Also, trivially, $1\mathbf{v} = \mathbf{v}$. This notation can be generalized so that the coefficient set F is a ring (not just a field). In fact, $F = \mathbb{Z}$ is a good example for this. Also, the addition in V and the addition in F do not need to be the same; similarly, the multiplication between F and V and the multiplication in F do not need to be the same. With these changes in mind, the formal definition of a *module* can be given as follows.

Definition 1. *Let R be a ring. An additive group M (with group operator $\boxplus$) is a R-**module** if (1) it has defined a multiplication operator $\bullet$ between R and M: for any $r \in R, m \in M$, $r \bullet m \in M$; and (2) the following conditions are satisfied: for any $r, s \in R$ and $x, y \in M$:*

1. $r \bullet (x \boxplus y) = (r \bullet x) \boxplus (r \bullet y)$;
2. $(r + s) \bullet x = (r \bullet x) \boxplus (s \bullet x)$
3. $(rs) \bullet x = r \bullet (s \bullet x)$
4. $1_R \bullet x = x$, *where 1_R is the multiplicative identity of R.*

2.2. Elements of Quantum Computing

We give a brief introduction to quantum computing through a list of notations and some facts, with interpretations if necessary; see [36,37] for details:

- A quantum system is a finite-dimensional complex vector space (called Hilbert space) $\mathcal{H}$ with an inner product $\langle \cdot | \cdot \rangle$.
- The state of a quantum system in $\mathcal{H}$ is a unit vector $|\psi\rangle$. Its conjugate transpose is denoted by $\langle \psi |$.

- Let $\mathcal{Y}$ be a finite Abelian group. We use $\{|y\rangle\}_{y\in\mathcal{Y}}$ to represent an orthonormal basis for $\mathcal{H} = \mathbb{C}^{|\mathcal{Y}|}$. We denote $\mathcal{H}$ by $\mathbb{C}[\mathcal{Y}]$ to emphasize that $\mathcal{H}$ is expanded by $\{|y\rangle\}_{y\in\mathcal{Y}}$.
- For two quantum systems $\mathcal{H}_1$ and $\mathcal{H}_2$, the joint system is a tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$.
- For $|\psi_1\rangle \in \mathcal{H}_1$ and $|\psi_2\rangle \in \mathcal{H}_2$, their product state in $\mathcal{H}_1 \otimes \mathcal{H}_2$ is $|\psi_1\rangle \otimes |\psi_2\rangle$. We write it as $|\psi_1\rangle|\psi_2\rangle$ for simplicity.
- A quantum register is a system holding the quantum state. It is the quantum analogue of the classical processor register. We use $|\psi\rangle_A$ to represent the register A containing quantum state $|\psi\rangle$.
- For an ordered set $\mathcal{X} = \{x_1, \cdots, x_n\}$, $\mathbb{C}[\mathcal{Y}]^{\otimes\mathcal{X}}$ represents the tensor product of $|\mathcal{X}|$ copies of $\mathbb{C}[\mathcal{Y}]$ with the ith copy labeled by x_i.
- Assume that quantum system $\mathcal{H}$ has an orthonormal basis $\{|\psi_1\rangle, \cdots, |\psi_n\rangle\}$. With this, a quantum state $|\psi\rangle \in \mathcal{H}$ can be represented as $|\psi\rangle = \sum_{i=1}^{n} \lambda_i|\psi_i\rangle$, with $\sum_i |\lambda_i|^2 = 1$.
- Let $\mathcal{L}(\mathcal{H})$ denote the set of linear operators from $\mathcal{H}$ to $\mathcal{H}$. For $A, B \in \mathcal{L}(\mathcal{H})$, their commutator is defined as $[A, B] = AB - BA$.
- Physically realizable quantum operations on $\mathcal{H}$ are unitaries and measurements.
- A unitary U on $\mathcal{H}$ is an operator from $\mathcal{H}$ to $\mathcal{H}$ with $UU^\dagger = I$, where $U^\dagger$ is the conjugate transpose of U.
- Measurement $M = \{M_i\}_i$ on a quantum state $|\psi\rangle \in \mathcal{H}$ is the operator for extracting the classical information from $|\psi\rangle$, where each M_i must be Hermitian (i.e., $M_i^\dagger = M_i$) and satisfies the completeness condition $\sum_i M_i^\dagger M_i = I$. When M is applied, it will result in a post-measurement state $M_i|\psi\rangle/||M_i|\psi\rangle||$ with probability $||M_i|\psi\rangle||^2$.
- A quantum algorithm A is represented by a sequence of unitaries/measurements. Due to deferred measurement principle ([36], p. 186), the measurement can be deferred to the end of operations of A. Hence, whenever applicable, we assume that A before the final measurement is represented by a list of unitaries $U_1, \cdots, U_\ell$.
- If $|1\rangle, \cdots, |n\rangle$ is an orthonormal basis of $\mathcal{H}$, then $P = \sum_{k\in A} |k\rangle\langle k|$ for $A \subset [n]$ is a projector from $\mathcal{H}$ onto the subspace expaned by $\{|k\rangle\}_{k\in A}$.
- The norm of linear operator A on $\mathcal{H}$ is defined as $||A|| = \max_v ||A|v\rangle||$, where $|v\rangle$ goes over all the possible unit vectors in $\mathcal{H}$. According the singular value decomposition theorem, we can write $A = \sum_i \lambda_i|v_i\rangle\langle y_i|$, where $\{|v_i\rangle\}_i$ and $\{|y_i\rangle\}_i$ are, respectively, a set of orthonormal vectors in $\mathcal{H}$ and $\{\lambda_i\}_i$ is the set of positive singular values of A. Hence, $||A|| = \max_i \lambda_i$.
- For states $|\psi_i\rangle$ and $0 \le \lambda_i \le 1, i = 1, \cdots, n$ with $\sum_{i=1}^{n} \lambda_i = 1$, $\rho = \sum_{i=1}^{n} \lambda_i|\psi_i\rangle\langle\psi_i|$ is called a *mixed state* or simply *state* when the context is clear. When $\{|\psi_i\rangle\}_i$ are orthonormal, ρ can be explained as $|\psi_i\rangle$ is sampled with probability λ_i.
- The trace distance between two mixed states ρ, σ is defined as $D_t(\rho, \sigma) = \frac{1}{2}\mathrm{tr}(|\rho - \sigma|)$, where $|A| := \sqrt{A^\dagger A}$. If $\rho = \sum_{i=1}^{n} p_i|\psi_i\rangle\langle\psi_i|$ and $\sigma = \sum_{i=1}^{n} q_i|\psi_i\rangle\langle\psi_i|$ for orthonormal basis $\{|\psi_i\rangle\}_i$; then, $D_t(\rho, \sigma) = \frac{1}{2}\sum_{i=1}^{n} |p_i - q_i|$, which coincides with the statistical distance of distributions $P = (p_1, \cdots, q_n)$ and $Q = (q_1, \cdots, q_n)$.

2.3. Multi-Signature

In this section, we introduce the multi-signature and its security model.

2.3.1. Syntax

A *multi-signature* scheme is a protocol that allows a group of signers to jointly generate a signature. The signers can generate their public/private keys independently without a trusted party. The signers have a joint *public key* (called an aggregated public key) that is derived from all signers' public keys. The signature should be valid against their *aggregated public key*. The multi-signature is motivated by the blockchain application, where one can pay to the signers through the aggregated public key, and the signers can spend the received money by jointly generating a multi-signature as an authorization of their pay. This system must be able to prevent an attacker (possibly an insider) from forging a signature under the aggregated public key.

A straightforward multi-signature is to let all signers generate individual signatures and concatenate them together. But in this case, the signature size is linear in the number of signers. A good multi-signature should be much shorter, and the aggregated public key is desired to be as short as possible too. This is because the both signature and the aggregated public key will be part of the transaction in the blockchain application.

Definition 2. *A* **multi-signature** *scheme is a quadruple of algorithms (***Setup, KeyGen, Sign, and Verify***) described as follows.*

Setup. *Given 1^λ, it generates a system parameter* param*. Note:* param *should be part of the input for* **KeyGen, Sign, and Verify**. *But, we usually omit it for brevity.*

KeyGen. *It takes* param *as input and generates a private key sk and a public key pk. In applications, this will be executed by a user himself.*

Sign. *Given public keys $(pk_1, \cdots, pk_n)$ and a message M user i has the private key sk_i with respect to pk_i. Then, they interact with each other and finally output a signature σ with respect to an aggregated public key $\overline{pk} := F(pk_1, \cdots, pk_n)$, where F is called an aggregation function.*

Verify. *Upon (σ, M) and an aggregated public key $\overline{pk} = F(pk_1, \cdots, pk_n)$, the verifier outputs either 1 (for* accept*) or 0 (for* reject*).*

Remark 1. *The aggregated key $\overline{pk}$ carries the information of the signers' public keys. It is desired that it has a size independent of n. But, this is not enforced in the definition.*

2.3.2. Security Model

In the following, we define the existential unforgeability of a multi-signature in the quantum random oracle model. Essentially, it says that no quantum adversary can forge a valid signature on a new message as long as the signing group contains an honest member. Toward this, the attacker can access to a signing oracle and quantum random oracle and create fake public keys at will. In the blockchain setting, this captures the security concern: an attacker can create many fake accounts, but he cannot represent a group containing a honest user to enable a transaction without this honest user's participating, even if the attacker has seen many transactions involving this user. We consider the security in the quantum setting, where the attacker could have an internal quantum computer, and its quantum state will be updated after each interaction with an external challenger. This captures the concern that the attacker makes use of an internal quantum computer to help break the multi-signature system that is used externally. Formally, the multi-signature security is defined through a game between a challenger CHAL and a quantum attacker $\mathcal{A}$ that has oracle access to the quantum random oracle and signing oracle from CHAL.

Initially, CHAL generates param and a challenge public key pk^* with a private key sk^*. It then provides $pk^*|$param to $\mathcal{A}$, who has an initial state $|\psi\rangle = \sum_{xyw} \lambda_{xyw} |x\rangle_X |y\rangle_Y |w\rangle_W$, where X, Y, W represent the query register, response register, and working register, respectively. Next, $\mathcal{A}$ interacts with CHAL through the signing oracle and random oracle RO and finally generates a forgery.

Sign(PK, M). Here, PK is a set of *distinct* public keys with $pk^* \in PK$. Upon this query, CHAL represents the signer of pk^*, and $\mathcal{A}$ represents signer of $PK - \{pk^*\}$ to run the signing protocol on message M. Finally, it outputs the multi-signature σ (if it succeeds) or $\perp$ (if it fails).

RO. $\mathcal{A}$ can query the random oracle RO by providing his XY registers to CHAL, who applies RO on XYD so that $RO|x\rangle_X|y\rangle_Y|H\rangle_D = |x\rangle|y + H(x)\rangle|H\rangle$, where H is the random function, and D is the random oracle register maintained by a challenger. Finally, it returns registers XY back to $\mathcal{A}$. See the first paragraph of Section 4.1 for details.

Forgery. Finally, $\mathcal{A}$ outputs a signature σ^* for a message M^* with respect to a set of *distinct* public keys $(pk_1^*, \cdots, pk_N^*)$ such that $pk^* = pk_i^*$ for some i. $\mathcal{A}$ succeeds if

(a) $\mathsf{Verify}(\overline{pk^*}, \sigma^*, M^*) = 1$ and (b) $((pk_1^*, \cdots, pk_N^*), M^*)$ were not issued to **Sign** oracle. We denote a successful forgery event by **succ**.

Definition 3. *A multi-signature scheme (***Setup, KeyGen, Sign, and Verify***) is existentially unforgeable against a chosen message attack (or EU-CMA for short) in the quantum random oracle model if the following holds:*

- **Correctness.** *For $(sk_1, pk_1), \cdots, (sk_n, pk_n)$ generated by KeyGen, the signature generated by signing an algorithm on a message M will pass the verification, except for a negligible probability.*
- **Existential Unforgeability.** *For any quantum polynomial time adversary $\mathcal{A}$ in the above forgery game, $\Pr(\mathbf{succ}(\mathcal{A}))$ is negligible.*

2.4. Canonical Linear Identification

An identification system is a protocol that allows a user who has a public key and a private key to prove that he is the owner of the public key. Here, the public key is known to the verifier, while the private key is known only to the prover. A canonical identification system is a three-round public coin protocol where the first round message is from the prover, while the second message is a random number from the verifier. In addition, the first message has a super logarithmic entropy, which guarantees that correctly guessing it is difficult. The formal definition is presented as follows (also see Figure 1).

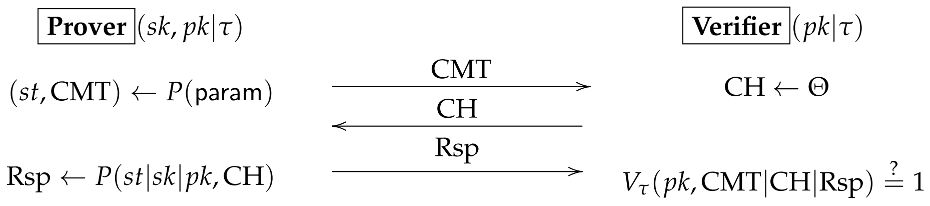
$\boxed{\textbf{Prover}}(sk, pk|\tau)$
$\boxed{\textbf{Verifier}}(pk|\tau)$
$(st, \mathrm{CMT}) \leftarrow P(\mathsf{param})$
CMT
CH
$\mathrm{CH} \leftarrow \Theta$
Rsp
$\mathrm{Rsp} \leftarrow P(st|sk|pk, \mathrm{CH})$
$V_\tau(pk, \mathrm{CMT}|\mathrm{CH}|\mathrm{Rsp}) \overset{?}{=} 1$

Figure 1. Canonical Identification Protocol.

Definition 4. *A canonical identification scheme with parameter $\tau \in \mathbb{N}$ is a quadruple of algorithms $\mathcal{ID} = (\mathbf{Setup}, \mathbf{KeyGen}, P, V_\tau)$, where* **Setup** *takes security parameter λ as input and generates a system parameter* param*;* **KeyGen** *is a key generation algorithm that takes* param *as input and outputs a public key pk and a private key sk; P is an algorithm executed by* Prover*; and V_τ is an algorithm parameterized by τ and executed by* Verifier*. $\mathcal{ID}$ is a three-round protocol, where Prover starts with a committing message* CMT *with $H_\infty(\mathrm{CMT}) = \omega(\log \lambda)$, then Verifier replies with a challenge $\mathrm{CH} \leftarrow \Theta$, and finally Prover finishes with a response* Rsp*, which will be either rejected or accepted by V_τ.*

The domains of sk, pk, CMT, and Rsp are respectively denoted by $\mathcal{SK}, \mathcal{PK}, \mathcal{CMT}, \mathcal{RSP}$. We are interested in a canonical ID scheme with linearity [23] and simulability in the following sense.

The motivation for the linearity is that if we linearly combine the transcripts of two protocol executions (with probably different provers), they become the identification transcript of the linearly combined public keys. This property will be used to combine the several ID transcripts into a compact multi-signature.

Linearity: A canonical ID scheme $\mathcal{ID} = (\mathbf{Setup}, \mathbf{KeyGen}, P, V_\tau)$ is **linear** if it satisfies the following conditions:

1. $\mathcal{SK}, \mathcal{PK}, \mathcal{CMT}, \mathcal{RSP}$ are $\mathcal{R}$-modules for some ring $\mathcal{R}$ with $\Theta \subseteq \mathcal{R}$ (as a set);
2. For any $\lambda_1, \cdots, \lambda_t \in \Theta$ and public/private pairs (sk_i, pk_i) $(i = 1, \cdots, t)$, we have that $\overline{sk} = \sum_{i=1}^t \lambda_i \bullet sk_i$ is a private key of $\overline{pk} = \sum_{i=1}^t \lambda_i \bullet pk_i$.
   Note: The operator $\bullet$ between $\mathcal{R}$ and $\mathcal{SK}$ (respectively defined as $\mathcal{PK}, \mathcal{CMT}, \mathcal{RSP}$) might be different. But, we will use the same symbol $\bullet$ as long as it is clear from the context.

3. Let $\lambda_i \leftarrow \Theta$ and $(pk_i, sk_i) \leftarrow \textbf{KeyGen}(1^\lambda)$, for $i = 1, \cdots, t$. If $\mathrm{CMT}_i|\mathrm{CH}|\mathrm{Rsp}_i$ is a *faithfully* generated transcript of the ID scheme with respect to pk_i, then

$$V_\tau(\overline{pk}, \overline{\mathrm{CMT}}|\mathrm{CH}|\overline{\mathrm{Rsp}}) = 1, \tag{1}$$

where $\overline{pk} = \sum_{i=1}^t \lambda_i \bullet pk_i, \overline{\mathrm{CMT}} = \sum_{i=1}^t \lambda_i \bullet \mathrm{CMT}_i$ and $\overline{\mathrm{Rsp}} = \sum_{i=1}^t \lambda_i \bullet \mathrm{Rsp}_i$.
Note: we require Equation (1) to hold only if the keys and transcripts are faithfully generated. If some are contributed by an attacker, this equality might fail.

Simulability. $\mathcal{ID}$ is simulatable if there exists a polynomial time algorithm **SIM** such that for $(sk, pk) \leftarrow \textbf{KeyGen}(1^\lambda)$, $\mathrm{CH} \leftarrow \Theta$, and $(\mathrm{CMT}, \mathrm{Rsp}) \leftarrow \textbf{SIM}(\mathrm{CH}, pk, \mathsf{param})$, it holds that CMT|CH|Rsp is indistinguishable from a real transcript, even if the quantum distinguisher is given $pk|\mathsf{param}$ and has access to oracle $\mathcal{O}_{id}(sk, pk)$, where $\mathcal{O}_{id}(sk, pk)$ acts as follows: $(st, \mathrm{CMT}) \leftarrow P(\mathsf{param})$; $\mathrm{CH} \leftarrow \Theta$; $\mathrm{Rsp} \leftarrow P(st|sk|pk, \mathrm{CH})$; output CMT|CH|Rsp.

Now, we define the security for a linear ID scheme. Essentially, it is desired that an attacker is unable to impersonate a prover with respect to an aggregated public key, where at least one of the participating public keys is not generated by an attacker. Here, we use the aggregated public key as the challenge key, because we will later convert an ID scheme into a multi-signature scheme. while the unforgeability security of a multi-signature is against the aggregated public key. In addition, we consider the security in the quantum setting: although the protocol itself does not involve a quantum message, an attacker could have a quantum computer internally and use this computer to help attack the classical protocol. Toward this, we allow the attacker to have an internal quantum state and will update it after receiving each message externally.

Definition 5. *A canonical identification scheme* $\mathcal{ID} = (\textbf{Setup}, \textbf{KeyGen}, P, V_\tau, \Theta)$ *with linearity and* $\tau \in \mathbb{N}$ *is* **secure** *if it satisfies correctness and security below.*

Correctness. *When no attack presents, Prover will convince Verifier.*

Soundness. *For any quantum polynomial time algorithm* $\mathcal{A}$, $\Pr(\mathrm{EXP}_{\mathcal{ID},\mathcal{A}} = 1)$ *is negligible, where* $\mathrm{EXP}_{\mathcal{ID},\mathcal{A}}$ *is defined below with* $pk_i \in \mathcal{PK}$ *for* $i \in [t]$ *and* $\overline{pk} = \sum_{i=1}^t \lambda_i \bullet pk_i$.

```
Experiment $\textsc{Exp}_{\mathcal{ID},\mathcal{A}}(\lambda)$
    $param \leftarrow \textbf{Setup}(1^\lambda)$;
    $(pk_1, sk_1) \leftarrow \textbf{KeyGen}(param)$;
    $(|st_0\rangle, pk_2, \cdots, pk_t) \leftarrow \mathcal{A}(param, pk_1)$
    $\lambda_1, \cdots, \lambda_t \leftarrow \Theta$
    $(|st_1\rangle, CMT) \leftarrow \mathcal{A}(|st_0\rangle, \lambda_1, \cdots, \lambda_t)$;
    $CH \leftarrow \Theta$; $Rsp \leftarrow \mathcal{A}(|st_1\rangle, CH)$;
    $b \leftarrow V_t(\overline{pk}, CMT|CH|Rsp)$;
    output $b$.
```

3. Basic Properties in Quantum Computing

In this section, we give some fundamental properties in quantum computing.

3.1. Properties of Commutators

Recall that a commutator between operators A and B is $[A, B] = AB - BA$. The commutator property is very useful in analyzing the quantum state that goes through a sequence of operators. For example, if A, B commute, then $AB|\psi\rangle = BA|\psi\rangle$. So, instead of analyzing $AB|\psi\rangle$, we can study $AB|\psi\rangle$. Further, if $||[A, B]||$ is small, then $AB|\psi\rangle$ and $BA|\psi\rangle$ will be very close in Euclidean distance. So, we can still reduce analyzing $AB|\psi\rangle$ to the analysis of $BA|\psi\rangle$ without losing much accuracy. The following are some identities on commutators.

Lemma 1. *Let $A, B, C \in \mathcal{L}(\mathcal{H})$. Then, the following holds:*

1. $[AB, C] = A[B, C] + [A, C]B$;
2. $[ABC, D] = AB[C, D] + A[B, D]C + [A, D]BC$;
3. $[A^n, B] = \sum_{i=0}^{n-1} A^i [A, B] A^{n-i-1}$.

The proof can be done by simple calculations. For example, $[AB, C] = ABC - CAB = ABC - ACB + ACB - CAB = A[B, C] + [A, C]B$. The other two can be proved using item 1 by noticing that $ABC = AB \cdot C$ and $A^n = A^{n-1} \cdot A$. The details are omitted.

The following notation of control register with respect to a basis will be useful to determine if two operators commute sometimes.

Definition 6. *Register D is a* **control register** *in the orthonormal basis $\{|y\rangle\}_y$ for operator B that operates on registers WD, if B can be written as $B = \sum_y B_y \otimes |y\rangle\langle y|_D$, where B_y operates on W.*

Remark 2. *Intuitively, if register D has y, then W will be applied by operator B_y, while register D is not changed. The requirement for a control register is very loose. Indeed, if B does not operate on D, then by default, it is understood as $B \otimes I_D = \sum_x B \otimes |x\rangle\langle x|_D$ for a basis $\{|x\rangle\}_x$, and so D is a control register for B.*

It is clear that if two operators operate on completely disjoint registers, then they commute. The following lemma states that this commutative property still holds even if they further share a common control register in the same basis.

Lemma 2. *Let XYD be three quantum registers. The following properties hold:*

1. *If A operates on XD, while B operates on YD with D being a control register in the same basis $\{|y\rangle\}_{y \in D}$ for both A and B, then $[A, B] = 0$.*
2. *If A is a projector on D in basis $\{|y\rangle\}_y$, and B operates on YD with D being a control register in the same basis, then $[A, B] = 0$.*

Proof. 1. Since A does not operate on Y, and B does not operate on X, we can write $A = \sum_y A_y \otimes I_Y \otimes |y\rangle\langle y|_D$ and $B = I_X \otimes B_y \otimes |y\rangle\langle y|_D$, with $\{|y\rangle\}_y$ being an ortonormal basis, where A_y operates on register X, and B_y operates on register Y. Thus, both AB and BA equal $\sum_y A_y \otimes B_y \otimes |y\rangle\langle y|$. The conclusion follows.

2. If $A = \sum_{y \in T} |y\rangle\langle y|_D$, and $B = \sum_y B_y \otimes |y\rangle\langle y|_D$, then AB and BA both equal to $\sum_{y \in T} B_y \otimes |y\rangle\langle y|_D$. Thus, $[A, B] = 0$. □

3.2. Properties of Norm

This section gives some simple properties of the operator or state norm. The following was stated in [32] without a proof. We give a proof for completeness.

Lemma 3. *Let $A, B, A_1, , A_2 \in \mathcal{L}(\mathcal{H})$. Then, the following holds.*

1. *If $A_1, A_2 \in \mathcal{L}(\mathcal{H})$, then $||A_1 \otimes A_2|| = ||A_1|| \cdot ||A_2||$.*
2. *If $A^\dagger B = 0$ and $AB^\dagger = 0$, then $||A + B|| \leq \max(||A||, ||B||)$. Especially, if $A = \sum_x |x\rangle\langle x| \otimes A^x$, then $||A|| \leq \max_x ||A^x||$.*

Proof. 1. By singular value decomposition, we can write $A_1 = U_1 D_1 V_1$ and $A_2 = U_2 D_2 V_2$ for $D_i = \mathtt{diag}(\mu_{i1}, \cdots, \mu_{it_i})$, with $\mu_{ij} \geq 0$ and unitary U_1, U_2, V_1, V_2. Then, $A_1 \otimes A_2 = (U_1 \otimes U_2)(D_1 \otimes D_2)(V_1 \otimes V_2)$. Hence, $||A_1 \otimes A_2|| = (\max_t \mu_{1t})(\max_j \mu_{2j}) = ||A_1|| \cdot ||A_2||$, as $U_1 \otimes U_2$ and $V_1 \otimes V_2$ are unitary.
2. By the singular value decomposition theorem, we can write $A = \sum_{i=1}^{s} \lambda_i |x_i\rangle\langle y_i|$ and $B = \sum_{i=1}^{t} \beta_i |u_i\rangle\langle v_i|$, where $\{|x_i\rangle\}_i, \{|y_i\rangle\}_i, \{|u_i\rangle\}_i, \{|v_i\rangle\}_i$ are, respectively, orthonormal sets of vectors in $\mathcal{H}$ and $\lambda_j, \beta_i > 0$. Then, from $A^\dagger B = 0$, we have $\sum_{i,j} \lambda_i^* \beta_j \langle x_i|u_j\rangle \cdot |y_i\rangle\langle v_j| = 0$. As $\langle y_i|A^\dagger B|v_j\rangle = 0$, we know that $\langle x_i|u_j\rangle = 0$ for $i = 1, \cdots, s$, and $j = 1, \cdots, t$.

Similarly, from $AB^\dagger = 0$, we have $\langle y_i | v_j \rangle = 0$. Hence, $\{|y_i\rangle\}_{i=1}^s, \{|v_i\rangle\}_{i=1}^t$ are disjoint and together orthonormal states. Together, they can be extended to an orthonormal basis. Let $|x\rangle$ be any normalized state represented under this basis with coordinate vector $(w_1, \cdots, w_n)$. Then, $(A+B)|x\rangle = \sum_{i=1}^s \lambda_i w_i |x_i\rangle + \sum_{j=1}^t \beta_j w_{s+j} |u_j\rangle$. Its norm is upper bounded by $\max_{ij}(|\lambda_i|, |\beta_j|) = \max(||A||, ||B||)$, which is desired! This result implies the second claim as $(|x\rangle\langle x| \otimes A^x)^\dagger (|y\rangle\langle y| \otimes A^y) = 0$ for any $x \neq y$. □

The following lemma (from Equation (9.100) [36]) builds the connection between Euclidean distance of pure states and their trace distance. We give a proof here for clarity.

Lemma 4. *Let $|u\rangle, |v\rangle$ be two states for a quantum system. $D_t(|u\rangle\langle u|, |v\rangle\langle v|) \leq |||u\rangle - |v\rangle||$.*

Proof. Let $|0\rangle = |u\rangle$ and take $|1\rangle$ as a unit orthogonal state of $|0\rangle$ so that $|v\rangle = \omega(\cos(\theta)|0\rangle + \sin(\theta)|1\rangle)$ with $\theta \in [0, \pi/2]$, by absorbing the complex unit factor (if any) into $|1\rangle$, where ω is a complex unit factor. By calculation, $D_t(|u\rangle\langle u|, |v\rangle\langle v|) = |\sin(\theta)|$. On the other hand, $|||u\rangle - |v\rangle|| = \sqrt{|1 - \omega\cos(\theta)|^2 + \sin^2(\theta)} \geq \sqrt{(1-\cos(\theta))^2 + \sin^2(\theta)} = 2|\sin(\theta/2)|$. Since $|\sin(\theta)| = 2|\sin(\theta/2) \cdot \cos(\theta/2)| \leq 2|\sin(\theta/2)|$, the result follows. □

3.3. Impact of Intermediate Measurement on the Final Output

In the quantum security analysis, it is very common that some intermediate measurements are performed during a quantum algorithm. It is useful to ask if these intermediate measurements affect the final algorithm output or not. The following lemma states that if an intermediate measurement is a projective measurement on a control register in the same basis as for the control register, then the final algorithm output will not be affected.

Lemma 5. *Let $|\psi\rangle = \sum_y t_y |\psi_y\rangle_X |y\rangle_Y$ be a joint state for register XY, with $\{|y\rangle\}_{y\in\mathcal{Y}}$ as the orthonormal basis of register Y. Let $P = \{|y\rangle\langle y|\}_y$ be the projective measurement on register Y. Let $Q = \{Q_x\}_x$ be the measurement on register X. Let U_y be a unitary on register X, which is labeled with $y \in \mathcal{Y}$. Consider procedure A: apply $\sum_{y\in\mathcal{Y}} U_y \otimes |y\rangle\langle y|$ to $|\psi\rangle$, and then apply measurement Q on X to output x. Also, consider procedure A', which starts with measurement P on Y and continues with procedure A, with the final output denoted by x'. Then, the distributions of x and x' are identical.*

Proof. Procedure A outputs x with probability $||\sum_y t_y Q_x U_y |\psi_y\rangle|y\rangle||^2$. The procedure A' outputs y, resulting in the collapsed state $U_y|\psi_y\rangle|y\rangle$ with probability $||t_y||^2$. Following the measurement Q, it outputs x with probability $||t_y Q_x U_y|\psi_y\rangle|y\rangle||^2$. So, the overall probability to output x with probability $\sum_y ||t_y Q_x U_y|\psi_y\rangle|y\rangle||^2 = ||\sum_y t_y Q_x U_y|\psi_y\rangle|y\rangle||^2$, as $\{|y\rangle\}_y$ is orthogonal, is desired. □

Remark 3. *In Lemma 5, it is important that projective measurement $P = \{|y\rangle\langle y|\}_y$ uses the same basis as $\{|y\rangle\}_y$ as in $\sum_y U_y \otimes |y\rangle\langle y|$. That is, the unitary needs to use register Y as a control register on the basis of the projective measurement P. Otherwise, the result will be incorrect. For example, let $|\psi\rangle = |0\rangle|+\rangle$, where $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, and $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. Define $U_+ = |1\rangle\langle 0| + |0\rangle\langle 1|$ and $U_- = I$. Let $Q = \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ on register X and $P = Q$ but on register Y. Let $U = U_+ \otimes |+\rangle\langle +| + U_- \otimes |-\rangle\langle -|$. Then, for procedure A, the state before measurement Q is $|1\rangle|+\rangle$, and hence, the outcome of Q is 1 with probability 1. But for procedure A', after measurement P, the state is $|0\rangle|1\rangle$ or $|0\rangle|0\rangle$, each with probability 1/2. Since $|1\rangle = \frac{|+\rangle-|-\rangle}{\sqrt{2}}$ and $|0\rangle = \frac{|+\rangle+|-\rangle}{\sqrt{2}}$, after applying U, the result is $\frac{1}{\sqrt{2}}(|1\rangle|+\rangle \pm |0\rangle|-\rangle)$ ($\pm$ depending 1 or 0 on Y register), and next, the measurement Q on register X gives the outcome 1 with probability $1/2 \cdot 1/2 + 1/2 \cdot 1/2 = 1/2$. This is different from the procedure A.*

The above example shows that an intermediate measurement could change the final output distribution. But, the following result states that the probabilities on the final output without such an intermediate measurement are actually related. This result was given by Boneh and Zhandry [38] (but the intermediate measurement M seemingly needs to be projective). For clarity, we give a proof here.

Lemma 6. *Let A be a quantum algorithm and $\Pr[x]$ be the probability that A outputs x. Let A' be the algorithm that runs A until some stage and then performs a projective measurement M, which gives an outcome m (out of k possible choices) and next continues the execution of A with the post-measurement state. Let $\Pr'[x]$ be the probability that A' outputs x. Then, $\Pr'[x] \geq \Pr[x]/k$.*

Proof. Let $M = \{M_i\}_{i=1}^k$ be the measurement. Let $|\phi\rangle$ be the state right before this measurement. Then, the probability of M giving outcome m occurs with probability $p_m = \langle\phi|M_m^* M_m|\phi\rangle$, and the post-measurement state is $|\phi_m\rangle = M_m|\phi\rangle/\sqrt{p_m}$. According to the deferred measurement principle, we can assume that A after this consists of a unitary U and a final projective measurement $\{P_i\}_i$. Then,

$$\Pr{}'[x] = \sum_m p_m \langle\phi_m|U^\dagger P_x^\dagger P_x U|\phi_m\rangle = \sum_m \langle\phi|M_m^\dagger U^\dagger P_x^\dagger P_x U M_m|\phi\rangle \tag{2}$$

$$= \sum_m ||P_x U M_m|\phi\rangle||^2 \geq ||\sum_m P_x U M_m|\phi\rangle||^2/k \tag{3}$$

$$= ||P_x U|\phi\rangle||^2/k = \Pr[x]/k. \tag{4}$$

where the inequality follows from Cauchy–Schwarz inequality, and Equation (4) uses the fact that M is the projective measurement, so $\sum_m M_m = \sum_m M_m^\dagger M_m = I$. □

3.4. Making Intermediate Measurement Unitaries

It is very common that a quantum algorithm will make intermediate measurements. A deferred measurement principle [36] states that we can move these measurements to the end of the algorithm (without affecting the output). From this principle, we only need to consider an algorithm that consists of a sequence of unitaries except for the final measurement. The following lemma and its corollary are essentially to capture this. We give a proof here, as it demonstrates how this can be made and it will be useful for us later to understand other results later. We start with a simpler version where the algorithm only has one intermediate measurement.

Lemma 7. *Let $|\phi\rangle$ be a quantum state. We apply the following operators on register A: first a unitary U, then a measurement $M = \{M_y\}_y$ that results in y, next a unitary V_y, and finally a measurement $N_y = \{N_{yx}\}_x$ that results in x. Then, there exist a unitary W on A and additional registers BC and a projective measurement P on C that results in x with the same probability.*

Proof. It is clear that procedure A outputs x with probability $\Pr(x) = \sum_y ||N_{yx} V_y M_y U|\phi\rangle||^2$. Then, define a unitary operator U_M so that $U_M|\phi\rangle_A|0\rangle_B = \sum_y M_y|\phi\rangle_A|y\rangle_B$ ([36], pp. 95). Also, define unitary V on AB with $V = \sum_y V_y \otimes |y\rangle\langle y|_B$. Also, define unitary U_N so that $U_N|u\rangle_A|y\rangle_B|0\rangle_C = \sum_x \sum_r (N_{rx} \otimes |r\rangle\langle r|)|u\rangle_A|y\rangle_B|x\rangle_C$. Finally, define P to be the projective measurement $P = \{|x\rangle\langle x|\}_x$. Then, consider $U_N V U_M U|\phi\rangle_A|0\rangle_B|0\rangle_C$, followed by P on C. Then, the probability of outcome x, by first applying $W = U_N V U_M$, followed by measurement P on C, is

$$||\sum_r (N_{rx} \otimes |r\rangle\langle r|_B) \cdot \sum_{y'} V_{y'} \otimes |y'\rangle\langle y'|_B \cdot \sum_y M_y|\phi\rangle_A|y\rangle_B|x\rangle_C||^2$$

$$= ||\sum_y N_{yx} V_y M_y U|\phi\rangle|y\rangle||^2$$

$$= \sum_y ||N_{yx} V_y M_y U|\phi\rangle||^2 = \Pr(x), \text{desired!}$$

□

Remark 4. *In this lemma, register B is a control register in the basis $\{|y\rangle_B\}_y$ for other operators; register C is a control register in the basis $\{|x\rangle_C\}_x$ for other operators. Hence, the projective measurement $\{|x\rangle\langle x|\}_x$ on B commutes with other operators and so can be moved to the end of the operations (especially after measurement P on C) and hence does not affect the distribution of outcome x of P; thus, it can be removed. This justifies the proof idea of the above lemma. With this in mind, the following generalization corollary of the lemma is straightforward.*

Corollary 1. *Let $|\phi\rangle$ be a quantum state of register A. For $\ell = 1, \cdots, N$, run a unitary U_ℓ, measurement $M_{y^{\ell-1}} = \{M_{y^\ell}\}_{y_\ell}$ that results in y_ℓ, followed by unitary V_{y^ℓ}, where y^i represents the sequence $y_1 \cdots y_i$. Finally, it applies measurement $N_{y^N} = \{N_{y^N x}\}_x$ that results in x. Then, there is unitary W and projective measurement P that apply to the initial state $|\phi\rangle|0\rangle_1 \cdots |0\rangle_N|0\rangle$, which results in x with the same probability.*

4. Quantum Random Oracles

In this section, we will introduce the quantum random oracles. As a convention in this paper, we use bold font to represent the random oracle (e.g., **RO**) and the italic font (e.g., *RO*) to represent the operator for the random oracle query. We distinguish an oracle and its operator, because some oracle could offer more operators.

We introduce the standard random oracle in Section 4.1. That is, this is the classical random oracle extended to the quantum setting. Then, we introduce Zhandry's compressed random oracle [31] (**CStO**) in Section 4.2, which allows a simulator to detect if an input x has been queried to the oracle or not. Next, we present in Section 4.3 our extension of **CStO**, called the compressed random oracle with adaptive special points (**CStO**$_s$), and its connection to **CStO**. Finally, we address in Section 4.4 how **CStO**$_s$ and **CStO** can be efficiently implemented.

4.1. Standard Random Oracle

In the random oracle model, a cryptographic hash function $H : \mathcal{X} \to \{0,1\}^n$ is treated as an external oracle so that whenever one needs to compute $H(x)$, he queries x to this oracle and receives $H(x)$. We assume that $\mathcal{X}$ has a finite bit length. The oracle uses a random function from $\mathcal{X}$ to $\mathcal{Y}$ to answer the queries. Let $\mathcal{X} = \{x_1, \cdots, x_N\}$ be an ordered set with $x_1 < x_2 < \cdots < x_N$. Function H can be represented by its truth table $H(x_1), H(x_2), \cdots, H(x_N)$. In the *quantum random oracle* model, H is represented by state $|H\rangle$ (using its truth table). An algorithm $\mathcal{A}$ can query a superposition to random oracle **RO**. For query $|x\rangle|y\rangle$, *RO* maps $|x\rangle|y\rangle|H\rangle$ to $|x\rangle|y \oplus H(x)\rangle|H\rangle$.

The *standard random oracle* **StO** has an initial state in a uniform superposition $\frac{1}{\sqrt{2^{n|\mathcal{X}|}}}\sum_H |H\rangle$. For query $|x\rangle|y\rangle$, *StO* maps $\frac{1}{\sqrt{2^{n|\mathcal{X}|}}}\sum_H |x\rangle|y\rangle|H\rangle$ to $\frac{1}{\sqrt{2^{n|\mathcal{X}|}}}\sum_H |x\rangle|y \oplus H(x)\rangle|H\rangle$. Notice that **RO** can be obtained from **StO** by starting with a projective measurement on an oracle register (resulting in $|H\rangle$). Even though **RO** and **StO** are different, no adversary can distinguish them. This can be seen from Lemma 2(2) by observing that oracle register is a control register in the computational basis for adversarial operators (which do not operate on oracle register) and *StO*. Hence, the projective measurement on the oracle register can be moved to after $\mathcal{A}$ makes the final measurement.

Fact 1. *Let $\mathcal{A}$ be a quantum algorithm with oracle access to the quantum random oracle. Then,* $\Pr(\mathcal{A}^{\mathbf{RO}}() = 1) = \Pr(\mathcal{A}^{\mathbf{StO}}() = 1)$.

4.2. Compressed Random Oracle

The *compressed* random oracle **CStO** was introduced in [31], and our exposition mainly follows [32]. It is a powerful tool for security proof in the quantum random oracle model (QROM). Let $\mathcal{Y} = \{0,1\}^n$ and $\bar{\mathcal{Y}} = \mathcal{Y} \cup \{\perp\}$. Let H be the quantum Walsh–Hadamard transform over $\mathbb{C}[\mathcal{Y}]$. Define $\phi_y = H|y\rangle$ for $y \in \{0,1\}^n$. Since $\{|y\rangle\}_{y\in\{0,1\}^n}$ is orthonormal, and $H^2 = I$, $\{|\phi_y\rangle\}_{y\in\{0,1\}^n}$ is orthonormal as well. Then, we define an unitary operator F over $\mathbb{C}[\bar{\mathcal{Y}}]$ such that

$$F|\bot\rangle = |\phi_0\rangle, \quad F|\phi_0\rangle = |\bot\rangle, \quad F|\phi_y\rangle = |\phi_y\rangle, \quad \forall y \in \mathcal{Y} - \{\mathbf{0}\}. \tag{5}$$

It is Hermitian (i.e., $F^\dagger = F$) because $F = |\phi_0\rangle\langle\bot| + |\bot\rangle\langle\phi_0| + \sum_{y\neq 0}|\phi_y\rangle\langle\phi_y|$. Furthermore, notice that $|y\rangle = 2^{-n/2}\sum_{\eta\in\{0,1\}^n}(-1)^{y\cdot\eta}|\phi_\eta\rangle$. This implies $F|y\rangle = |y\rangle + 2^{-n/2}(|\bot\rangle - |\phi_0\rangle)$.

We consider the multi-register $D = \{D_x\}_{x\in\mathcal{X}}$ for the random oracle, where D_x has a state space $\mathbb{C}[\bar{\mathcal{Y}}]$ spanned by the computational basis $\{|y\rangle\}_{y\in\mathcal{Y}} \cup \{|\bot\rangle\}$. The initial state of D is $\otimes_x|\bot\rangle_{D_x}$. We assume that the adversary has a query register X, a response register Y, and a work register W. To query the oracle, adversary provides XY registers to oracle, who then applies unitary

$$CStO_{XYD} = \sum_{x\in\mathcal{X}} |x\rangle\langle x|_X \otimes CStO_{YD_x} \tag{6}$$

on XYD, where $CStO_{YD_x} = F_{D_x}\cdot \text{CNOT}_{YD_x}\cdot F_{D_x}$, and $\text{CNOT}|y\rangle_Y|u\rangle_{D_x} = |y+u\rangle_Y|u\rangle_{D_x}$. This oracle has the property that if $|x\rangle$ has not been queried before, then D_x will remain as $|\bot\rangle_{D_x}$. Also, as shown in the following lemma by Zhandry [31], an (unbounded) attacker can not distanguish **StO** and **CStO**. We stress that this indistinguishability holds only if no operator other than $CStO$ (respectively, StO) is applied on D; otherwise, it might fail.

Lemma 8. *[31] Let $\mathcal{A}$ be a quantum algorithm with oracle access to the quantum random oracle. Then,* $\Pr(\mathcal{A}^{\mathbf{StO}}() = 1) = \Pr(\mathcal{A}^{\mathbf{CStO}}() = 1)$.

4.3. Compressed Random Oracle with Adaptive Special Points

CStO has the advantage that it can record oracle queries. But, it can not allow a simulator (as in a classical random oracle) to set the random oracle values for some special points. Liu and Zhandry [33] introduced **CStO** with non-adaptive special points to resolve this issue. However, it seems the Fiat–Shamir-based signature proof in their work seems to require adaptive special points, as the adversary's signing query cannot be guessed or predicted before the query. In this section, we formalize the compressed random oracle with adaptive special points (denoted by $\mathbf{CStO}_s$) as a natural generalization of **CStO**. It allows a simulator to set special points on the fly. But, this needs some considerations. We need to make it connected to **CStO**. For example, if an adversary, interacting with a challenger in the **CStO** model, has a success probability ϵ, we probably want it to have a success probability at $\epsilon/poly(\lambda)$ when interacting with the challenger in the $\mathbf{CStO}_s$ model. We need this, as in applications, we will have a game with **CStO**, and then we want to transit to a game with $\mathbf{CStO}_s$ with the adversary success probability degraded only by at most a polynomial fraction. Liu and Zhandry [33] introduced a random experiment (to be detailed in Section 7) to make the connection. In the adaptive case, it needs some care (in order to be compatible with the random experiment). In the following, we first describe our $\mathbf{CStO}_s$ and then outline this subtlety.

The $\mathbf{CStO}_s$ oracle initially has state $\otimes_x|\bot\rangle_{D_x}$. We maintain two initially empty sets Ξ_0 and Ξ_1 to record the special points at different stages. We also allow the oracle to abort after certain measurements, and the motivation will be discussed later. The oracle can be accessed through three types of queries below.

- *PointReg0 Query*. One can send a new point $x \in \mathcal{X}$ to the oracle. If $x \in \Xi_0 \cup \Xi_1$, it does nothing; otherwise, the oracle updates $\Xi_0 = \Xi_0 \cup \{x\}$.
- *Random Oracle Query.* One can issue a random oracle query by providing a query register X and a response register Y to the oracle. If this is the ith random oracle query, the oracle applies a projective measurement $\Lambda_i = (\Lambda_{i0}, \Lambda_{i1})$ in the computational basis to oracle register D_{Ξ_0} (Λ_i can be determined by i and some parameters that are

determined before the oracle starts). If the outcome is 1, it **aborts**; otherwise, it applies $CStO_s = \sum_{x\in\mathcal{X}} |x\rangle\langle x| \otimes CStO_{sYD_x}$ to the XYD registers, where

$$CStO_{sYD_x} = \begin{cases} CStO_{YD_x}, & x \notin \Xi_1 \\ \mathrm{CNOT}_{YD_x}, & \text{otherwise.} \end{cases}$$

Finally, it returns register XY.

- *PointReg1 Query.* One can send $x \in \Xi_0$ to the oracle. If $x \notin \Xi_0$, it does nothing. Otherwise, it measures D_x with $\Pi = (\Pi_0, \Pi_1)$, where $\Pi_0 = |\perp\rangle\langle\perp|, \Pi_1 = I - \Pi_0$. If the outcome is 1, it **aborts**; otherwise, it updates $|\perp\rangle_{D_x}$ with $|r\rangle$ for a random $r \in \mathcal{Y}$ (this can be done, as $|\perp\rangle_{D_x}$ is now classic; or, we can apply unitary $|\perp\rangle\langle r| + |r\rangle\langle\perp| + \sum_{v\in\mathcal{Y}-\{r\}} |v\rangle\langle v|$). Finally, it updates $\Xi_1 = \Xi_1 \cup \{x\}$ and $\Xi_0 = \Xi_0 - \{x\}$.

Remark 5. *It is time to justify this strange random oracle. It is in fact motivated by the requirements in the security proof. The main motivation is to find a modified random oracle so that the random experiment (with* **CStO***) in Section 7 can be easily converted into a game with this modified random oracle. We want to define this modified oracle with respect to this random experiment because adversary success in this experiment, in comparison with the original security game, is degraded only by a polynomial fraction. So, this compatible random oracle is denoted by* $\mathbf{CStO}_s$*. Here the compatibility means that, given the random experiment with* **CStO***, we can easily transit to a game with* $\mathbf{CStO}_s$ *that preserves the adversary success probability. Further remarks on the definition of* $\mathbf{CStO}_s$ *are as follows:*

- *In the classical random oracle, a simulator can set the random oracle values of special queries to his own choices. In the* $\mathbf{CStO}_s$*, a special point will be first recorded in* Ξ_0 *and later set to a planned value (when a PointReg1 query on this point is issued). We handle special points in two stages for technical reasons (See the remark after Theorem 5) only. Essentially, if we define the random oracle value of a special point early (e.g., at the time of adding into* Ξ_0*), it could make the previously selected experiment change to a different one.*
- $\mathbf{CStO}_s$ *is to formulate the random experiment in Section 7 as a well-defined random oracle model. Especially, measurement* Λ_i *in a random oracle query is to make sure the interaction with oracle follows the restriction of the selected experiment. If the measurement outcome is 1, it indicates that the game is not consistent with the selected experiment and hence can stop now; otherwise, it continues. This randomly selected but consistent experiment can guarantee the adversary to have a good success probability compared to the original game.*
- *In the classical random oracle, a simulator can pay attention to each query to make sure that each special point is not queried before it is set to the designated value. In the quantum setting, recording each query is difficult, as one can query* $\frac{1}{|\mathcal{X}|}\sum_x |x\rangle_X |0\rangle_Y$*, which indicates that every* x *is actually queried. To overcome this, we need to confirm that* $RO(x)$ *is not defined by measurement* Π *on* D_x*. If the measurement is successful, then* D_x *will have* $|\perp\rangle_{D_x}$ *now, the non-*$\perp$ *components in the superposition are pruned, and we can define the random oracle value for this* x*; if the measurement fails, we have no way to set the random oracle value for* x *and so abort. This is why we abort in* PointReg1 *when the measurement outcome is 1.*

One might hope that an attacker cannot distinguish **CStO** and $\mathbf{CStO}_s$. However, this is not true, as the latter simply has different interfaces. However, we can define a variant of **CStO** to achieve this indistinguishability as long as the abortion event does not occur.

Precisely, we define $\mathbf{CStO}'$ to be a variant of $\mathbf{CStO}_s$ so that $CStO_s$ in the random oracle query is replaced by $CStO$ and also in PointReg1 query in case of the measurement outcome 0, where it leaves $|\perp\rangle_{D_x}$ as it is (instead of replacing it by $|r\rangle$). Essentially, $\mathbf{CStO}'$ has three interfaces as in $\mathbf{CStO}_s$, but the random oracle query uses **CStO** (after the measurement Λ_i with outcome 0), and the PointReg1 query only makes Π measurements on D.

The following lemma shows that $\mathbf{CStO}_s$ is perfectly indistinguishable from $\mathbf{CStO}'$, which is conditional on that the abort event in the oracle does not occur.

Lemma 9. *Let $\mathcal{A}$ be a quantum algorithm with access to a quantum random oracle and* abort *be the oracle abortion event. Then,*

$$\Pr(\mathcal{A}^{\mathbf{CStO}'}() = 1 \wedge \neg\mathsf{abort}) = \Pr(\mathcal{A}^{\mathbf{CStO}_s}() = 1 \wedge \neg\mathsf{abort}). \tag{7}$$

Proof. We use the hybrid argument with a variant $\mathbf{CStO}'_s$ of $\mathbf{CStO}_s$ to bridge $\mathbf{CStO}_s$ and $\mathbf{CStO}'$.

Oracle $\mathbf{CStO}'_s$. We modify $\mathbf{CStO}_s$ to $\mathbf{CStO}'_s$ so that upon PointReg1 query x with D_x measured with outcome 0 (i.e., $|\perp\rangle$), it updates $|\mathbf{y}\rangle_D$ to $\frac{1}{2^{n/2}}\sum_r |\mathbf{y} \cup (r)_x\rangle_D$ (instead of $|\mathbf{y} \cup (r)_x\rangle_D$ for a random r), where $\mathbf{y} \cup (r)_x$ (which is well defined as $y_x = \perp$) is the vector with $y_{x'}$ at index $x' \neq x$ and r at index x. Notice that right after this, $x \in \Xi_1$. Furthermore, D_x for this x is a *control register* (Definition 6) in the computational basis for adversary operations, $\Pi_0, \Pi_1, \Lambda_{i0}, \Lambda_{i1}$ and $CStO_{sYD_u}$. To see this, it suffices to check $CStO_{sYD_x}$ only, as the other cases are clear (e.g., $CStO_{sYD_u}$ for $u \neq x$ does not operate on D_x at all). Since $x \in \Xi_1$, we know that $CStO_{YD_x} = \mathrm{CNOT}_{YD_x}$, which obviously can be written as a format of $\sum_{y \in \tilde{\mathcal{Y}}} B_y \otimes |y\rangle\langle y|_{D_x}$. Furthermore, $\mathbf{CStO}_s$ is obtained from $\mathbf{CStO}'_s$ by projective measurement on D_x in the computational basis for every $x \in \Xi_1$ (right after x is put in Ξ_1). By Lemma 2 (2), the projective measurement on D_x can be moved to the end of the interaction (after $\mathcal{A}$ outputs). Thus, the output of $\mathcal{A}$ with access to $\mathbf{CStO}'_s$ is the same as with access to $\mathbf{CStO}_s$.

Oracle $\mathbf{CStO}'$. We show that under the event $\neg\mathsf{abort}$, if the final (unnormalized) state after interacting with $\mathbf{CStO}'_s$ is $|\psi\rangle$, then the final state (unnormalized) after interacting with $\mathbf{CStO}'$ will be $F_{D_{\Xi_1}}|\psi\rangle$. This can be shown by induction on the query. It is correct initially, as $\Xi_1 = \emptyset$ initially, and hence $F_{D_{\Xi_1}}$ is its identity. Then, if it is correct after query $i-1$, consider query i. Before query i, $\mathcal{A}$ will operate on XYW registers (for simplicity, assume it is a unitary). But, since the adversary does not operate on D, the induction assumption on query $i-1$ implies the following: if the state right before query i (when interacting with $\mathbf{CStO}'_s$) is $|\psi\rangle$, then the state right before query i (when interacting with $\mathbf{CStO}'$) will be $F_{D_{\Xi_1}}|\psi\rangle$. Let us consider their relation *after* query i, which has three cases.

If query i is a PointReg0 query, then the claim still holds after the query, as no operation on the quantum state is executed.

If query i is a PointReg1 query x, then it suffices to consider $x \in \Xi_0$. Since $x \notin \Xi_1$ and the outcome of Π is 0 (otherwise, abort occurs in contradiction to the probability condition), so x will be added to Ξ_1, and the conclusion holds after the query as $F|\perp\rangle = |\phi_0\rangle$ (while, after the query, D_x in case of $\mathbf{CStO}'_s$ will have $|\perp\rangle$ and D_x in the case that $\mathbf{CStO}'$ will yield $|\phi_0\rangle$).

If query i is a random oracle query, we show that the induction still holds. First, $[F_{D_{\Xi_1}}, \Lambda_{ib}] = 0$ for both $b = 0, 1$, as Λ_i only operates on register D_{Ξ_0}. Thus, after the measurement (with the same outcome), the relation still holds. Second, the relation still holds after operator $CStO_s$ (in the case of $\mathbf{CStO}'_s$) and operator $CStO$ (in the case of $\mathbf{CStO}'$): for query $|x\rangle_X|y\rangle_Y$ with $x \notin \Xi_1$, both oracles use $CStO_{YD_x}$ to respond, and hence, their states after the query maintain the same relation (as D_{Ξ_1} is untouched); for query $|x\rangle_X|y\rangle_Y$ with $x \in \Xi_1$, $\mathbf{CStO}'$ uses $CStO_{YD_x}$, and $\mathbf{CStO}'_s$ uses CNOT_{YD_x}, but the two applications of F_{D_x} in $CStO_{YD_x}$ will cancel out. So, after the query, the relation still holds. The induction holds too.

Let $|\psi\rangle$ be the final unnormalized state under $\neg\mathsf{abort}$ and the final measurement of $\mathcal{A}$ be (P_0, P_1), with P_1 corresponding to outcome 1. Then, $\Pr(\mathcal{A}^{\mathbf{CStO}'_s}() = 1 \wedge \neg\mathsf{abort})$ is $||P_1|\psi\rangle||^2$, while $\Pr(\mathcal{A}^{\mathbf{CStO}'}() = 1 \wedge \neg\mathsf{abort})$ is $||P_1 \cdot \mathsf{F}_{D_{\Xi_1}}|\psi\rangle||^2$. However, $||P_1 \cdot \mathsf{F}_{D_{\Xi_1}}|\psi\rangle||^2 = ||P_1|\psi\rangle||^2$, as $F_{D_{\Xi_1}}$ commutes with P_1 (since they operate on disjoint registers) and $F^2 = I$. □

The following lemma essentially states that if x^* has large min-entropy and we measure D_{x^*} of the adversary–oracle joint state, then, with high probability, the post-measurement state with outcome $\perp$ is close to the original state.

Lemma 10. *Let the current adversary–oracle joint state be* $|\psi\rangle = \sum_{z\mathbf{y}} \lambda_{z\mathbf{y}} |z\rangle|\mathbf{y}\rangle_D$ *after* q *queries to* **CStO**$_s$ *(or* **CStO***). Let* $|\psi_x\rangle = \sum_{z\mathbf{y}:\ y_x=\perp} \lambda_{z\mathbf{y}}|z\rangle|\mathbf{y}\rangle_D$*, and* x^* *is a random variable over* $\mathcal{X}$ *with a min-entropy of at least* μ*. Then, with probability* $1-2^{-\mu/2}$ *(over* x^**),* $|||\psi\rangle - |\psi_{x^*}\rangle|| \leq q^{1/2}2^{-\mu/4}$*.*

Proof. Let $|\psi'_x\rangle = \sum_{z\mathbf{y}:y_x\neq\perp} \lambda_{z\mathbf{y}}|z\rangle|\mathbf{y}\rangle_D$. Then, $|\psi\rangle = |\psi'_x\rangle + |\psi_x\rangle$. Consider $L := \sum_x |||\psi'_x\rangle||^2$. Let $N_\mathbf{y}$ be the number of x so that $y_x \neq\perp$ in $\mathbf{y}$. Then, given $\mathbf{y}$, $|\mathbf{y}\rangle$ appears in $|\psi'_x\rangle$ for exactly $N_\mathbf{y}$ possible xs. Thus, $L = \sum_{z\mathbf{y}} |\lambda_{z\mathbf{y}}|^2 N_\mathbf{y}$. Since each $\mathbf{y}$ in $|\psi\rangle$ has at most q possible non-$\perp$ entries, it follows that $N_\mathbf{y} \leq q$, and hence, $L \leq q$. Hence, there are at most $2^{\mu/2}$ choices for x so that $|||\psi'_x\rangle|| \geq q^{1/2}2^{-\mu/4}$. Since x^* has min-entropy μ, we have that $|||\psi'_{x^*}\rangle|| < q^{1/2}2^{-\mu/4}$ with a probability of at least $1-2^{-\mu/2}$. The lemma follows. □

4.4. Efficient Encoding of **CStO** *and* **CStO**$_s$

Notice that, so far, the oracle state is represented via basis states $|\mathbf{y}\rangle_D \in \bar{\mathcal{Y}}^{\mathcal{X}}$ with at most q non-$\perp$ entries. However, we need to show how the operators used so far can be efficiently implemented. Zhandry [31] showed how to efficiently encode and compute O_{XYD}. In our work, more operators on D are introduced. It is necessary to show that Zhandry's encoding can be extended. In Appendix B, we detail how these operators can be efficiently executed on the encoded oracle state.

5. Relation Measurement in **CStO**$_s$

In this section, we want to measure if the record in register D of **CStO**$_s$ satisfies some relation R. In applications, this R could be some properties of a simulator's interest. Thus, a successful measurement implies a detection of satisfaction of such a property. In Section 5.1, we introduce a unitary operator U_R that writes the smallest input x_i satisfying property R into a new register P and show that the commutator norm $||[CStO_s, U_R]||$ is small. With this, we can later reduce the analysis of $CStO_s \cdot U_R|\psi\rangle$ to that of $U_R \cdot CStO_s|\psi\rangle$, without worrying about the difference. In Section 5.2, we give an upper bound on the probability that relation R is satisfied in the record of **CStO**$_s$ after q random oracle queries.

5.1. Relation Measurement

Given a record $|\mathbf{y}\rangle_D$, we sometimes are interested in checking if there exists y_x in $\mathbf{y}$ so that (x, y) satisfies a certain property. In this section, we show how to measure such a property, where the property will be represented by a relation. Don et al. [32] has studied this in the **CStO** setting. Our exposition is to present it in alternative and seemingly simpler way and looks at the norm of its commutator with **CStO**$_s$.

Let $R \subset \mathcal{X}\times\mathcal{Y}$ be a fixed and efficiently verifiable relation with $R(x,y) = 1$ if and only if $(x,y)\in R$. Especially, $R(x,y) = 0$ for any $(x,y)\notin \mathcal{X}\times\mathcal{Y}$. We assume that $0\notin\mathcal{X}$, and so $R(0,y) = 0$. Furthermore, $R(x,\perp) = 0$ as $\perp\notin\mathcal{Y}$. Let $\bar{\mathcal{X}} = \mathcal{X}\cup\{0\}$. We define function $f_R : \bar{\mathcal{Y}}^{|\mathcal{X}|} \to \bar{\mathcal{X}}$ so that

$$f_R(y_1,\cdots,y_N) = \begin{cases} x_i, & (x_j,y_j)\notin R \text{ for } j<i \text{ but } (x_i,y_i)\in\ R \\ 0, & i \text{ does not exist.} \end{cases}$$

where $\mathcal{X} = \{x_1,\cdots,x_N\}$ is an ordered set with $x_1 < x_2 < \cdots < x_N$. In other words, $f_R(y_1,\cdots,y_N)$ is the smallest x_i so that $(x_i,y_i)\in R$. It is easy to verify that

$$f_R(y_1,\cdots,y_{|\mathcal{X}|}) = \sum_{i=1}^{|\mathcal{X}|} x_i\cdot\bar{R}(x_1,y_1)\cdot\ldots\cdot\bar{R}(x_{i-1},y_{i-1})\cdot R(x_i,y_i). \quad (8)$$

Here, we emphasize that we do not require $\bar{\mathcal{X}}$ itself to be a group, but we implicitly assume that it can be regarded as a subset of an Abelian group $\tilde{\mathcal{X}}$ (e.g., $\bar{\mathcal{X}} = \{0,1,2,4\}$ can be regarded as a subset of $\mathbb{Z}_5$). Next, we define U_R to be a unitary on $\mathbb{C}[\bar{\mathcal{Y}}]^{\otimes \mathcal{X}} \otimes \mathbb{C}[\tilde{\mathcal{X}}]$ for register DP so that

$$U_R|\mathbf{y}\rangle_D|w\rangle_P = |\mathbf{y}\rangle_D|w + f_R(y_1, \cdots, y_{|\mathcal{X}|})\rangle_P, \tag{9}$$

where $|\mathbf{y}\rangle_D := |y_1\rangle_{D_{x_1}} \cdots |y_{|\mathcal{X}|}\rangle_{D_{x_{|\mathcal{X}|}}}$. Let

$$\Gamma_R = \max_x |\{y \mid (x, y) \in R\}| \text{ and } \Gamma_x = |\{y \mid (x, y) \in R\}|. \tag{10}$$

Notice that our U_R is an alternative specification, but it is identical to U_R in [32]. The following lemma was proved in [32] (we can obtain the same bound by a proof for our specification).

Lemma 11. *For any $x \in \mathcal{X}$, $||[F_{D_x}, U_R]|| \le 4\sqrt{2\Gamma_R/2^n}$.*

Lemma 12. $[\mathrm{CNOT}_{XYD}, U_R] = 0$.

Proof. It can be seen that $\mathrm{CNOT}_{XYD} = \sum_{\mathbf{y}}(\sum_{x,y} |x, y_x + y\rangle\langle x, y|) \otimes |\mathbf{y}\rangle\langle\mathbf{y}|_D$ and also that $U_R = \sum_{\mathbf{y}}(\sum_w |w + f_R(\mathbf{y})\rangle\langle w|_P) \otimes |\mathbf{y}\rangle\langle\mathbf{y}|_D$. Therefore, D is a control register for U_R and CNOT_{XYD} in the computational basis. According to Lemma 2(1), they commute. □

Theorem 1. $||[CStO_s, U_R]|| \le 8 \cdot 2^{-n/2}\sqrt{2\Gamma_R}$.

Proof. Notice that $CStO_s = \sum_{x\in\mathcal{X}} |x\rangle\langle x| \otimes CStO_{sYD_x}$ and for $x \in \Xi_1$, $CStO_{sYD_x} = \mathrm{CNOT}_{YD_x}$. Hence, according to Lemma 12, $[CStO_s, U_R] = \sum_{x\notin\Xi_1} |x\rangle\langle x|_X \otimes [F_{D_x} \otimes \mathrm{CNOT}_{YD_x} \otimes F_{D_x}, U_R]$, where we also use $[|x\rangle\langle x|_X, U_R] = 0$. By Lemmas 1 (3) and 3 (2),

$$||[CStO_s, U_R]|| \le 2\max_i ||[F_{D_{x_i}}, U_R]|| + ||[\mathrm{CNOT}, U_R]||.$$

By Lemmas 11 and 12, the result follows. □

5.2. Bounding the Probability for Relation Search Through Oracle Queries

We are interested in checking whether a relation R is satisfied (i.e., $R(x, y_x) = 1$ for some x) in the oracle register D after oracle queries. The following lemma upper bounds this probability. The proof idea is that $R(x, y_x) = 1$ can be detected by applying U_R and measuring the P register with outcome $\hat{x} \neq 0$. If we apply U_R and measure P at the beginning of the interaction, then $\hat{x} = 0$, because the initial oracle state is a dummy. Hence, the success probability at the end of interaction is obtained by sequentially switching the order of U_R and the operators during the interactions, as well as by looking at the norm of the commutator of these operators with U_R.

Lemma 13. *Let $\mathcal{A}$ be a quantum algorithm with access to $\mathbf{CStO}_s$, incurring L_0 random oracle queries and $q - L_0$ PointReg1 queries. The final state goes through U_R of relation R and a projective measurement on register P in the computational basis with outcome $\hat{x} \in \bar{\mathcal{X}}$. Then,*

$$\Pr(\hat{x} \neq 0 \wedge \neg\mathsf{abort}) \le 128q^2\Gamma_R/2^n. \tag{11}$$

Proof. Let $|\psi\rangle$ be the initial state of $\mathcal{A}$ with registers XYZ. The joint initial state with oracle is then $|\omega_0\rangle = |\psi\rangle_{XYZ} \otimes (\otimes_x|\perp\rangle_{D_x}) \otimes |0\rangle_P$ (after register P is added). Then, $\mathcal{A}$ has access to $\mathbf{CStO}_s$, incurring L_0 random oracle queries with intermediate operator V_{XYZ}, where, for simplicity, we assume that V_{XYZ} remains unchanged throughout the game. Finally, oracle applies U_R on DP and projective measurement $\mathbf{P}$ on P, outputting the outcome $\hat{x}$. The final state before measurement $\mathbf{P}$ is $|\omega\rangle = U_R(V \cdot \mathbf{CStO}_s)^L|\omega_0\rangle$ for some L, where $\mathbf{CStO}_s$ is the PointReg0 query or PointReg1 query or random oracle query. If the query is PointReg0, it does not operate on the state and so commutes with U_R; if it is PointReg1, then we only consider the case $x \in \Xi_0$. Under $\neg$abort, it consists of projector Π_0 and $U_{\perp,r} = |r\rangle\langle\perp| + |\perp\rangle\langle r| + \sum_{v \neq r} |v\rangle\langle v|$ for uniformly random r over $\mathcal{Y}$. We notice that $[\Pi_0, U_R] = 0$ by Lemma 2(2). Furthermore, it is not hard to verify that $U_{\perp,r}\Pi_0$ in PointReg1 commutes with U_R if $(x,r) \notin R$ (as $(x,\perp) \notin R$). If it is a random oracle query, we notice that $[\Lambda_i, U_R] = 0$, as D is the control register for both Λ_i and U_R in the computational basis. Therefore,

$$\begin{aligned}
&\Pr(\hat{x} \neq 0 \wedge \neg\mathsf{abort})\\
\leq\ & \mathbf{E}_{\mathbf{r}}(||(I - |0\rangle\langle 0|_P)|\omega\rangle||^2) \qquad /*\ r\text{'s from PointReg1; state } |\omega\rangle \text{ is consistent with } \neg\mathsf{abort} */\\
=\ & \mathbf{E}_{\mathbf{r}}(||(I - |0\rangle\langle 0|_P)[U_R, (V \cdot \mathbf{CStO}_s)^L]|\omega_0\rangle + (I - |0\rangle\langle 0|_P)(V \cdot \mathbf{CStO}_s)^L U_R|\omega_0\rangle||^2)\\
& /* \ \mathbf{CStO}_s \text{ requires the operator for measurement outcome (e.g., } \Pi_0, \Lambda_{i0}) \text{ is consistent with } \neg\mathsf{abort}*/\\
=\ & \mathbf{E}_{\mathbf{r}}(||(I - |0\rangle\langle 0|_P)[U_R, (V \cdot \mathbf{CStO}_s)^L]|\omega_0\rangle||^2)\\
& /* \text{ as } V \text{ and } \mathbf{CStO}_s \text{ do not operate on } P\text{, and so part 2 has } |0\rangle_P \text{ before applying } I - |0\rangle\langle 0|*/\\
\leq\ & \mathbf{E}_{\mathbf{r}}(||[U_R, (V \cdot \mathbf{CStO})^L]||^2) \leq \mathbf{E}_{\mathbf{r}}\{(L_0||[U_R, CStO_s]|| + \sum_i ||[U_R, U_{\perp,r_i}]||)^2\}\\
& /* \text{ Lemma 1(3), } [\Lambda_i, U_R] = [\Pi_0, U_R] = [V, U_R] = 0 \text{ and } L_0 \text{ are } \sharp \text{ of } CStO_s \text{ queries,}\\
& \quad \text{and } r_i \text{ corresponds to } r \text{ in the } i\text{th PointReg1 query. }*/\\
\leq\ & \mathbf{E}_{\mathbf{r}}\{(8L_0 \cdot 2^{-n/2}\sqrt{2\Gamma_R} + 2N_{\mathbf{r}})^2\}. \quad /* \text{ using Theorem 1 } */\\
& /*\ N_{\mathbf{r}} \text{ is the number of } r_i \text{ in } i\text{th PointReg1}(x_i) \text{ so that } (x_i, r_i) \in R */\\
& /*\ [U_R, U_{\perp,r}] = 0 \text{ for } (x,r) \notin R;\ ||[U_R, U_{\perp,r}]|| \leq 2 \text{ as } ||U_R|| = ||U_{\perp,r}|| = 1*/\\
\leq\ & 128q^2\Gamma_R/2^n,
\end{aligned}$$

where the last inequality follows from the calculation with the observation: $N_{\mathbf{r}}$ is the result of a Bernouli trial with probability $\Gamma_R/2^n$ for $q - L_0$ times; $\mathbf{E}(a + N_{\mathbf{r}})^2 = \mathrm{Var}(N_{\mathbf{r}}) + [a + \mathbf{E}(N_{\mathbf{r}})]^2$; $\mathrm{Var}(N_{\mathbf{r}}) = (q - L_0)\Gamma_R/2^n(1 - \Gamma_R/2^n)$ and $\mathbf{E}(N_{\mathbf{r}}) = (q - L_0)\Gamma_R/2^n$. The lemma follows. □

6. Query Extraction for $\mathbf{CStO}_s$

In a classical random oracle model, given $t = f(x, RO(x))$ for a fixed function f, a simulator can easily extract x by searching through the adversary's oracle query history. In the quantum setting, this strategy is not useful, as an attacker could query to an oracle in a superposition that includes x as one component. So generally, it is not clear how we can extract x without destroying the quantum state. In this section, we will show that this extraction is possible, and also, we make the extraction on the fly (i.e., right after t is given). This is an extension of Don et al. [32] from the **CStO** setting to the $\mathbf{CStO}_s$ setting.

This section is organized as follows. In Section 6.1, we present the simulation of $\mathbf{CStO}_s$ with an extraction interface. In Section 6.2, we show that if the extraction is conducted at the end of game, then the extraction is correct. In Section 6.3, we show that if we extract on the fly, then the extraction is still correct and the output is not disturbed. This last property is obtained from the extraction at the end of the game by observing that $\mathbf{CStO}_s$ almost commutes with the unitary measurement U_R (with high probability), and so we can move U_R gradually to the location where the attacker outputs the commitment t (to be extracted) without significantly disturbing the quantum state.

6.1. Simulating $\mathbf{CStO}_s$ with Extraction

In this section, we adapt the simulation of **CStO** with the extraction capability in [32] to the $\mathbf{CStO}_s$ setting. Essentially, the simulator simulates the oracle and also provides an interface for extracting the attacker's oracle query x that, together with y in D_x, is a witness of a target "*commitment*". Let $\theta(x,y)$ be an arbitrary but fixed function from $\mathcal{X} \times \mathcal{Y}$ to $\mathcal{T}$. For $t \in \mathcal{T}$, define relation $R_t = \{(x,y) \mid \theta(x,y) = t\}$, and U_t denotes unitary U_{R_t}. Then, the simulator is described in Figure 2.

- **Initialization**. The initial state for D is $\otimes_x|\perp\rangle_{D_x}$ and set $\Xi_0 = \Xi_1 = \emptyset$.
- **PointReg0 Query** $\mathcal{S}.PR_0$. Upon $x \in \mathcal{X}$, if $x \in \Xi_0 \cup \Xi_1$, it does nothing; otherwise, update $\Xi_0 = \Xi_0 \cup \{x\}$.
- **PointReg1 Query** $\mathcal{S}.PR_1$. Upon $x \in \mathcal{X}$, if $x \notin \Xi_0$, it does nothing; otherwise, it applies Π to register D_x. For outcome 1, it aborts; for outcome 0, it replaces $|\perp\rangle_{D_x}$ with $|r\rangle_{D_x}$ for a random $r \in \mathcal{Y}$ and finally updates $\Xi_0 = \Xi_0 - \{x\}$ and $\Xi_1 = \Xi_1 \cup \{x\}$.
- **Random Oracle Query** $\mathcal{S}.RO$. Upon the ith random oracle query with register XY, $\mathcal{S}$ applies a measurement Λ_i to register D_{Ξ_0}. For outcome 1, it aborts; for outcome 0, it applies $CStO_s$ to XYD. Finally, it returns register XY.
- **Extraction** $\mathcal{S}.E$. Upon a classical extraction query t, $\mathcal{S}$ applies unitary U_t to registers DP and projective measurement $\{|x\rangle\langle x|\}_{x\in\bar{\mathcal{X}}}$ to register P and returns outcome $\hat{x}$.

Figure 2. Simulator $\mathcal{S}$.

In the following two subsections, we prove that if $\mathcal{A}$ uses x and $y = RO(x)$ to generate t, then the extracted $\hat{x}$ from $\mathcal{S}.E(t)$ will equal to x. This is useful in a security proof where an attacker generates an output and we need to find out the witness of this output. We first prove a weaker version of this: if $\hat{x}$ is extracted at the end of game, the claim is true. Then, we extend to the case that $\hat{x}$ is extracted on the fly (i.e., right after $\mathcal{A}$ outputs t).

6.2. Extraction at the End of Game

We begin with a *collision* event in a computational basis $|\mathbf{y}\rangle_D$ in the oracle state with respect to function f in the sense that $f(x, y_x) = f(x', y_{x'})$ for some $x' \neq x$. We give a result that says that after q oracle queries, the probability of collision in the oracle is small. This is extended from [31] Theorem 2 in the setting of **CStO** to **CStO**$_s$; see Appendix C for a proof.

Lemma 14. *Let $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{T}$. Then, for any quantum algorithm $\mathcal{A}$ with access to* **CStO**$_s$, *incurring q oracle queries of either PointReg1 or random oracle,*

$$\Pr(\mathsf{col} \wedge \neg\mathsf{abort}) \leq 16q^3\Gamma_f/2^n, \tag{12}$$

where col *is the collision event in the final state ρ_q, and $\Gamma_f = \max_{x'\neq x, y'} |\{y \mid f(x,y) = f(x',y')\}|$.*

Now, we give an extraction theorem, where $\hat{x}$ is extracted at the end of oracle access. It states that if attacker computes t from x so that $t = f(x, RO(x))$, then $\mathcal{S}.E(t)$ at the end of game will most likely have $\hat{x} = x$. The idea is as follows. Assume $\hat{x} \neq x$. After an attacker's oracle access to **CStO**$_s$, we apply a classical oracle query on x with the result y_x. Assume that this state (right before $\mathcal{S}.E(t)$) is $\sum_{\mathbf{y}:\ y_x \text{ fixed}} \lambda_{\mathbf{y}} |\omega_{\mathbf{y}}\rangle |\mathbf{y}\rangle_{D_{\mathcal{X}-\{x\}}} F|y_x\rangle_{D_x}|0\rangle_P$. Furthermore, notice that $F|y_x\rangle = |y_x\rangle + |\delta\rangle$. If $\mathbf{y}$ in the sum leads to a measurement outcome $\hat{x}$ on register P (i.e., after $\mathcal{S}.E(t)$), then it has a collision (since $f(\hat{x}, y_{\hat{x}}) = t = f(x, y_x)$). This probability is small (by Lemma 14), and we can ignore it. If $|\mathbf{y}\rangle_{D_{\mathcal{X}-\{x\}}}|y'_x\rangle$ for $y'_x \neq y_x$ under $\mathcal{S}.E(t)$ gives $\hat{x}$, then y'_x must come from δ. However, $||\delta||$ is very small. So, this is unlikely too. This idea is from [32] Prop 4.5 in the $CStO$ case and can be generalized to prove the case of a vector $(\mathbf{t}, \mathbf{x})$.

Theorem 2. *Consider a quantum algorithm $\mathcal{A}$ with access to $\mathcal{S}$ (via interfaces other than $\mathcal{S}.E$), including q random oracle queries or PointReg1 queries and outputting $\mathbf{t} \in \mathcal{T}^\ell$ and $\mathbf{x} \in \mathcal{X}^\ell$. Let h_i be the output for an additional classical query x_i to $\mathcal{S}.RO$ and $\hat{x}_i = \mathcal{S}.E(t_i)$. Then,*

$$\Pr(\exists i : x_i \neq \hat{x}_i, f(x_i, h_i) = t_i \wedge \neg\mathsf{abort}) \leq 2^{-n+1}\ell + 16(q+\ell)^3\Gamma_f/2^n. \tag{13}$$

Proof. Let the adversary–oracle joint state be $|\psi_0\rangle$ after queries to $\mathcal{S}$ (including q random oracle queries or PointReg1 queries). In the following, we always assume that random the oracle query does not abort. Then, $\mathcal{A}$ measures and outputs $\mathbf{t}, \mathbf{x}$. Each x_i is then *classically* queried to $\mathcal{S}.RO$ and results in a joint state $|\psi_1\rangle$. We assume that $\mathbf{x} \cap \Xi_1 = \emptyset$ (the other case is similar). Hence, $|\psi_1\rangle$ can be written as $|\psi_1\rangle = |\mathbf{r}\rangle_{D_{\Xi_1}} \otimes F_{D_{\mathbf{x}}}|\mathbf{h}\rangle_{D_{\mathbf{x}}} \otimes \sum_{\omega\mathbf{u}:\ \mathbf{u}\in\bar{\mathcal{Y}}^A} \lambda_{\omega\mathbf{u}}|\omega\rangle_{XYZ}|\mathbf{u}\rangle_{D_A}$, where $\Xi_1 \cup \mathbf{x} \cup A$ is a decomposition of $\mathcal{X}$.

Finally, it applies the projective measurement $\Pi_D = \{|\mathbf{y}\rangle\langle\mathbf{y}|\}_{\mathbf{y}\in\bar{\mathcal{Y}}^{\mathcal{X}}}$ in the computational basis on D and applies $U_{t_i}, i = 1, \cdots, \ell$ followed by (projective) measurement on register P, as well as measurement $(\Pi_{col}, I - \Pi_{col})$ to the resulting state (assuming the collision measurement writes the result in a new register C), where Π_{col} is a projection into a space spanned by $|\mathbf{y}\rangle_D$, with $\mathbf{y} \in \bar{\mathcal{Y}}^{\mathcal{X}}$ satisfying $f(x, y_x) = f(x', y_{x'})$ for some $x' \neq x$ and $y_x, y_{x'} \in \mathcal{Y}$. Notice that D is a control register in the computational basis for $\Pi_D, \mathbf{P}U_{t_i}$, and collision measurement, where $\mathbf{P}$ is the projective measurement on P. Hence, by Lemma 2, they all commute. Hence, both the collision probability and $\Pr(\exists i : x_i \neq \hat{x}_i, f(x_i, h_i) = t_i)$ obtained after our ending measurements will remain the same as the original game (where Π_D and collision measurement are not applied). For the collision probability, it is the same as we move $\mathbf{P}U_{t_i}$ and Π_D to after collision measurement; for $\Pr(\exists i : x_i \neq \hat{x}_i, f(x_i, h_i) = t_i)$, it is similar by keeping $\mathbf{P}U_{t_i}$ while moving other two operators to the end of game. Let *col* be the output 0 of measurement $(\Pi_{col}, I - \Pi_{col})$. Notice that

$$\Pr(\exists i : x_i \neq \hat{x}_i, f(x_i, h_i) = t_i || \psi_1\rangle) \tag{14}$$

$$\leq \Pr(\exists i : x_i \neq \hat{x}_i \wedge f(x_i, h_i) = t_i \wedge \neg col || \psi_1\rangle) + \Pr(col || \psi_1\rangle) \tag{15}$$

Notice that register D_{x_i} in $|\psi_1\rangle$ is $|h_i\rangle + 2^{-n/2}(|\bot\rangle - |\phi_0\rangle)$. Since $f(x_i, h_i) = t_i$, it follows that under $\neg col$ condition, $x_i \neq \hat{x}_i$ implies that after measurement on P (that results in $\hat{x}_i$ in the ith component on register P), the post-measurement joint state $|\psi'\rangle_{XYZD}|\hat{\mathbf{x}}\rangle_P$ must have D_{x_i} content different from h_i (that is, $\langle h_i|\psi'\rangle = 0$). Since $|\psi_1\rangle$ has $F|h_i\rangle$ in D_{x_i}, this has a probability $1 - |\langle h_i|(|h_i\rangle + 2^{-n/2}|\phi_0\rangle)|^2 = 1 - (1 - 2^{-n})^2 \leq 2^{-n+1}$. There are at most ℓ possible is. So, the first item in Equation (15) is at most $2^{-n+1}\ell$. On the other hand, $|\psi_1\rangle$ is obtained by measurements and unitaries. Averaging over the choices of $|\psi_1\rangle$ satisfying $\neg$abort (due to intermediate measurements) gives $\Pr(\exists i : x_i \neq \hat{x}_i \wedge f(x_i, h_i) = t_i \wedge \neg col \neg \mathsf{abort}) \leq 2^{-n+1}\ell$. By Lemma 14, $\Pr(col \wedge \neg\mathsf{abort}) \leq 16(q+\ell)^3\Gamma_f/2^n$. Thus, $\Pr(\exists i : x_i \neq \hat{x}_i, f(x_i, h_i) = t_i \wedge \neg\mathsf{abort}) \leq 2^{-n+1}\ell + 16(q+\ell)^3\Gamma_f/2^n$. □

6.3. Extraction on the Fly

We have shown the extraction result where the extractions occur only at the *end* of the game. To be useful, it is expected that we can extract them "on the fly" (i.e., right after each commitment is given during the game). In the following, we consider this. The result is extended from [32] from the **CStO** setting to the $\mathbf{CStO}_s$ setting.

Let us consider a function $f : \mathcal{X} \to \mathcal{T} \cup \{\emptyset\}$ with some special set $\Xi \subset \mathcal{X}$ so that $f(\Xi, \mathcal{Y}) = \emptyset$ and $f(\mathcal{X}\backslash\Xi, \mathcal{Y}) \subseteq \mathcal{T}$. Consider the following games, where $\mathcal{S}.\mathbf{CStO}_s$ is $\mathcal{S}.RO$ or $\mathcal{S}.PR_0$ or $\mathcal{S}.PR_1$.

Game Γ_0. $\mathcal{A}$, with q'_1 queries to $\mathbf{CStO}_s$, outputs $t \in \mathcal{T}$, and then with q'_2 queries to $\mathbf{CStO}_s$, outputs $x \in \mathcal{X}$, and auxiliary output W. Finally, x is classically issued to $\mathbf{CStO}_s$ with response h.

Game Γ_1. $\mathcal{A}$, with q'_1 queries to $\mathcal{S}.\mathbf{CStO}_s$, outputs $t \in \mathcal{T}$, and $\mathcal{S}.E(t)$ is executed to output $\hat{x}$. Then, $\mathcal{A}$ continues q'_2 queries to $\mathcal{S}.\mathbf{CStO}_s$ and finally outputs $x \in \mathcal{X}$ and auxiliary output W. Finally, x is classically issued to $\mathcal{S}.\mathbf{CStO}_s$ with response h.

Let q_1 be the number of random oracle queries or PointReg1 queries in the first q'_1 queries to $\mathcal{S}.\mathbf{CStO}_s$. Similarly, we can define q_2. The pair $(X, Y)_\Gamma$ denotes (X, Y) in game Γ. Define $\Delta((X, Y = y)_{\Gamma_0}, (X, Y = y)_{\Gamma_1}) \overset{def}{=} \frac{1}{2}\sum_x |P_{XY}(x, y) - Q_{XY}(x, y)|$ (a partial sum in the statistical distance), where P_{XY} (respectively, Q_{XY}) is the joint distribution of XY in Γ_0 (respectively, Γ_1).

In the following, we show that the adversarial outputs from Γ_0 and Γ_1 are close. Also, the extraction $\hat{x}$ from $\mathcal{S}.E(t)$ in Γ_1 will be mostly identical to x. The idea is that Γ_0 can be regarded as the simulated game with extraction occurring at the end, because the extraction at the end does not affect the adversarial output. Then, we try to shift $\mathcal{S}.E(t)$ step by step toward right after the output of t and find out that the change of the quantum state throughout this shift process is small. The second claim $x = \hat{x}$ follows from the foregoing argument and Theorem 2.

Theorem 3. *Let $(\alpha)_\Gamma$ be the random variable α with respect to game Γ. Let $\mathcal{A}$ be a quantum algorithm with access to $\mathbf{CStO}_s$ such that $\Xi_1 \subseteq \Xi$. Let $q = q_1 + q_2$. Then,*

$$\Delta((t, x, h, W, \mathit{abort} = 0)_{\Gamma_0}, (t, x, h, W, \mathit{abort} = 0)_{\Gamma_1}) \le 8(q_2 + 1)\sqrt{2\Gamma_f/2^n}, \tag{16}$$

$$\Pr(x \ne \hat{x} \wedge f(x, h) = t \wedge \mathit{abort} = 0) \le 8(q_2 + 1)\sqrt{2\Gamma_f/2^n} + 2^{-n+1} + \frac{16(q+1)^3\Gamma_f}{2^n}. \tag{17}$$

Proof. Let U_t be the unitary measurement on DP, following which the projective measurement $\{P_x\}_{x\in\hat{\mathcal{X}}}$ on register P is applied, resulting in $\hat{x}$. Assume that $\{T_t\}_t$ is the measurement for t. Let V_{XYW} be the unitary operator of $\mathcal{A}$ between queries and $\{M_{xw}\}_{x,w}$ be the measurement for (x, w). The initial state is $|\gamma_0\rangle = |\omega\rangle_{XYW} \otimes (\otimes_x |\bot\rangle_{D_x}) \otimes |0\rangle_P$. Then, the final unnormalized state in Γ_1 is

$$\begin{aligned} |\gamma_1\rangle =& P_h \cdot \mathcal{S}.RO \cdot M_{xw} \cdot (\mathcal{S}.\mathbf{CStO}_s \cdot V)^{q_2} \cdot \mathcal{S}.E(t) \cdot T_t \cdot (\mathcal{S}.\mathbf{CStO}_s \cdot V)^{q_1} |\gamma_0\rangle & (18) \\ =& P_h \cdot \mathbf{CStO}_s \cdot M_{xw} \cdot (\mathbf{CStO}_s \cdot V)^{q_2} \cdot P_{\hat{x}} \cdot U_t \cdot T_t \cdot (\mathbf{CStO}_s \cdot V)^{q_1} |\gamma_0\rangle, & (19) \end{aligned}$$

where the last $\mathbf{CStO}_s$ in Equation (19) is a random oracle query and $P_{\hat{x}} = |\hat{x}\rangle\langle\hat{x}|_P$. Furthermore, if $\mathcal{A}$ makes a random oracle query, then under abort $= 0$, $\mathcal{S}.\mathbf{CStO}_s$ is $CStO_s \cdot \Lambda_{i0}$; if $\mathcal{A}$ makes PointReg1 query x and abort $= 0$, then oracle applies Π_0, and then $U_{\bot,r}$ to D_x. A PointReg0 query does not impact on the quantum state and hence does not occur in the above equation, but it is implicit to maintain Ξ_0. We assume that the operators other than the measurements mentioned are unitary (which can be made up with some auxiliary registers). Then, we have the probability of $xhw\hat{x}t\Xi_1$ with abort $= 0$ in Γ_1 (denoted by $p_{xhw\hat{x}t\Xi_1}$) is $||\gamma_1||^2$. Furthermore, $P_{\hat{x}}$ can be moved to the end of game (as variable $\hat{x}$ and register P are not related to operators currently on the left to $P_{\hat{x}}$), $p_{xhw\hat{x}t\Xi_1} = ||\gamma_2||^2$, where

$$|\gamma_2\rangle = P_{\hat{x}} P_h \cdot \mathbf{CStO}_s \cdot M_{xw} \cdot (\mathbf{CStO}_s \cdot V)^{q_2} \cdot U_t \cdot T_t \cdot (\mathbf{CStO}_s \cdot V)^{q_1} |\gamma_0\rangle. \tag{20}$$

If we remove $P_{\hat{x}} U_t$ from Equation (19), then $|\gamma_1\rangle$ becomes the final state of Γ_0. Then, the probability of $xhw\hat{x}t\Xi_1$ in Γ_0 with abort $= 0$ (denoted by $q_{xhw\hat{x}t\Xi_1}$) is $||\gamma_2'||^2$ (if further applying U_t and projective measurement $\{P_{\hat{x}}\}_{\hat{x}}$ at the end of Γ_0), where

$$|\gamma_2'\rangle = P_{\hat{x}} U_t P_h \cdot \mathbf{CStO}_s \cdot M_{xw} \cdot (\mathbf{CStO}_s \cdot V)^{q_2} \cdot T_t \cdot (\mathbf{CStO}_s \cdot V)^{q_1} |\gamma_0\rangle. \tag{21}$$

By the triangle inequality, Equation (16) is bounded by

$$\frac{1}{2} \sum_{xhw\hat{x}t\Xi_1} |\, ||\,|\gamma_2\rangle||^2 - ||\,|\gamma_2'\rangle||^2 \,| \le \frac{1}{2} \sum_{i=0}^{q_2} \sum_{xhw\hat{x}t\Xi_1} |\, ||\,|\gamma_{2(i+1)}\rangle||^2 - ||\,|\gamma_{2i}\rangle||^2 \,|, \tag{22}$$

where $|\gamma_{2i}\rangle$ is the variant of $|\gamma_2\rangle$ with U_t relocated (starting from the leftmost) to right after the ith $\mathbf{CStO}_s$ operator in $|\gamma_2\rangle$ (that is either random oracle query or PointReg1 query), and thus $\gamma_2' = |\gamma_{20}\rangle$ and $|\gamma_2\rangle = |\gamma_{2(q_2+1)}\rangle$.

We consider the inner summation at Equation (22) for a fixed i. We can separate $xhw\hat{x}t\Xi_1$ as AB, where A is the subset of variables obtained by measurements in $|\gamma_{2i}\rangle$ after U_t and B are the remaining variables. Let $|\psi_B\rangle$ be the state right before U_t and M_A' be the product of operators after U_t and the ith $\mathbf{CStO}_s$ in $|\gamma_{2i}\rangle$. Then, $|\gamma_{2i}\rangle = M_A' \cdot U_t \cdot$

$\mathbf{CStO}_s|\psi_B\rangle$, and $|\gamma_{2(i+1)}\rangle = M'_A \cdot \mathbf{CStO}_s \cdot U_t|\psi_B\rangle$, as $[U_t, V] = 0$. It is well known that the measurement can be made at the end of operation without changing the measurement outcome distribution. Hence, we can assume that $M'_A = M_A S$ for the projection M_A of A and unitary S. That is, we can assume that $|\gamma_{2i}\rangle = M_A \cdot S \cdot U_t \cdot \mathbf{CStO}_s|\psi_B\rangle$ and $|\gamma_{2(i+1)}\rangle = M_A \cdot S \cdot \mathbf{CStO}_s \cdot U_t|\psi_B\rangle$. Let $|\psi'_B\rangle$ be the normalized $|\psi_B\rangle$. Then,

$$\frac{1}{2}\sum_{xhwt\mathbf{b}\Xi} |\ |||\gamma_{2(i+1)}\rangle||^2 - |||\gamma_{2i}\rangle||^2\ | \tag{23}$$

$$=\sum_B |||\psi_B\rangle||^2 \cdot \frac{1}{2}\sum_A |\ ||M_A \cdot S \cdot U_t \cdot \mathbf{CStO}_s|\psi'_B\rangle||^2 - ||M_A \cdot S \cdot \mathbf{CStO}_s \cdot U_t|\psi'_B\rangle||^2\ | \tag{24}$$

If $\mathbf{CStO}_s$ is a random oracle query, then the inner sum is the statistical distance between measurement outcomes from $S \cdot U_t \cdot CStO_s \cdot \Lambda|\psi'_B\rangle$ and $S \cdot CStO_s \cdot U_t \cdot \Lambda|\psi'_B\rangle$ (Note: Here, Λ is some Λ_{i0}, and $[U_t, \Lambda] = 0$). By Theorem 9.1 [36], it is no more than their trace distance. Further, by Lemma 4, the trace distances of two states are no more than their Euclidean distances, which are further bounded by $||[CStO_s, U_t]||$ (by the form of Equation (24)). Hence, by Theorem 1,

$$\textit{Equation } (24) \le \sum_B |||\psi_B\rangle||^2 \cdot ||U_t, \mathbf{CStO}_s|| = ||U_t, \mathbf{CStO}_s|| \le 8 \cdot 2^{-n/2}\sqrt{2\Gamma_f}. \tag{25}$$

If $\mathbf{CStO}_s$ is PointReg1 query $x \in \Xi_0$ with $\mathsf{abort} = 0$, this will apply Π_0 and $U_{\perp,r} = |r\rangle\langle\perp| + |\perp\rangle\langle r| + \sum_{s\neq r}|s\rangle\langle s|$ to register D_x. Note that U_t commutes with $U_{\perp,r}$ if $f(x,r) \neq t$ (because $R_t(x,r) = R_t(x,\perp) = 0$ and so $|\perp\rangle_{D_x}$ replaced by $|r\rangle_{D_x}$ will not change $\hat{x}$). By Lemma 2(2), $[\Pi_0, U_t] = 0$. Thus, $\mathbf{CStO}_s$ (i.e., PointReg1) commutes with U_t if $f(x,r) \neq t$. By our assumption, $\mathcal{A}$ satisfies $\Xi_1 \subseteq \Xi$. Hence, $f(x,r) = \emptyset$, and so $f(x,r) = t$ will never hold. Hence, PointReg1 commutes with U_t. Hence, Equation (24) is 0 for this query.

Finally, since there are at most $q_2 + 1$ random oracle queries after t is measured, Equation (22) is bounded by $8(q_2+1)\sqrt{2\Gamma_f/2^n}$.

Now, we consider the second claim. Notice that Z is defined as a Boolean variable $(x \neq \hat{x} \wedge f(x,h) = t \wedge \mathsf{abort} = 0)$ of $(x,h,\hat{x},t)$. We still use p_Z to denote the distribution in Γ_1 and q_Z to denote the distribution of Z in Γ_0. Then, by the forgoing argument, $p_Z(1) \le q_Z(1) + 8(q_2+1)\sqrt{2\Gamma_f/2^n}$. Then, by Theorem 2, $q_Z(1) \le 2^{-n+1} + 16(q+1)^3\Gamma_f/2^n$. The result follows. □

The above theorem can be extended to the vector case, where M_{xw}, U_t are replaced with several $M_{x_iw_i}, U_{t_i}$ at location i. Then, we switch U_{t_i} with each $\mathbf{CStO}_s$ after t_i is measured, as in the above theorem. Denote the number of this kind of $\mathbf{CStO}_s$ (that is either a random oracle query or PointReg1 query) by q_{2i}. Then, $q_{2i} < q$. For each i, we obtain the similar bound as the above theorem. Summarizing the argument for $i = 1, \cdots, \ell$, the extension of the first claim can be obtained. The extension of the second claim is very similar to the second claim of the above theorem.

Corollary 2. *Let q be the total number of random oracle queries or PointReg1 queries and $\Xi_1 \subseteq \Xi$. If $(\mathbf{x}, \mathbf{t}, \mathbf{h}, \hat{\mathbf{x}})$ with vector length ℓ is the vector corresponding to $(x,t,h,\hat{x})$ in Theorem 3, then*

$$\Delta((\mathbf{t}, \mathbf{x}, \mathbf{h}, W, \mathsf{abort} = 0)_{\Gamma_0}, (\mathbf{t}, \mathbf{x}, \mathbf{h}, W, \mathsf{abort} = 0)_{\Gamma_1}) \le 8(q+\ell)\ell\sqrt{2\Gamma_f/2^n} \tag{26}$$

$$\Pr(\exists i : x_i \neq \hat{x}_i \wedge f(x_i,h_i) = t_i \wedge \mathsf{abort} = 0) \le 8(q+\ell)\ell\sqrt{\frac{2\Gamma_f}{2^n}} + \frac{2\ell}{2^n} + \frac{16(q+\ell)^3\Gamma_f}{2^n}.$$

Remark 6. *Theorem 3 requires $\Xi_1 \subset \Xi$. If this is not satisfied, then the proof cannot get through. However, this condition is only used in the PointReg1 query to guarantee that $f(x,r) \neq t$. Since r is taken uniformly and randomly after x is fixed, this condition holds for $2^n - \Gamma_t$ choices of r. If there are at most q_s PointReg1 queries, this holds for every PointReg1 query with a probability of at least $1 - q_s\Gamma_t/2^n$. When this holds, the proof of Theorem 3 remains valid. Furthermore, this argument extends to the vector case in Corollary 2 with further observation that Equation (26) holds with q replaced by $q - q_s$, as that is the bound from the number of the random oracle queries. Notice that $\Gamma_t/2^n < 8\ell\sqrt{2\Gamma_t/2^n}$. Hence, with this tighter analysis, we have the following corollary that preserves the same bound.*

Corollary 3. *Let q be the number of random oracle queries or PointReg1 queries. $(\mathbf{x}, \mathbf{t}, \mathbf{h}, \hat{\mathbf{x}})$ with vector length ℓ is the vector corresponding to $(x, t, h, \hat{x})$ in Theorem 3. Let $\mathcal{A}$ be a quantum algorithm with access to $\mathbf{CStO}_s$ with at most q_s PointReg1 queries. Then,*

$$\Delta((\mathbf{t}, \mathbf{x}, \mathbf{h}, W, \mathit{abort} = 0)_{\Gamma_0}, (\mathbf{t}, \mathbf{x}, \mathbf{h}, W, \mathit{abort} = 0)_{\Gamma_1}) \leq 8(q+\ell)\ell\sqrt{2\Gamma_f/2^n},$$

$$\Pr(\exists i : x_i \neq \hat{x}_i \wedge f(x_i, h_i) = t_i \wedge \mathit{abort} = 0) \leq 8(q+\ell)\ell\sqrt{\frac{2\Gamma_f}{2^n}} + \frac{\ell}{2^{n-1}} + \frac{16(q+\ell)^3\Gamma_f}{2^n}.$$

7. Extracting Queries to CStO that Witness the Future Adversarial Output

7.1. Motivation

In the last section, we have learned how to extract a query for a given commitment on the fly. However, how can we achieve an early extraction for the future output (i.e., no commitment is given at the time of extraction)? For example, in the multi-signature security model, an adversary will finally make a forgery with respect to a set of public keys. However, this set of public keys (say, PK) will be revealed only at the end of the game when the attacker shows its forgery. We can not guess attackers' public keys, as they are completely created by himself. In this case, if the attacker has queried PK to a random oracle, then in the classical setting, we can guess which query is PK while in the quantum setting, this is not clear how to guess because the query might be in a superposition. Liu and Zhandry [33] developed a random experiment by measuring a random query to give PK as a special point and showed that it matches the final output with good probability. In the following, we will extend their technique to the setting of multiple special points.

7.2. Random Experiment

In the above motivation, we consider the extraction of PK for a multi-signature forgery. In general, we want to extract an adversarial query that matches the adversary's final output which is unknown at the time of the extraction. This extraction technique is very useful in a security proof when the final adversary output is the final solution of the attack, while the query input to be extracted is a certain witness of this solution. In the following, we extend their technique to the setting of multiple extractions (but still interacting with **CStO**). This modified game can be used to extract multiple queries that are collectively used to derive a witness for the final adversary output. This game can be easily converted to one where the random oracle is $\mathbf{CStO}_s$, and so our extraction theorems in the previous sections can be used.

Assume that adversary $\mathcal{A}$ makes at most q oracle queries to **CStO** oracle. In the end, we measure the adversary–oracle joint state and obtain $(w, \mathbf{y})$ so that D has the collapsed state $F_D|\mathbf{y}\rangle_D$ (i.e., measuring the final state on D using $\{F_D|\mathbf{y}\rangle_\mathbf{D}\}_\mathbf{y}$ basis). Let $\lambda_{w,\mathbf{y}}$ denote the probability of outcome $(w, \mathbf{y})$. We define game $\mathsf{Exp}_{i,j,k}$ (with either $i = j = k$ or $i < j < k$ for $i, j, k \in [q]$). Before this, we define $\underline{x}$ as an *equivalence class* (which is a subset of $\mathcal{X}$, including x and also determined by x) in the sense that $\underline{x} = \underline{u}$ for any $u \in \underline{x}$. We assume that the cardinality of $\underline{x}$ is polynomially bounded. For $\mathbf{y} \in \mathcal{Y}^\mathcal{X}$, $\mathbf{y}(\underline{x}) = \vec{\perp}$ means that $y_u = \perp$ for $\forall u \in \underline{x}$.

$\mathsf{Exp}_{i,i,i}$: In this game, it proceeds normally until the ith oracle query. Assume the attacker–oracle state is $\sum_{xuz\mathbf{y}} \alpha_{xuz\mathbf{y}}|x, \phi_u, z, \mathbf{y}\rangle$, where we recall that Y register is represented using Fourier basis $\{\phi_u\}_{u\in\mathcal{Y}}$. Then, we measure the query input with outcome $\underline{x}^*$, which can be done we follows: let $\mathsf{rep}(\underline{x}) \in \mathcal{X}$ be the representative of $\underline{x}$, and assume that it can be efficiently computed from any $u \in \underline{x}$; let U_C be a unitary with $|x\rangle_X|0\rangle_C \mapsto |x\rangle|\mathsf{rep}(\underline{x})\rangle$; measuring register C in the computational basis gives $\mathsf{rep}(\underline{x})$.

We further measure to test (by two measurements) whether it holds: $D(\underline{x}^*) = \vec{\bot}$ before the oracle query (which can be done as follows: $D(\underline{x}^*) = \vec{\bot}$ can be tested by a projective measurement $\Pi_\bot = (\Pi^0_\bot, I - \Pi^0_\bot)$ with $\Pi^0_\bot = \sum_{\mathbf{y}:\mathbf{y}(\underline{x}^*)=\vec{\bot}} |\mathbf{y}\rangle\langle\mathbf{y}|$, which can be implemented by writing bit $\mathbf{y}(\underline{x}^*) == \vec{\bot}$ onto a new register and measuring it) but $D(\underline{x}^*) \neq \vec{\bot}$ after the oracle query (which can be done as follows: if $D(x') = \bot$ before the oracle query, then it remains $D(x') = \bot$ *after the oracle query* (i.e., after applying $CStO$) if and only if Y register is currently $|\phi_0\rangle$. Thus, to test if $D(x') = \bot$ *after the oracle query, we can simply apply the unitary* $|\phi_y\rangle_Y|0\rangle_Q \mapsto |\phi_y\rangle_Y|y\rangle_Q$ *and measure if* Q *register has* 0. *That is, we can make the test* without *applying the* $CStO$ *operation*). If both test measurements succeed, then the resulting state before applying $CStO$ oracle will be

$$\sum_{x'uz\mathbf{y}:\ y_{x'}=\bot, u\neq 0,\ x'\in\underline{x}^*} \alpha_{x'uz\mathbf{y}}|x', \phi_u, z, \mathbf{y}\rangle, \tag{27}$$

where the case $u = 0$ is removed because these components will still have $D(\underline{x}^*) = \bot$ after the $CStO$ query. In this case, the state after the $CStO$ query will become

$$\sum_{x'uz\mathbf{y}:\ y_{x'}=\bot, u\neq 0, x'\in\underline{x}^*} \alpha_{x'uz\mathbf{y}}|x', \phi_u, z\rangle \frac{1}{\sqrt{2^n}} \sum_{y\in\mathcal{Y}} (-1)^{u\cdot y}|\mathbf{y} \cup (y)_{x'}\rangle. \tag{28}$$

Then, the game proceeds normally. If one or both measurements fails, the game aborts.

$\mathsf{Exp}_{i,j,k}$ with $i < j < k$: In this game, it proceeds normally until the ith oracle query. Let the attacker–oracle state be $\sum_{xuz\mathbf{y}} \alpha_{xuz\mathbf{y}}|x, \phi_u, z, \mathbf{y}\rangle$. Then, we measure the query input to output $\underline{x}^*$ and then measure (similar to that in $\mathsf{Exp}_{i,i,i}$) to test whether the followings are satisfied throughout the ith oracle query to the kth oracle query (using the methods mentioned above):

- Right before the ith query, $D(\underline{x}^*) = \vec{\bot}$; but after it, $D(\underline{x}^*) \neq \vec{\bot}$.
- After ith query and before the jth query, it remains that $D(\underline{x}^*) \neq \vec{\bot}$.
- After jth query and before the kth query, $D(\underline{x}^*) = \vec{\bot}$.
- Right after the kth query, $D(\underline{x}^*) \neq \vec{\bot}$.

If the test measurement fails, the game aborts; otherwise, it proceeds normally. It should be emphasized that we do not care if $D(\underline{x}^*) = \vec{\bot}$ after any other query than those listed above.

We remark that $\mathsf{Exp}_{i,i,i}$ in fact is a special case of $\mathsf{Exp}_{i,j,k}$, with $i = j = k$ as "after ith query and before the j query" and "after jth query and before the k query" in $\mathsf{Exp}_{i,j,k}$ are both null statements in this setting.

Further, although $\mathsf{Exp}_{i,j,k}$ is defined in the game between adversary and **CStO**, by inspecting its definition, we can see that $\mathsf{Exp}_{i',j',k'}$ in $\mathsf{Exp}_{i,j,k}$ is also well defined (as the conducted measurements are well defined). It is not hard to see that the game $\mathsf{Exp}_{i,j,k}$ in $\mathsf{Exp}_{i',j',k'}$ and the game $\mathsf{Exp}_{i',j',k'}$ in $\mathsf{Exp}_{i,j,k}$ are the same. By iteration, we can define $\mathsf{Exp}_{i^t,j^t,k^t}$ as game $\mathsf{Exp}_{i_t,j_t,k_t}$ in $\mathsf{Exp}_{i^{t-1},j^{t-1},k^{t-1}}$, where v^t is the sequence $v_1, \cdots, v_t$. Let $\mathcal{U}_{IJK}$ be the distribution of (i, j, k) that is uniformly random in $\{(i, i, i) \mid i \in [q]\} \cup \{(i, j, k) \mid 1 \leq i < j < k \leq q\}$. Furthermore, $\mathcal{U}^c_{IJK}$ is the product distribution of $\mathcal{U}_{IJK}$ of c copies.

7.3. Extraction Theorem

The following is the main result in this section. This is an extension of Corollary 6 [33] with the proof mainly extending Theorem 9 [33] . It essentially states that if the adversary has a successful probability in the original game, then in the random experiment $\mathsf{Exp}_{i^c,j^c,k^c}$ for $(i^c, j^c, k^c) \leftarrow \mathcal{U}^c_{IJK}$, it will have a successful probability that is degraded only by a polynomial fraction. With this result, we can reduce our security analysis to this random experiment. The advantage of this result is that we can set the special points of $\underline{x}_{i_j}$ to any value of our choices during the $k_j$the query because $D(\underline{x}_{i_j}) = \vec{\perp}$, where $j = 1, \cdots, c$. This is a similar capability in a classical random oracle model. The detailed proof of this theorem can be found in Appendix D.

Theorem 4. *Let $c > 0$ be a constant. Take $(i^c, j^c, k^c) \leftarrow \mathcal{U}^c_{IJK}$. Let S be a subset of the possible output $(w, \mathbf{y})$ in the game with CStO oracle. Define the measurement (P_0, P_1) with $P_0 = \sum_{(w,\mathbf{y})\in S} |w, \tilde{\mathbf{y}}\rangle\langle w, \tilde{\mathbf{y}}|$ (where we use the basis $F_D|\mathbf{y}\rangle = |\tilde{\mathbf{y}}\rangle$ for the consistency with the measurement at the beginning of this section) and $P_1 = I - P_0$. Let $x_{w,\mathbf{y},t} \in \mathcal{X}$ for $t = 1, \cdots, c$ be representatives for c (possibly repeating) classes, being determined by $(w, \mathbf{y})$ with $\mathbf{y}(\underline{x}_{w,\mathbf{y},t}) \neq \perp$. Let λ be the probability in the random game $\mathsf{Exp}_{i^c,j^c,k^c}$ that gives $\underline{x}_{w,\mathbf{y},t}$ for some $(w, \mathbf{y}) \in S$ from the measurement on the i_tth oracle query for $t = 1, \cdots, c$, and the final measurement (P_0, P_1) gives outcome 0. Let γ be the probability that the final measurement in the normal game gives outcome 0. Then, $\lambda \geq \frac{\gamma}{(q+\binom{q}{3})^{3c}}$.*

8. Quantum Security of the JAK Multi-Signature Framework

Jiang et al. [23] proposed a framework that converts a linear ID scheme into a compact multi-sinagure scheme. In this framework, each signer i with public key pk_i starts with a commitment $r_i = H_0(\mathrm{CMT}_i|pk_i)$ to his first ID message CMT_i. The aggregated public key is $\overline{pk} = \sum_i \lambda_i \bullet pk_i$, where $\lambda_i = H_0(pk_i, \{pk_j\}_{j=1}^n)$. They proved its security in the classical random oracle model. In that proof, a simulator can extract CMT_i of signer i (played by attacker) by searching through the oracle query history that matches r_i. This strategy cannot be used in the quantum setting, as an attacker might query $\mathrm{CMT}_i|pk_i$ in a superposition. To resolve this difficulty, we use the extraction technique in Section 6.3 to handle it. Similarly, the proof in the classical random oracle model can detect early, which public key set $\{pk_j\}_{j=1}^n$ will be used for the forgery by randomly guessing from all possible queries toward some λ_i. Again, this guessing cannot be directly used in the quantum setting. To resolve this, we use the technique in Section 7 to handle. This gives an outline of the main technical differences from a classical proof.

This section is planned as follows. We review the multi-signature framework [23] in Section 8.1. Then, we prove its security in the quantum random oracle model in Section 8.2 using the techniques outlined above.

8.1. Review of JAK Mutli-Signature Framework

In this section, we review the multi-signature framework in [23]. Essentially, to generate a multi-signature on message M, each signer signs M by converting a canonical ID scheme but with the same challenge CH (from Fiat–Shamir) and then linearly combines these linear signatures in a compact signature.

Let

$$\mathcal{ID} = (\mathbf{Setup}_{id}, \mathbf{KeyGen}_{id}, P, V_\tau, \Theta)$$

be a canonical linear ID with parameter $\tau \in \mathbb{N}$. Let H_0, H_1 be two random oracles from $\{0,1\}^*$ to Θ with $\Theta \subseteq \mathcal{R}$, where $\mathcal{R}$ is the ring defined for the linearity property of $\mathcal{ID}$. The JAK multi-signature scheme $(\mathbf{Setup}, \mathbf{KeyGen}, \mathbf{Sign}, \mathbf{Verify})$ is as follows.

Setup. Sample and output $\mathsf{param} \leftarrow \mathbf{Setup}_{id}(1^\lambda)$.

KeyGen. Sample $(pk, sk) \leftarrow \mathbf{KeyGen}_{id}(\mathsf{param})$; output public key pk and private key sk.

Sign. Assume that signers with public keys $\{pk_i\}_{i=1}^t$ want to jointly sign message M. Let $\lambda_i = H_0(pk_i, PK)$ and $\overline{pk} = \sum_{i=1}^t \lambda_i \bullet pk_i$, where $PK = (pk_1, \cdots, pk_t)$. They execute the following:

- *R-1.* Signer i takes $(st_i, \mathrm{CMT}_i) \leftarrow P(\mathsf{param})$ and sends $r_i := H_0(\mathrm{CMT}_i|pk_i)$ to all signers.

- *R-2.* Upon r_j for all j (we do not restrict $j \neq i$ for brevity), signer i sends CMT_i to all signers.
- *R-3.* Upon $\mathrm{CMT}_j, j = 1, \cdots, t$, signer i checks if $r_j = H_0(\mathrm{CMT}_j|pk_j)$ for all j. If no, it rejects; otherwise, it computes $\overline{\mathrm{CMT}} = \sum_{j=1}^{t} \lambda_j \bullet \mathrm{CMT}_j$, $\mathrm{CH} = H_1(\overline{pk}|\overline{\mathrm{CMT}}|M)$ and $\mathrm{Rsp}_i = P(st_i|sk_i|pk_i, \mathrm{CH})$. Finally, it sends Rsp_i to all signers.
- *Output.* Upon $\mathrm{Rsp}_j, j = 1, \cdots, t$, signer i computes $\overline{\mathrm{Rsp}} = \sum_{j=1}^{t} \lambda_j \bullet \mathrm{Rsp}_j$ and outputs the aggregated public key $\overline{pk}|t$ and multi-signature $\overline{\mathrm{CMT}}|\overline{\mathrm{Rsp}}$.

Verify. Upon signature $(\overline{\mathrm{CMT}}, \overline{\mathrm{Rsp}})$ on message M with the aggregated public key $\overline{pk}|t$, it outputs $V_t(\overline{pk}, \overline{\mathrm{CMT}}|\mathrm{CH}|\overline{\mathrm{RSP}})$, where $\mathrm{CH} = H_1(\overline{pk}|\overline{\mathrm{CMT}}|M)$.

8.2. Security Theorem

In this section, we prove the security of the JAK framework in the quantum random oracle model. Our proof strategy is to use the sequence of game techniques. We first replace two random oracles $|H_0\rangle$ and $|H_1\rangle$ with a single oracle $|H\rangle$ so that $H(0|x) = H_0(x)$ and $H(1|x) = H_1(x)$. Since the distributions of $H(b|x)$ and $H_b(x)$ are identical, the adversary success does not decrease. Then, we replace $|H\rangle$ by **CStO**, and this will not change the adversary success by Fact 1 and Lemma 8. Next, we sample experiment $\mathbf{Exp}_{i^2,j^2,k^2}$ so that the i_1th query has measurement outcome $\underline{x}_1^*$ with $x_1^* = 0|pk_1'|PK'$, where PK' is the signature group in the attacker's forgery, and the measurement outcome for the i_2th query is $\underline{x}_2^*$, with $x_2^* = 1\overline{pk'}|\overline{\mathrm{CMT}'}|M$ being the attacker's input to compute CH' in its forgery. By Theorem 4, the adversary success in this experiment is degraded only by a polynomial fraction. Then, we consider the signing oracle in $\mathbf{Exp}_{i^2,j^2,k^2}$. We will try to confirm (by measurement) that the query input $x = 1|\overline{pk}|\overline{\mathrm{CMT}}|M$ to compute CH is not recorded in **CStO** (so that we can set this CH by ourselves). Since $\overline{\mathrm{CMT}}$ contains the challenger's committing message (that has super-logarithmic min-entropy), this confirmation measurement will succeed with high probability (Lemma 10). Then, we reformulate $\mathbf{Exp}_{i^2,j^2,k^2}$ as the game with **CStO**$'$. The format of $\mathbf{Exp}_{i^2,j^2,k^2}$ is very compatible with **CStO**$'$, and so this switch is just a simple formatting problem. Then, we further change to a game with **CStO**$_s$, and by Lemma 9, the adversary success probability will not change. Now, under the game with **CStO**$_s$, we can use the extraction technique to extract the committing messages from adversary in a signing oracle and treat $x = 1|\overline{pk}|\overline{\mathrm{CMT}}|M$ as a special point. We also treat $\underline{x}_1^*, \underline{x}_2^*$ as special points. We can set the random oracle value of these special points by ourselves. With this benefit, we use the ID simulator to simulate the honest signer's messages in a signing oracle without its secret. Finally, we can reduce the adversary success to break the ID scheme by setting the CH in the attacker's forgery as the challenge from the ID challenger. So, the attacker's forgery will help us to break the ID security.

Theorem 5. *Assume that $h \leftarrow \Theta$ is invertible in $\mathcal{R}$ with probability $1 - negl(\lambda)$. Let $\mathcal{ID} = (\mathbf{Setup}_{id}, \mathbf{KeyGen}_{id}, P, V_\tau)$ be a secure ID scheme with linearity and simulability. Then, the JAK multi-signature scheme is* **EU-CMA**-*secure in the quantum random oracle model.*

Proof. Our proof follows the sequence of game strategy. The game consists of quantum polynomial time adversary $\mathcal{D}$ and a challenger $\mathcal{C}$ who maintains the quantum random oracle and the signing oracle that jointly signs a message M with $\mathcal{D}$. We use $\mathsf{Succ}(\mathbf{G})$ to denote the adversary success probability in game $\mathbf{G}$.

Game $\mathbf{G}_0$. This is the real forgery game. The challenger runs $\mathbf{Setup}(1^\lambda)$ to generate param and executes $\mathbf{KeyGen}(\mathsf{param})$ to generate a challenge key pair (pk^*, sk^*). Then, it provides (pk^*, param) to $\mathcal{D}$ and maintains two quantum random oracles $|H_0\rangle, |H_1\rangle$ and signing oracle $\mathcal{O}_s$ to interact with $\mathcal{D}$. Finally, $\mathcal{D}$ outputs a forgery (σ^*, M^*) with a set of public keys $(pk_1^*, \cdots, pk_N^*)$, where $pk^* = pk_1^*$. He succeeds if $\mathsf{Verify}(\overline{pk^*}, \sigma^*, M^*) = 1$ and no query $(pk_1^*, \cdots, pk_N^*, M^*)$ was issued to $\mathcal{O}_s$.

Game $\mathbf{G}_1$. We modify $\mathbf{G}_0$ to $\mathbf{G}_1$ so that $H_0(x) = H(0|x)$ and $H_1(x) = H(1|x)$ for a random oracle H. This does not reduce the adversary success probability, as the tables for $H(0|\cdot), H(1|\cdot)$ and the tables for $H_0(\cdot), H_1(\cdot)$ are jointly identically distributed (i.e., purely random in both cases). Any query $|\psi\rangle$ to $H_b(\cdot)$ is a special case of query $|b\rangle|\psi\rangle$ to $|H\rangle$. Thus, $\Pr(\mathbf{Succ}(\mathbf{G}_1) \geq \Pr(\mathbf{Succ}(\mathbf{G}_0))$.

Game $\mathbf{G}_2$. We modify $\mathbf{G}_1$ to $\mathbf{G}_2$ so that the random oracle is implemented using **CStO**. By Fact 1 and Lemma 8, the success probabilities of $\mathcal{D}$ in $\mathbf{G}_1$ and $\mathbf{G}_2$ are identical.

Game $\mathbf{G}_3$. We modify $\mathbf{G}_2$ to $\mathbf{G}_3$ so that it selects the game (involving $\mathcal{D}$) $\mathbf{Exp}_{i^2,j^2,k^2}$ for $(i^2, j^2, k^2) \leftarrow \mathcal{U}^2_{IJK}$. Let the measurement at the i_tth oracle query be $\underline{x}^*_t$ for some x^*_t for $t = 1, 2$. At the end of the game, let $(w, \mathbf{y})$ be the measurement output, where w is the forgery (α, β, PK', M) measured by $\mathcal{D}$ on register XYW, and $\mathbf{y}$ is the measurement outcome on D (which represents the quantum state $F_D|\mathbf{y}\rangle_D$, and hence, $\mathbf{y}$ satisfies $y_x = RO(x)$). Define $x_{w,\mathbf{y},1} = 0|pk'_1|PK'$ for $PK' = (pk'_1, \cdots, pk'_n)$. Furthermore, define $\underline{x}_{w,\mathbf{y},1} = \{0|pk'_v|PK' : v = 1, \cdots, n\}$ and $\underline{x} = \{x\}$ (for any x that cannot be written in $0|pk_v|PK$ with $pk_v \in PK$). Hence, the equivalence class is well defined. In addition, define $x_{w,\mathbf{y},2} = 1|\overline{pk'}|\alpha|M$. We consider the case $x^*_t = x_{w,\mathbf{y},t}$ for $t = 1, 2$. Define S in Theorem 4 as the set of all pairs $(w, \mathbf{y})$ so that w is a valid forgery under random oracle assignments $y_x = RO(x)$. Since the probability $(w, \mathbf{y}) \in S$ is the success probability of $\mathcal{D}$ in $\mathbf{G}_2$, by Theorem 4, the success probability of $\mathcal{D}$ in $\mathbf{G}_3$ will be at least $\frac{\epsilon}{(q+\binom{q}{3})^6}$.

Game $\mathbf{G}_4$. We modify $\mathbf{G}_3$ to $\mathbf{G}_4$ so that in the signing oracle, right before the classical oracle query $x = 1|\overline{pk}|\overline{\mathrm{CMT}}|M$ to generate CH, it does a measurement $(|\perp\rangle\langle\perp|, I - |\perp\rangle\langle\perp|)$ to the register D_x of the oracle. If it gives the outcome 1, it aborts with Fail (indicating the failure of the simulation); otherwise, it continues normally. By Lemma 10, this Fail occurs only with a negligible probability (recall that $H_\infty(\mathrm{CMT})$ is super-logarithmic for randomly generated CMT), and hence, the success probability $\mathcal{D}$ in $\mathbf{G}_4$ is at least $\frac{\epsilon}{(q+\binom{q}{3})^6} - \mathbf{negl}(\kappa)$

Game $\mathbf{G}_5$. We reformat $\mathbf{G}_4$ as a game between an adversary $\mathcal{D}$ and challenger $\mathcal{C}'$ that has oracle access to $\mathbf{CStO}'$ (ref. Section 4.3) so that $\mathcal{D}$ in $\mathbf{G}_5$ has the success probability exactly identical to that of $\mathcal{D}$ in $\mathbf{G}_4$. The code of $\mathcal{C}'$ as follows. It follows $\mathcal{C}$ to set up $\mathbf{G}_4$ to invoke $\mathcal{D}$ with the public parameters and then interacts with $\mathcal{D}$. $\mathcal{C}'$ also follows $\mathcal{C}$ to choose the random game $\mathbf{Exp}_{i^2,j^2,k^2}$:

- Whenever a random oracle query is issued, $\mathcal{C}'$ does as follows. Assume that this is the ℓth random oracle query. If $\ell = i_1$ or i_2, then $\mathcal{C}'$ (like challenger $\mathcal{C}$ in $\mathbf{G}_4$) will apply a projective measurement on X register in the computational basis and results in $\underline{x}^*_1$ or $\underline{x}^*_2$, and then it issues a PointReg0 query with each $x \in \underline{x}^*_1$ or $\underline{x}^*_2$ to $\mathbf{CStO}'$. If $\ell = k_t$ (for $t = 1$ or 2), it issues a $PointReg1$ query with $x' \in \underline{x}^*_t$ (which does measurement Π on $D_{x'}$ like challenger in Γ_4). Then (no matter what is ℓ), recall that, in $\mathbf{G}_4$, the challenger will conduct a projective measurement Λ' (determined by ℓ and i_1, j_1, k_1) on D and another projective measurement Λ'' (still determined by ℓ, i_2, j_2, k_2) on D. These measurements are described in $\mathbf{Exp}_{i^2,j^2,k^2}$, and it can be seen that they are only applied on D_{Ξ_0} as desired by $\mathbf{CStO}'$. These two measurements can be combined into one projective measurement $\Lambda_\ell = (\Lambda_{\ell 0}, I - \Lambda_{\ell 0})$ in the computational basis on D_{Ξ_0}. Then, to be consistent with $\mathbf{G}_4$, $\mathcal{D}'$ in $\mathbf{G}_5$ issues the random oracle query with its register XY to $\mathbf{CStO}'$, which will handle it first with measurement Λ_ℓ and then with $CStO$ (if it does not abort). Under this reformatting, the action on the joint state is the same as in $\mathbf{G}_4$.
- When $\mathcal{D}$ issues a signing query (PK, M) so that PK contains pk^*_1, $\mathcal{C}'$ in $\mathbf{G}_5$ computes $\overline{pk}$, $\overline{\mathrm{CMT}}$, and $x = 1|\overline{pk}|\overline{\mathrm{CMT}}|M$ normally as in $\mathbf{G}_4$, with possibly random oracle access to $\mathbf{CStO}'$ as in the previous item. Next, it issues $PointReg0$ query, then $PointReg1$ query both with x to $\mathbf{CStO}'$, and finally a classical random oracle query with x (if it does not abort), where the random oracle queries are handled as the above reformatting. In turn, if $\mathbf{CStO}'$ does not abort, $\mathcal{C}'$ receives the reply $y = RO(x)$, and it continues normally as in $\mathbf{G}_4$ to generate the signature. Note that $\mathcal{C}'$, together with $\mathbf{CStO}'$, acts

the same as $\mathcal{C}$ together with **CStO** in $\mathbf{G}_4$. Thus, this does not change the view of $\mathcal{D}$ and the joint quantum state.

From our description, we can see that $\mathcal{D}$ in $\mathbf{G}_4$ and $\mathbf{G}_5$ have the same view, as this is just a reformatting of $\mathbf{G}_4$. Hence, $\mathcal{D}$ in $\mathbf{G}_5$ has the same success probability as in $\mathbf{G}_4$.

Game $\mathbf{G}_6$. We modify $\mathbf{G}_5$ to $\mathbf{G}_6$ s.t. $\mathbf{CStO'}$ is replaced by $\mathbf{CStO}_s$. By Lemma 9, the success probability of $\mathcal{D}$ in $\mathbf{G}_6$ is the same as in $\mathbf{G}_5$ by checking the output of $\mathcal{C}'$, which is defined as 1 if and only if $\mathcal{D}$ succeeds ($\neg$**abort** can be removed in the lemma, as $\mathcal{C}'$ outputting 1 indicates $\neg\textbf{abort} = 1$).

Game $\mathbf{G}_7$. We modify $\mathbf{G}_6$ to $\mathbf{G}_7$ so that $\mathbf{CStO}_s$ is now simulated by $\mathcal{S}$. Since $\mathcal{S}.E$ is not used, the adversary success probability is identical to $\mathbf{G}_6$.

Game $\mathbf{G}_8$. We modify $\mathbf{G}_7$ to $\mathbf{G}_8$ so that in the signing query $O_s(pk_1, \cdots, pk_n, M)$, after receiving r_i, challenger extracts $\text{CMT}_i' = \mathcal{S}.E(r_i)$ and later in round R-3, when it receives CMT_i; if $\text{CMT}_i \neq \text{CMT}_i'$ but $\mathcal{S}.RO(\text{CMT}_i) = r_i$, it terminates with Fail. By Corollary 2, this occurs negligibly. Thus, the success probability of $\mathcal{D}$ in $\mathbf{G}_8$ is negligibly close to that in $\mathbf{G}_7$.

Game $\mathbf{G}_9$. We modify $\mathbf{G}_8$ to $\mathbf{G}_9$ so that in $O_s(pk_1, \cdots, pk_n, M)$ with $pk_t = pk^*$ for some t, it generates $(\text{CMT}_t, \text{Rsp}_t) \leftarrow \mathbf{SIM}(\text{CH}, pk^*, \mathsf{param})$, where $\text{CH} \leftarrow \Theta$. It does the same as $\mathbf{G}_8$: measures $(|\bot\rangle\langle\bot|, I - |\bot\rangle\langle\bot|)$ on D_x (specified since $\mathbf{G}_4$), issues a *PointReg*0 query, and then *PointReg*1 queries with $x = 1|\overline{pk}|\overline{\text{CMT}}|M$ to $\mathbf{CStO}_s$, where *PointReg*1 will define r in $\mathbf{CStO}_s$ for D_x (if it does not abort) as the random oracle value for x. In $\mathbf{G}_9$, it defines this r as CH. By the simulability of ID, this has the same distribution as $\mathbf{G}_8$. So, the adversary success probability remains the same as in $\mathbf{G}_8$ (specifically, any non-negligible difference in this success probability can be straightforwardly reduced through hybrid argument on $(\text{CMT}_t, \text{Rsp}_t, \text{CH}_t)$ in the signing queries to break the ID simulability; the details are omitted). We recall that the secret key sk is no longer used in $\mathbf{G}_9$.

Game $\mathbf{G}_{10}$. We modify $\mathbf{G}_9$ to $\mathbf{G}_{10}$ so that it will embed the ID challenge into the attack. Specially, $\mathcal{C}'$ sets up the game so that pk_1^* is the ID challenge key. In addition, after obtaining $\underline{x}_1^*$ (by measuring the i_1th random oracle query) with $x_1^* = 1|pk_1^*|\{pk_1^*, \cdots, pk_n^*\}$, it sends $pk_2^*, \cdots, pk_n^*$ as its response of group keys to its own ID challenger and in turn will receive $\lambda_1, \cdots, \lambda_n$. Upon *PointReg*1 queries $x_u \in \underline{x}_1^*$ (from $\mathcal{C}'$), $\mathbf{CStO}_s$ sets its random oracle value (recall that in $\mathbf{G}_5$-$\mathbf{G}_9$, the PointReg1 query for $x \in \underline{x}_1^*$ occurs when $\mathcal{D}$ issues the k_1th random oracle query, where the test measurement Π has the outcome $|\bot\rangle_{D_x}$ as it does not abort, and hence $D(x) = \bot$) $\mathcal{S}.RO(x_u)$ as λ_u $(u = 1, \cdots, n)$, which is provided by the ID challenger. In addition, later for $x_2^* = 1|\overline{pk'}|\alpha|M$, in *PointReg*1 query x_2^*, it sets the hash value $r = \text{CH}$, which is provided by the ID challenger. This will not change the distribution of the game, because λ_u for any u, and these CH are all uniformly random, remaining as the same distribution as in $\mathbf{G}_9$. When $\mathcal{D}$ outputs its forgery, if the output $(\mathbf{w}, \mathbf{y}) \in S$, then it sends the response Rsp in w to the ID challenger as its response. Obviously, $\mathcal{C}'$ succeeds in its ID challenge session if and only if $\mathcal{D}$ succeeds with $(w, \mathbf{y}) \in S$ (that is, the forgery is valid). Thus, the adversary success probability is the same as in $\mathbf{G}_9$, and hence, $\mathcal{C}'$ has a success probability negligibly close to $\frac{\epsilon}{(q+\binom{q}{3})^6}$. This contradicts the security of the ID scheme. □

Remark 7. *In* $\mathbf{G}_5$*, we convert the game with* **CStO** *to the game with* $\mathbf{CStO'}$*, where we register* $\underline{x}_t^*$ *to* Ξ_0 *at the* i_t*th oracle random oracle query, while it registers to* Ξ_1 *only at the* k_t*th random oracle query. This generally is the routine to convert* $\mathbf{Exp}_{i^c, j^c, k^c}$ *to a game with* $\mathbf{CStO'}$*. One might wonder why we register* $\underline{x}_t^*$ *twice. The issue in fact comes from the switch from* $\mathbf{CStO'}$ *to* $\mathbf{CStO}_s$ *in* $\mathbf{G}_6$*.* $\mathbf{CStO}_s$ *requires that after registration in* Ξ_1*, no measurement for testing* $D(x) = \bot$ *will be performed. If we register it once, this should happen at the* i_t*th query for* $\underline{x}_t^*$*. But in this case, we cannot guarantee that* $\mathbf{G}_5$ *(with* $\mathbf{CStO'}$*) will be indistinguishably switched to* $\mathbf{G}_6$ *with* $\mathbf{CStO}_s$*: after the* i_t*th query, we still need to measure if* $D(\underline{x}_t^*) = \bot$*. But in* $\mathbf{G}_6$*, this will never be true, as* $|\bot\rangle$ *is replaced by* $|r\rangle$*, while in* $\mathbf{G}_5$ *(with* $\mathbf{CStO'}$*), it is still possible. This distinguishing event does not violate Lemma 9, because this test is no longer performed in* $\mathbf{CStO}_s$ *after updating* $|\bot\rangle$ *by* $|r\rangle$*.*

9. Quantum Security of The JAK ID Scheme

In this section, we prove the quantum security of the lattice-based ID scheme in [23] (which we call the JAK ID scheme). Together with Theorem 5, it gives a secure lattice-based multi-signature in the quantum random oracle model. We will use the following notations:

- As a convention for lattice over ring, the security parameter is denoted by n (a power of 2);
- q is a prime with $q \equiv 3 \bmod 8$;
- $R = \mathbb{Z}[x]/(x^n+1)$; $R_q = \mathbb{Z}_q[x]/(x^n+1)$; R_q^* is the set of invertible elements in R_q;
- A vector $\mathbf{w}$ is implicitly a column vector, and the ith component is w_i or $\mathbf{w}[i]$;
- for a matrix or vector X, X^T is its transpose;
- $\mathbf{1}$ denotes the all one-vector $(1, \cdots, 1)^T$ value of a clear dimension only in the specific context;
- For $u = \sum_{i=0}^{n-1} u_i x^i \in R$, $||u||_\infty = \max_i |u_i|$;
- $\alpha \in \mathbb{Z}_q$ always uses the default representative with $-(q-1)/2 \le \alpha \le (q-1)/2$, and similarly, for $u \in R_q$, each coefficient of u by default belongs to this range;
- $e = 2.71828\cdots$ is the Euler's number;
- $\mathcal{C} = \{c \in R \mid ||c||_\infty \le \log n, \deg(c) < n/2\}$;
- $\mathcal{Y} = \{y \in R \mid ||y||_\infty \le n^{1.5}\sigma \log^3 n\}$;
- $\mathcal{Z} = \{z \in R \mid ||z||_\infty \le (n-1)n^{1/2}\sigma \log^3 n\}$.

9.1. Ring-LWE and Ring-SIS

In the following, we introduce the ring-LWE and ring-SIS assumptions (see [39–41] for details). For $\sigma > 0$, distribution $D_{\mathbb{Z}^n,\sigma}$ assigns the probability proportional to $e^{-\pi||\mathbf{y}||^2/\sigma^2}$ for any $\mathbf{y} \in \mathbb{Z}^n$ and 0 for other cases. As in [42], $y \leftarrow D_{R,\sigma}$ samples $y = \sum_{i=0}^{n-1} y_i x^i$ from R by taking $y_i \leftarrow D_{\mathbb{Z},\sigma}$.

The Ring Learning With Error (Ring-LWE$_{q,\sigma,2n}$) problem over R with standard deviation σ is defined as follows. Initially, it takes $s \leftarrow D_{R,\sigma}$ as secret. It then takes $a \leftarrow R_q, e \leftarrow D_{R,\sigma}$ and outputs $(a, as+e)$. The problem is to distinguish $(a, as+e)$ from a tuple (a, b) for $a, b \leftarrow R_q$. The Ring-LWE$_{q,\sigma,2n}$ assumption [43,44] is to say that no quantum polynomial time algorithm can solve Ring-LWE$_{q,\sigma,2n}$ problem with a non-negligible advantage.

The Small Integer Solution problem with parameters q, m, β over ring R (Ring-SIS$_{q,m,\beta}$) is as follows: given m uniformly random elements $a_1, \cdots, a_m$ over R_q, find $(t_1, \cdots, t_m)$ so that $||t_i||_\infty \le \beta$ and $a_1 t_1 + \cdots + a_m t_m = 0$. We consider the case $m = 3$. We assume that $q = 3 \bmod 8$, in which case, by Theorem 1 [45], $x^n + 1 = \Phi_1(x)\Phi_2(x)$ for irreducible polynomials $\Phi_1(x), \Phi_2(x)$ of degree $n/2$. So by the Chinese remainder theorem, a_i is invertible, except for probability $2q^{-n/2}$. Hence, ring-SIS is equivalent to the case of invertible a_2, which is further equivalent to problem $a_1 t_1 + t_2 + a_3 t_3 = 0$, as we can multiply it by a_2^{-1}. The quantum hardness of ring-SIS can be found in [39,46].

9.2. The JAK ID Scheme

We now review the JAK ID scheme [23]. Initially, take $s_1, s_2 \leftarrow D_{R,\sigma}, a_1, a_2 \leftarrow R_q$ and compute $u = a_1 s_1 + a_2 s_2$. The system parameter is (a_1, a_2); the public key is u and the private key is (s_1, s_2). The ID scheme is as follows (also see Figure 3):

1. Prover generates $\mathbf{y}_1, \mathbf{y}_2 \leftarrow \mathcal{Y}^\mu$ and computes $\mathbf{v} = a_1\mathbf{y}_1 + a_2\mathbf{y}_2$; it sends $\mathbf{v}$ to Verifier, where $\mu \ge \log^2 n$.
2. Receiver samples $c \leftarrow \mathcal{C}$ and sends it to Prover.
3. Upon c, Prover computes $z_1 = s_1 c + \sum_j y_{1j}$, $z_2 = s_2 c + \sum_j y_{2j}$.
4. Upon z_1, z_2, Verifier checks if $\sum_{i=1}^{\mu} v_i \overset{?}{=} a_1 z_1 + a_2 z_2 - uc$ and $||z_b||_\infty \overset{?}{\le} \eta_t$ for $b = 1, 2$, where $\eta_t = 5\sigma n^2 \sqrt{t\mu} \log^6 n$, and t is a positive integer (that represents the number of signers when converted to a signature scheme); recall that (as a convention) v_i is the ith component of $\mathbf{v}$. If all are valid, it accepts; otherwise, it rejects.

The above specification uses the public-key $u = a_1s_1 + a_2s_2$, while the original protocol uses $u = as_1 + s_2$. This change is only for convenience for our proof for Lemmas A3 (that is needed for the ID security). It will not affect other properties—correctness, simulatability, linearity, and classical security—as if we define $a = a_1a_2^{-1}$ (ignore the negligible probability $2q^{-n/2}$ that a_2 is not invertible: recall that $x^n + 1 = \Phi_1(x)\Phi_2(x)$ and a_2 are invertible if and only if they are non-zero modular Φ_1, Φ_2 both); the current version is different from the original one only by a scaling factor a_2 (in $\mathbf{v}$ and u), and all the proofs go through. Furthermore, Step 3 in the above specification is a simplified but equivalent version of the original protocol (see the remark after the scheme description in [23]). The proofs of the correctness and linearity do not involve the adversary and hence remain unchanged, as in [23]. The simulability given in [23] holds statistically. It hence holds against a quantum attacker, where the model is the same except that the attacker can internally run a quantum computer (which can be simulated by unbounded adversary).

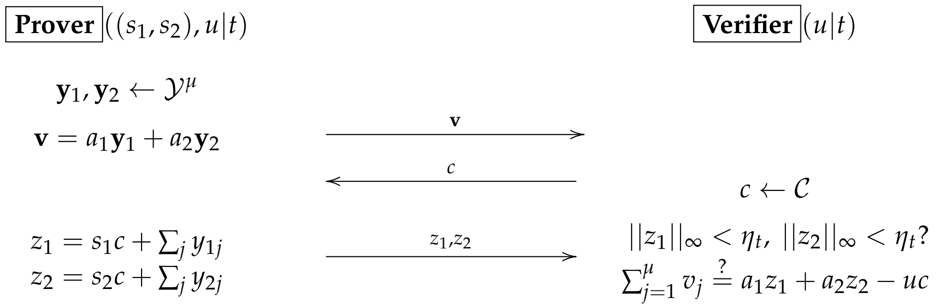
$\boxed{\textbf{Prover}}((s_1, s_2), u|t)$
$\boxed{\textbf{Verifier}}(u|t)$
$\mathbf{y}_1, \mathbf{y}_2 \leftarrow \mathcal{Y}^\mu$
$\mathbf{v} = a_1\mathbf{y}_1 + a_2\mathbf{y}_2$
$\mathbf{v}$
c
$c \leftarrow \mathcal{C}$
$z_1 = s_1c + \sum_j y_{1j}$
$z_2 = s_2c + \sum_j y_{2j}$
z_1, z_2
$||z_1||_\infty < \eta_t, \ ||z_2||_\infty < \eta_t?$
$\sum_{j=1}^{\mu} v_j \overset{?}{=} a_1z_1 + a_2z_2 - uc$

Figure 3. The JAK ID Scheme.

It remains to prove the quantum security of this ID scheme under Definition 5. The idea is to implement the classical rewinding technique in the quantum world. We start with the security game below with u_1 being the honest signer's public key. We first make the change that $\lambda_2, \cdots, \lambda_t$ are provided by attacker (which will increase the attacker A's success probability only):

1. $a_1, a_2 \leftarrow \textbf{Setup}(1^\lambda)$;
2. $(|st_0\rangle, \lambda_2, u_2, \cdots, \lambda_t, u_t) \leftarrow A(a_1, a_2, u_1)$;
3. $\lambda_1 \leftarrow \mathcal{C}$;
4. $(|st_1\rangle, \mathbf{v}) \leftarrow A(|st_0\rangle, \lambda_1)$;
5. $c \leftarrow \mathcal{C}$; $z_1|z_2 \leftarrow A(|st_1\rangle, c)$;
6. **Check**: $\sum_{j=1}^{\mu} v_j \overset{?}{=} a_1z_1 + a_2z_2 - \bar{u}c$, $||z_1||_\infty < \eta_t$, $||z_2||_\infty < \eta_t$?

In the proof in the classical model, we first obtain a transcript $(\{\lambda_i|u_i\}_{i=2}^t, \lambda_1, \mathbf{v}, c, z_1|z_2)$ and then rewind A to line 5 and produce another valid transcript $(\{\lambda_i|u_i\}_{i=2}^t, \lambda_1, \mathbf{v}, c', z_1'|z_2')$. This allows us to derive a short solution $(o_1, o_2, o_3) = (z_1 - z_1', z_2 - z_2', c - c')$ for equation $a_1o_1 + a_2o_2 - \bar{u}o_3 = 0$. In the quantum world, this rewinding strategy is not quite working, because when A produces z_1, z_2, it might do a measurement that is not reversible. If it only uses unitary (e.g., U), then the rewinding can be done by applying $U^\dagger$. Unruh [34] introduced the notion of the collapsing property for a protocol: even with the measurement, the rewinding still can produce a successful new transcript with a good probability. In our quantum security proof, we will guarantee that this property is satisfied. Next, we rewind A to step 3 with a new challenge λ_1' and repeat the above procedure to obtain a new solution (o_1', o_2', o_3') satisfying $a_1o_1' + a_2o_2' - \overline{u'}o_3' = 0$, where $\overline{u'}$ is updated as $u_1\lambda_1' + \sum_{i=2}^t \lambda_iu_i$. Combining these two solutions allows us to derive a short solution (x_1, x_2, x_3) for $a_1x_1 + a_2x_2 + u_1x_3 = 0$. If u_1 is uniformly random in R_q, this is the solution for Ring-SIS. However, even though u_1 is sampled as $a_1s_1 + a_2s_2$, it is indistinguishable from the uniformly random u_1 by Ring-LWE assumption. Since the secret (s_1, s_2) is never used in the above game, if we use the uniformly random u_1 in the game, we can obtain the solution (x_1, x_2, x_3) with the similar probability. This contradicts the Ring-SIS assumption. The detailed implementation of this strategy is given Appendix A.

Theorem 6. *Under ring-LWE$_{q,\sigma,2n}$ and ring-SIS$_{3,q,\beta}$ assumptions, the JAK ID scheme is secure (under Definition 5), where $\beta \geq 16\eta_t\sqrt{n}\log^2 n$.*

Applying the compiler theorem to the JAK ID scheme, it gives a quantum-secure multi-signature scheme (denoted by RLWE-Multisig scheme). For a complete description of this scheme, see [23]. The following is a summary of its security.

Corollary 4. *Under Ring-LWE$_{q,\sigma,2n}$ and Ring-SIS$_{3,q,\beta}$ assumptions,* RLWE-MultiSig *is EU-CMA secure in the quantum random oracle model, where $\beta \geq 16\eta_t\sqrt{n}\log^2 n$.*

10. Conclusions

In this paper, we investigated the security analysis techniques in the quantum random oracle model. We extended Zhandry's compressed random oracle **CStO** to a compressed random oracle with adaptive special points (**CStO**$_s$). In **CStO**$_s$, we can set the random oracle value at the special point to whatever we want, which is well known to be a powerful property in a classical random oracle model. We extended the sampling experiment of Liu and Zhandry that identifies special points in **CStO**, witnessing the future adversarial output that can be easily converted to a game with **CStO**$_s$. We also extended the online query extraction technique of Don et al. [32] from **CStO** to the **CStO**$_s$ setting, which allows us to extract the input to any adversarial commitment on the fly, just as we can do in a classical random oracle model. We applied this new random oracle and its extraction techniques to prove the security of our recent compact multi-signature scheme. This gives the first compact multi-signature provable secure in the quantum random oracle model. We believe that this random oracle technique will be useful to prove the post-quantum security of many cryptographic systems. To realize the quantum secure multi-signature framework, we proved the quantum security of the ID scheme in [23]. Our strategy is to derive two public coin protocols from that ID scheme and prove that they are weakly collapsing (in the sense of [33]), as well as iteratively apply Unruh's quantum rewinding lemma [34] to reduce the security of the ring-SIS problem.

There are several questions deserving further investigations. First, our conversion from the **StO** and **CStO** model to the **CStO**$_s$ model was through the sampling experiment in the **CStO** model. It degrades the adversarial success probability by a factor of order $O(q^{-6c})$ (Theorem 4), where q is the number of oracle queries, and c is the number of witness for the final adversarial output. This factor will carry to the overall reduction advantage in a security proof. It is interesting to know if one can find a new method that bridges **CStO** and **CStO**$_s$ with a much better factor. It is even more interesting to know if one can find a new random oracle model so that it is much simpler than **CStO**$_s$, and the transition from **CStO** to this model has much less security loss. Second, the proof of JAK ID security has applied Unruh's lemma twice and results in a successful probability of order $O(\epsilon^6)$ if the adversary has a success probability ϵ in breaking the original ID scheme. In general, if it applies this lemma k times, then the resulting success probability will reduce to the order of $O(\epsilon^{3k})$. An interesting open question is to find a *polynomial* strategy with a significantly better success probability. Third, the JAK ID scheme needs to combine $\mu = \omega(\log n)$ copies of element ID executions. It will be interesting if this μ can be dramatically reduced.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study.

Acknowledgments: The author would like to thank all reviewers for their invaluable comments that helped significantly improve the quality of this paper.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A. Proof of Theorem 6

In this appendix, we will prove the security of the JAK ID scheme. Before this, we first define a *public-coin protocol*, which is a simple generalization of a sigma protocol.

Definition A1. *A n-round public-coin protocol Σ is a tuple of algorithms $(\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ that executes as follows:*

- *Initially, $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ is executed so that pk is given to $\mathcal{P}$ and $\mathcal{V}$ as a public key, and sk is given to $\mathcal{P}$ as a private key. $\mathcal{P}$ has an initial state $st_P = pk|sk$, while $\mathcal{V}$ has an initial state $st_V = pk$.*
- *The protocol proceeds in n rounds. In round $\ell = 1, \cdots, n$, $\mathcal{P}$ executes $a_\ell \leftarrow \mathcal{P}.com_\ell(st_P, c_{\ell-1})$ and sends it to $\mathcal{V}$, where $c_0 = nil$. For $\ell < n$, $\mathcal{V}$ replies with a challenge $c_\ell \leftarrow \Theta_\ell$. For $\ell = n$, $\mathcal{V}$ runs $\mathcal{V}.ver(pk, a_1|c_1|\cdots|a_n)$, and outputs 0 (for reject) or 1 (for accept).*

Appendix A.1. Collapsing Public-Coin Protocol

For any quantum polynomial time distinguisher $\mathcal{D}$, we define a collapsing game $\mathsf{clpsExp}(\mathcal{D})$ between $\mathcal{D}$ and a challenger Chal with respect to an n-round public-coin protocol $\Sigma = (\mathbf{Gen}, \mathcal{P}, \mathcal{V})$:

- Initially, Chal generates pk and gives it to $\mathcal{D}$.
- Then, $\mathcal{D}$ (in the role of $\mathcal{P}$) and Chal (in the role of $\mathcal{V}$) execute the protocol Σ, except for round n. At round n, $\mathcal{D}$ generates a quantum superposition $|\phi\rangle$ (over the response a_n), which might be entangled with states in extra registers. It then provides $|\phi\rangle$ to Chal.
- Upon $|\phi\rangle$, Chal uses a measurement to check if a_n in $|\phi\rangle$ is a valid response for $a_1|c_1|\cdots|a_{n-1}|c_{n-1}$. If the verification fails, Chal aborts; otherwise, let $|\phi'\rangle$ be the superposition containing all the valid a_n's. Then, Chal flips a coin $b \leftarrow \{0,1\}$. If $b = 0$, it does nothing; otherwise, it measures $|\phi'\rangle$ in the computational basis. Finally, it sends the resulting superposition back to $\mathcal{D}$.
- Finally, $\mathcal{D}$ outputs a guess bit b' for b, which is also set as the output of the game.

We use $\mathsf{clpsExp}^b_{\mathcal{D}}$ to denote the game with challenge bit b.

Definition A2. *A Σ-protocol is collapsing if*

$$\Pr(\mathsf{clpsExp}^1_{\mathcal{D}} = 0) = \Pr(\mathsf{clpsExp}^0_{\mathcal{D}} = 0) + \mathbf{negl}(\lambda). \tag{A1}$$

It is γ-weakly collapsing if

$$\Pr(\mathsf{clpsExp}^1_{\mathcal{D}} = 0) \geq \gamma \cdot \Pr(\mathsf{clpsExp}^0_{\mathcal{D}} = 0) - \mathbf{negl}(\lambda). \tag{A2}$$

Remark A1. *This definition was extended from [33] for the Sigma protocol to a general public-coin protocol. In this definition, the collapsing property states that no attacker can detect whether the final round is a superposition or a classical response by measuring the former. This property is concerned only with the last round, and all the previous $n-1$ prover messages are still classic.*

Appendix A.2. Two Public-Coin Protocols from Our ID Scheme

We define two public-coin protocols Σ_1 and Σ_2 between quantum algorithm A and challenger, which are derived from the JAK ID protocol. We keep the notations and their computations as in Section 9.2 unless specified.

Appendix A.2.1. Protocol Σ_1

Let $u_1, a_1, a_2 \leftarrow R_q$. A interacts with a challenger as follows:

1. A sends $(\lambda_2, u_2, \cdots, \lambda_t, u_t)$ to the challenger and holds a state $|\psi_1\rangle$, where $\lambda_i \leftarrow \Theta$.
2. The challenger sends $\lambda_1 \leftarrow \Theta$ to A.

3. A applies a unitary U_{λ_1} to $|\psi_1\rangle$ and results in $\sum_{o,\psi_o} |o, \psi_o\rangle$. It measures $o = (o_1, o_2, o_3)$ in the computational basis and sends it to the challenger.
4. The challenger accepts if $a_1 o_1 + a_2 o_2 - \bar{u} o_3 = 0$ and $||o_i||_\infty \leq 2\eta_t$ for $i = 1, 2, 3$, where $\bar{u} = \sum_{i=1}^{t} \lambda_i u_i$.

Appendix A.2.2. Protocol Σ_2

Let $u_1, a_1, a_2 \leftarrow R_q$. A interacts with the challenger as follows:

1. A sends $(\lambda_2, u_2, \cdots, \lambda_t, u_t)$ to the challenger, where $\lambda_i \leftarrow \Theta$.
2. The challenger sends $\lambda_1 \leftarrow \Theta$ to A.
3. A computes and sends $\mathbf{v} \in R_q^{\mu}$ to the challenger and also prepares a state $|\psi_1\rangle$.
4. The challenger replies with $c \leftarrow \Theta$.
5. A applies a unitary $V_{\lambda_1 c}$ to its state $|\psi_1\rangle$ and results in $\sum_{\mathbf{z},\psi_{\mathbf{z}}} |\mathbf{z}, \psi_{\mathbf{z}}\rangle$, where, although not stated,$V_{\lambda_1 c}$ also depends on the previous messages. It measures $\mathbf{z} = (z_1, z_2)$ in the computational basis and sends it to the challenger.
6. The challenger accepts if $\sum_{i=1}^{\mu} v_i = a_1 z_1 + a_2 z_2 - \bar{u} c$ and $||z_1||_\infty \leq \eta_t, ||z_2||_\infty \leq \eta_t$.

Appendix A.3. Security of the JAK ID Scheme When Σ_1 and Σ_2 Are Weakly Collapsing

In the following, we prove that the JAK ID is secure (with repect to Definition 5) based on the assumptions that Σ_1 and Σ_2 are both weakly collapsing. This proof is threaded by two observations.

First, in Σ_2, if we can rewind the execution to the beginning of Step 4 easily, then we can obtain two tuples (z_1, z_2, c) and (z_1', z_2', c'), with short z_1, z_2, z_1', z_2' satisfying

$$\sum_{i=1}^{\mu} v_i = a_1 z_1 + a_2 z_2 - \bar{u} c, \quad \sum_{i=1}^{\mu} v_i = a_1 z_1' + a_2 z_2' - \bar{u} c'. \tag{A3}$$

This gives a solution (o_1, o_2, o_3) with short o_i (as c, c' are also short) so that $a_1 o_1 + a_2 o_2 - \bar{u} o_3 = 0$. If Step 5 was completely done using a unitary operator (say, U), then the rewinding is just to apply $U^\dagger$. Unfortunately, it has a measurement for (z_1, z_2) that makes the rewound execution unpredictable. Fortunately, The weakly collapsing property of Σ_2 can be used to show that even if it measures (z_1, z_2), the rewinding by $V_{\lambda_1 c}^\dagger$ only (that is, we ignore the impact by the measurement of (z_1, z_2)) can still produce two accepting tuples (z_1, z_2, c) and (z_1', z_2', c') with a good probability.

Second, in Σ_1, if we can rewind the execution to the beginning of Step 2, we obtain two solutions $(o_1, o_2, o_3, \lambda_1)$ and $(o_1', o_2', o_3', \lambda_1')$ so that

$$a_1 o_1 + a_2 o_2 - \bar{u} o_3 = 0, \quad a_1 o_1' + a_2 o_2' - \bar{u'} o_3' = 0, \tag{A4}$$

where $\bar{u'} = \lambda_1' u_1 + \sum_{i=2}^{t} \lambda_i u_i$. This allows us to derive a short solution (t_1, t_2, t_3) for $a_1 t_1 + a_2 t_2 + u t_3 = 0$, which is in contradiction to the ring-SIS assumption. Again, due to the weakly collapsing property of Σ_1, this rewinding with measuring (o_1, o_2, o_3) can still succeed with good probability compared to the rewinding without measuring (o_1, o_2, o_3).

With these observations, we can now return the ID security game (Definition 5). We notice that this game can be formulated as Σ_2. On the other hand, Σ_1 can be regarded as the internal execution of Σ_2 after Step 2, the rewinding of which gives a solution (o_1, o_2, o_3). This leads to an attack for ring-SIS: the attacker runs A to run Σ_2 to produce (o_1, o_2, o_3) and with rewinding, it produces another (o_1', o_2', o_3'). As seen above, this gives a solution to the ring-SIS problem. This contradicts the ring-SIS assumption.

Lemma A1. *If Σ_1 is γ_1-weakly collapsing and Σ_2 is γ_2-weakly collapsing, then under ring-LWE$_{q,\sigma,2n}$ and ring-SIS$_{3,q,\beta}$ assumptions, the JAK ID scheme is secure, where $\beta \geq 16\eta_t\sqrt{n}\log^2 n$.*

Proof. Assume that A has a success probability ϵ in the security game of an ID scheme (see Definition 5). We revise the game so that u_1 is uniformly random over R_q (instead of $u_1 = a_1 s_1 + a_2 s_2$ which is indistinguishable from uniformly random over R_q under the ring-LWE assumption, as a_2 is invertible in R_q except for a negligible probability). Then, by the ring-LWE assumption, the success of A is changed only negligibly. Furthermore, we change the game so that A chooses $\lambda_2, \cdots, \lambda_t$. This will only increase the success of A. Finally, we change the game so that A is unitary (whenever operating on its quantum state) except when it needs to measure its state to produce a protocol message (in the computational basis). This does not change the success probability of A, as any A can always be made into this kind without changing its output distribution by adding more ancilla registers and also applying the deferred measurement principle. Now, the security game is simply Σ_2. For brevity, we still assume that A can succeed with probability ϵ. Let τ be the partial transcript $(u_1, a_1, a_2, \{u_i, \lambda_i\}_{i=2}^{t}, \lambda_1, \mathbf{v})$. Let ω_τ be the probability of τ. For a fixed τ, let $P_{\tau c}$ be the projection to the subspace from all $|z_1, z_2\rangle\langle z_1, z_2|$ so that $(\mathbf{v}, c, (z_1, z_2))$ is accepting. Furthermore, let ϵ_τ be the accepting probability (over c), given the partial transcript τ. We modify Σ_2 to Σ_2' so that A does not measure (z_1, z_2), and instead, it only measures $P_{\tau c}$. It is not hard to see that A in Σ_2' and Σ_2 has the same success probability ϵ (by Lemma 2(2)). Let $|\psi_\tau\rangle$ be the state after A sending $\mathbf{v}$. Then, $\epsilon_\tau = \frac{1}{|\Theta|}\sum_{c\in\Theta} ||V_{\tau c}^{\dagger} P_{\tau c} V_{\tau c}|\psi_\tau\rangle||^2$, and $\epsilon = \sum_\tau \omega_\tau \epsilon_\tau$. Define $\tilde{P}_{\tau c} = V_{\tau c}^{\dagger} P_{\tau c} V_{\tau c}$. Before moving on, we recall a claim from Lemma 7 [34].

Claim 1. *Let E be a set. Let $(Q_e)_{e\in E}$ be orthogonal projectors on Hilbert space $\mathcal{H}$. Let $|\Phi\rangle \in \mathcal{H}$ be a unit vector. Let $V = \sum_{e\in E} \frac{1}{|E|}||Q_e|\Phi\rangle||^2$ and $F = \sum_{e_1, e_2 \in E} \frac{1}{|E|}||Q_{e_1} Q_{e_2}|\Phi\rangle||^2$. Then, $F \geq V^3$.*

From this claim, we have that $\frac{1}{|\Theta|^2}\sum_{c', c\in\Theta} ||\tilde{P}_{\tau c'}\tilde{P}_{\tau c}|\psi_\tau\rangle||^2 \geq \epsilon_\tau^3$. This is the probability that we rewind A in Σ_2', after $P_{\tau c}$ projection, to produce a second response (z_1', z_2') using challenge c'. If we require $c' \neq c$, then this probability will change to $\epsilon_\tau^3 - \epsilon_\tau/|\Theta|$, as $\tilde{P}_{\tau c'}\tilde{P}_{\tau c} = \tilde{P}_{\tau c}$ when $c' = c$.

Now consider this success probability in Σ_2 (not Σ_2') when $c' \neq c$, where the projective measurement for (z_1, z_2) after $P_{\tau c}$ and the projective measurement for (z_1', z_2') after $P_{\tau c'}$ will be applied. By the γ-weakly collapsing property of Σ_2, it is easy to show that this probability is at least $\gamma_2^2(\epsilon_\tau^3 - \epsilon_\tau/|\Theta|)$ (similar to [33, Lemma 5] and the analysis right after it). Therefore, Σ_2 rewindings produce two accepting transcripts (c, z_1, z_2) and (c', z_1', z_2') for $c' \neq c$, with probability being at least $\gamma_2^2(\epsilon_\tau^3 - \epsilon_\tau/|\Theta|)$. Notice that these two accepting transcripts will result in a witness $(o_1, o_2, o_3) = (z_1 - z_1', z_2 - z_2', c - c')$ so that $a_1 o_1 + a_2 o_2 - \bar{u} o_3 = 0$. When $\tau' = (u_1, a_1, a_2, \{u_i, \lambda_i\}_{i=2}^{t})$ is fixed, this occurs with a probability of at least $\sum_{\lambda_1 \mathbf{v}} P_{\lambda_1 \mathbf{v}|\tau'} \gamma_2^2(\epsilon_{\tau'\lambda_1\mathbf{v}}^3 - \epsilon_{\tau'\lambda_1\mathbf{v}}/|\Theta|) \geq \gamma_2^2(\epsilon_{\tau'}^3 - \epsilon_{\tau'}/|\Theta|)$ by the Cauchy–Schwarz inequality, where $\epsilon_{\tau'} = \mathbf{E}_{\lambda_1\mathbf{v}}(\epsilon_{\tau'\lambda_1\mathbf{v}}|\tau')$, and marginal probability $P_{\tau'} = \sum_{\lambda_1\mathbf{v}} P_{\tau'\lambda_1\mathbf{v}}$ is the occurrence of τ'.

We then modify A in Σ_2 to an attacker A' for Σ_1: in Σ_1, A' follows A to prepare the Step 1 message, and after receiving λ_1, it makes use of A in Σ_2 in the above rewinding technique (where the challenges c', c are sampled randomly) to produce (o_1, o_2, o_2). We then modify A' so that it defers the measurements (after receiving λ_1) other than measuring (o_1, o_2, o_3) to the end of the game (where A' has already produced (o_1, o_2, o_3)). This does not change the success probability of A' by the deferred measurement principle (with some ancilla registers as in Corollary 1 extended from Lemma 7). Next, we modify A' so that A' does not do the deferred measurements mentioned above. This does not change the success probability of A', as the deferred measurements are done after (o_1, o_2, o_3) are obtained. Let $\epsilon_{\tau'}'$ be the success probability of this A' that produces (o_1, o_2, o_3) with short (o_1, o_2, o_3) so that $a_1 o_1 + a_2 o_2 - \bar{u} o_3 = 0$ with $||o_i||_\infty \leq 2\eta_t$. By our foregoing argument, $\epsilon_{\tau'}' \geq \gamma_2^2(\epsilon_{\tau'}^3 - \epsilon_{\tau'}/|\Theta|)$. Let $|\psi_{\tau'\lambda_1}\rangle$ be the state right before the projective measurement that results in (o_1, o_2, o_3) and $Q_{\tau'\lambda_1}$ be the test measurement on $|\psi_{\tau'\lambda_1}\rangle$ to check if $a_1 o_1 + a_2 o_2 - \bar{u} o_3 = 0$. Let A'' be the variant of A' so that projective measure resulting in (o_1, o_2, o_3) is not made and instead only makes the test measurement $Q_{\tau'\lambda_1}$. Under this, A'' still has the success probability $\epsilon'_{\tau'}$. Let the unitary that produces $|\psi_{\tau'\lambda_1}\rangle$ be $U_{\tau'\lambda_1}$. Then, using the claim above, we similarly have that $\frac{1}{|\Theta|^2}\sum_{\lambda_1,\lambda'_1}||\tilde{Q}_{\tau'\lambda'_1}\tilde{Q}_{\tau'\lambda_1}|\psi_{\tau'}\rangle||^2 \geq \epsilon'^3_{\tau'}$, where $\tilde{Q}_{\tau'\lambda_1} = U^{\dagger}_{\tau'\lambda_1}Q_{\tau'\lambda_1}U_{\tau'\lambda_1}$. Furthermore, if we require $\lambda_1 \neq \lambda'_1$, then $\frac{1}{|\Theta|^2}\sum_{\lambda_1\neq\lambda'_1}||\tilde{Q}_{\tau'\lambda'_1}\tilde{Q}_{\tau'\lambda_1}|\psi_{\tau'}\rangle||^2 \geq \epsilon'^3_{\tau'} - \epsilon'_{\tau'}/|\Theta|$. Again, by applying the weakly collapsing property of Σ_1, if A'' does the measurement for (o_1, o_2, o_3) after $Q_{\tau'\lambda_1}$ and the measurement for (o'_1, o'_2, o'_3) after $Q_{\tau'\lambda'_1}$, then the success probability producing successful (o_1, o_2, o_3) and (o'_1, o'_2, o'_3) with probability at least $\gamma_1^2(\epsilon'^3_{\tau'} - \epsilon'_{\tau'}/|\Theta|) \geq \gamma_1^2(\epsilon'^3_{\tau'} - 1/|\Theta|)$. Since $\epsilon'_{\tau'} \geq \gamma_2^2(\epsilon^3_{\tau'} - \epsilon_{\tau'}/|\Theta|)$, averaging over τ' and using the Cauchy–Schwarz inequality, the success probability to produce two accepting (o_1, o_2, o_3) and (o'_1, o'_2, o'_3) with $\lambda_1 \neq \lambda'_1$ is at least $\gamma_1^2(\gamma_2^6(\epsilon^3 - \epsilon/|\Theta|)^3 - 1/|\Theta|)$. Since γ_1, γ_2 and ϵ are all non-negligible, this lower bound is also non-negligible. However, (o_1, o_2, o_3) and (o'_1, o'_2, o'_3) with $\lambda_1 \neq \lambda'_1$ lead to a solution (x_1, x_2, x_3) for the ring-SIS problem $a_1x_1 + a_2x_2 + u_1x_3 = 0$ (see Equations (36)–(38) in [23], where our length bound β for $||x_i||_\infty$ is summarized from there). This contradicts the ring-SIS $_{q,n,\beta}$ assumption! □

Appendix A.4. Σ_2 and Σ_1 Are Weakly Collapsing

In this section, we prove that Σ_2 and Σ_1 are weakly collapsing. We will rely on the notation of the compatible lossy function. We extend the compatible lossy function of an n-round public-coin protocol from [33] for a sigma protocol.

Definition A3. *A compatible lossy function for an n-round public-coin protocol* $\Sigma = (\mathbf{Gen}, \mathcal{P}, \mathcal{V})$ *is an efficiently computable function generator* $\mathsf{CLF.gen}(\lambda, pk, sk, \{a_i|c_i\}_{i=1}^{n-1}, \mathsf{mode})$*, which takes* λ *(security parameter), pk, sk, partial transcripts* $\{a_i|c_i\}_{i=1}^{n-1}$ *in* Σ *and the* mode *(either constant or injective), and outputs an efficiently computable function f so that we have the following:*

- constant *mode: Let the domain of f be all r, with* $\{a_i|c_i\}_{i=1}^{n-1}|r$ *being a valid transcript when* $a_n = r$*. Then, the probability that f has an image size of at most p is at least* γ*. That is,* $\Pr_f(Im(f) \leq p) \geq \gamma$ *for* $f \leftarrow \mathsf{CLF.gen}(\lambda, pk, sk, \{a_i|c_i\}_{i=1}^{n-1}, \mathsf{constant})$.
- injective *mode: For* $f \leftarrow \mathsf{CLF.gen}(\lambda, pk, sk, \{a_i|c_i\}_{i=1}^{n-1}, \mathsf{injective})$*, f is injective over all r such that* $(\{a_i|c_i\}_{i=1}^{n-1}|r)$ *is a valid transcript when* $a_n = r$*, except for a negligible probability.*
- indistinguishability. *We first define game* $\mathsf{clfExp}^b_{\mathcal{D},pk,sk}$ *for* $b = 0, 1$.
  - $\mathcal{D}$ *is given pk, and challenge* Chal *has pk, sk.*
  - $\mathcal{D}$ *(in the role of* $\mathcal{P}$*) and* Chal *(in the role of* $\mathcal{V}$*) execute* Σ *in the first* $n-1$ *rounds, resulting in the partial transcript* $\{a_i|c_i\}_{i=1}^{n-1}$.
  - *If* $b = 0$*, let* mode = constant*; otherwise,* mode=injective*. Then, challenger samples*

    $$f \leftarrow \mathsf{CLF.gen}(\lambda, pk, sk, \{a_i|c_i\}_{i=1}^{n-1}, \mathsf{mode})$$

    and provides it to $\mathcal{D}$*. Then,* $\mathcal{D}$ *outputs a guess bit* b' *for b, which is also defined as the output of the game.*

  The function generator CLF.gen *is* (p, γ)**-compatible** *with respect to* Σ *if, for any polynomial time quantum algorithm* $\mathcal{D}$ *and for* $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$*, we have*

  $$\Pr(\mathsf{clfExp}^0_{\mathcal{D},pk,sk} = 0) = \Pr(\mathsf{clfExp}^1_{\mathcal{D},pk,sk} = 0) + \mathbf{negl}(\lambda). \tag{A5}$$

The following lemma is adapted from Liu and Zhandry [33, Lemma 1], which shows that the existence of a compatible function for Σ implies that Σ is weakly collapsing. The result is stated with respect to a quantum secure sigma protocol. But, their proof does not require the quantum security of the sigma protocol and can also be trivially extended to an n-round public-coin protocol. Thus, we state it without a proof.

Lemma A2. *[33] If an n-round public-coin protocol* Σ *has a* (p, γ)*-compatible lossy function, then* Σ *is* γ/p*-weakly collapsing.*

In the following, we prove that Σ_2 has a compatible lossy function.

Lemma A3. *Let $\mathcal{F}_0$ and $\mathcal{F}_1$ with respect to $a_1|a_2|\{u_i|\lambda_i\}_{i=1}^t|\mathbf{v}|c$ in Σ_2 be two distributions of function families: for each valid $(z_1, z_2) \in R_q^2$ (with respect to $\{u_i|\lambda_i\}_{i=1}^t|\mathbf{v}|c$),*

$$\mathcal{F}_0 = \{f \ | \ f(z_1, z_2) = \lfloor (\mathbf{s}(a_1, a_2) + \mathbf{e})(z_1, z_2)^T + \mathbf{r} \rceil_\theta, \mathbf{s} \leftarrow R_q^{2\log n}, \mathbf{e} \leftarrow D_{R,\sigma}^{2\log n \times 2}, \mathbf{r} \leftarrow R_q^{2\log n}\}$$
$$\mathcal{F}_1 = \{f \ | \ f(z_1, z_2) = \lfloor \mathbf{B}(z_1, z_2)^T + \mathbf{r} \rceil_\theta, \mathbf{B} \leftarrow R_q^{2\log n \times 2}, \mathbf{r} \leftarrow R_q^{2\log n}\},$$

where $8\sigma n \eta_t^{1.5} \log n < \theta < \frac{q}{n \log n}$ and $\lfloor \mathbf{x} \rceil_\theta$ for $\mathbf{x} \in R_q^2$ round each coefficient $x_i \in \mathbb{F}_q$ (when representing $\mathbf{x}$ as a vector in $\mathbb{F}_q^{2n}$) using the $\lfloor x \rceil_\theta$ function: they first repsent $x = k\theta + y$ with $y \in (-\theta/2, \theta/2]$ and $k \in \mathbb{Z}$ and then output $k\theta$. Then, $\mathcal{F}_0$ and $\mathcal{F}_1$ are $(\frac{2^6}{3^6}, 1)$-compatible with respect to Σ_2.

Proof. First, we show that $\mathcal{F}_0$ is a constant function family; second, we show that $\mathcal{F}_1$ is an injective function family; finally, we show that they are indistinguishable. In Σ_2, the message flows in order are $\{\lambda_i|u_i\}_{i=2}^t$, λ_1, $\mathbf{v}$, c, and (z_1, z_2). The transcript is valid if $||z_1||_\infty < \eta_t$ and $||z_2||_\infty < \eta_t$ and $\sum_{i=1}^{\mu} v_i = a_1 z_1 + a_2 z_2 - \bar{u}c$, where $\bar{u} = \sum_{i=1}^t \lambda_i u_i$.

To show that $\mathcal{F}_0$ is a constant function family, we first show that

$$\mathcal{F}_0' = \{f \ | \ f(z_1, z_2) = \lfloor \mathbf{s}(a_1, a_2)(z_1, z_2)^T + \mathbf{r} \rceil_\theta, \mathbf{s} \leftarrow R_q^{2\log n}, \mathbf{r} \leftarrow R_q^{2\log n}\} \tag{A6}$$

is a constant function family for Σ_2. Indeed, the transcript is valid: $f(z_1, z_2) = \lfloor \mathbf{r} + \mathbf{s}(\sum_i v_i + \bar{u}c) \rceil_\theta$ (invariant). Then, we continue to show that $\mathcal{F}_0$ is a constant function family. The strategy is to show that there is a constant probability that

$$\lfloor \mathbf{r} + \mathbf{s}(\sum_i v_i + \bar{u}c) \rceil_\theta = \lfloor \mathbf{s}(a_1, a_2)(z_1, z_2)^T + \mathbf{r} + \mathbf{e}(z_1, z_2)^T \rceil_\theta, \forall \text{valid } (z_1, z_2). \tag{A7}$$

Since the left side is constant, $\mathcal{F}_0$ is a constant family. Now, we implement this strategy.

Claim. *Let $\sigma > \omega(\sqrt{n})$. For $e \leftarrow D_{R,\sigma}$ and $z \in R_q$ with $||z||_\infty < \eta_t$, then $\Pr(||ez||_\infty \geq \eta_t^{1.5}\sigma) < n \cdot \exp(-\pi\eta_t)$.*

Proof. Notice that the ith component of $ez \in R_q$ is $\sum_{j=0}^{n-1} \pm e_j z_{i-j}$, where $i - j$ means $(i - j) \bmod n$, and the sign is - when $i < j$ and is + otherwise. By Lemma 4.4 [47], $\Pr(|\sum_{j=0}^{n-1} \pm e_j z_{i-j}| > \sigma ||z||_\infty \sqrt{\eta_t}) < e^{-\pi\eta_t}$. The union bound on i gives the result. □

Back to our proof, the above claim implies that

$$\Pr(||e_{b1}z_1 + e_{b2}z_2||_\infty > 2\sigma\eta_t\sqrt{\eta_t} : \exists b \in [2\log n]) < 2n \log n \cdot \exp(-\pi\eta_t). \tag{A8}$$

The space of $x \in R_q$ with $||x||_\infty \leq \eta_t$ has a size of at most $(2\eta_t)^n$. Since $||z_1||_\infty \leq \eta_t$ and $||z_2||_\infty \leq \eta_t$, (z_1, z_2) has at most $(2\eta_t)^{2n}$ choices. By the union bound, $||e_{b1}z_1 + e_{b2}z_2||_\infty > 2\sigma\eta_t\sqrt{\eta_t}$ for some (z_1, z_2, b) only has an exponentially small probability (over $(\mathbf{e}_1, \mathbf{e}_2)$), as $\eta_t = \omega(n \log n)$. Assume that $||e_{b1}z_1 + e_{b2}z_2||_\infty \leq 2\sigma\eta_t^{1.5}$ holds for any (b, z_1, z_2). Notice that $\mathbf{w} := \mathbf{s}(a_1, a_2)(z_1, z_2)^T + \mathbf{r}$ is uniformly random in $R_q^{2\log n}$ (as $\mathbf{r}$ is). For $x \in R_q$, we use $\underline{x}$ to denote the coeffient vector of x over $\mathbb{F}_q$. Similarly, for a vector $\mathbf{x} \in R_q^\ell$, we still use $\underline{\mathbf{x}}$ to denote the concatenated vector from $\underline{x_i}$ for all $i = 1, \cdots, \ell$ and use $\underline{\mathbf{x}}[j]$ to denote the jth coordinate in $\underline{\mathbf{x}}$. Then, $\underline{\mathbf{w}}$ is uniformly random over $\mathbb{F}_q^{2n\log n}$. If all $\underline{\mathbf{w}}[i]$ modes of θ belong to $(-\theta/2 + 2\sigma\eta_t^{1.5}, \theta/2 - 2\sigma\eta_t^{1.5})$, then $\lfloor \underline{\mathbf{w}}[i] \rceil_\theta = \lfloor \underline{\mathbf{w}}[i] + \underline{(\mathbf{e}_1, \mathbf{e}_2)(z_1, z_2)^T}[i] \rceil_\theta$ for all i. By a simple calculation, the statistical distance between $\underline{\mathbf{w}}[i]$ modes of θ and the uniform distribution over $(-\theta/2, \theta/2)$ is at most $\frac{\theta}{2q}$. Hence, $\underline{\mathbf{w}}[i]$ modes of θ is in that interval for all i with probability at least $(1 - \frac{4\sigma\eta_t^{1.5}}{\theta} - \frac{\theta}{2q})^{2n\log n} \geq (1 - \frac{1}{n\log n})^{2n\log n}$, which is at least $2^6/3^6$ by our assumption on θ due to the fact that $(1 - 1/x)^x$ is increasing when $x \geq 3$.

This indicates that $(\mathbf{e}_1, \mathbf{e}_2)(z_1, z_2)^T$ does not change the value of $f(z_1, z_2)$. In addition, $\mathbf{w}$ is unchanged over all valid (z_1, z_2) (as seen in $\mathcal{F}_0'$). Hence, f is constant, which occurs with probability at least $2^6/3^6$.

Next, we prove that $\mathcal{F}_1$ is injective. That is, $\mathbf{B}(z_1, z_2) + \mathbf{r}$ is injective. Indeed, $\mathbf{B}$ is invertible if $\det(B)$ is invertible in R_q, where B is $\mathbf{B}_i \in R_q^{2\times 2}$ for some i while $\mathbf{B} = (\mathbf{B}_i)_{i=1}^{\log n}$. Let $B = (a_{ij})_{i,j=1,2}$. If a_{11} is invertible, we can use Gaussian ellimination to make entry (1, 2) zero and a_{22} be updated as $a_{22}' = a_{22} - a_{11}^{-1}a_{12}$, which is still uniformly random in R_q. Furthermore, since $x^n + 1 = \Phi_1(x)\Phi_2(x)$ with $\Phi_1(x), \Phi_2(x)$ is an irreducible of degree $n/2$, a random element in R_q is inveritble with a probability of $1 - 2q^{-n/2}$ by the Chinese Remainder Theorem. Thus, B is invertible with probability at least $1 - 4q^{-n/2}$. Thus, the statement that no $\mathbf{B}_i$ is invertible, has a negligible probability.

Finally, we prove that $\mathcal{F}_0$ and $\mathcal{F}_1$ are indistinguishable. This directly follows from the ring-LWE assumption, as $s_b(a_1, a_2) + (e_{b1}, e_{b2})$ for $s_b \leftarrow R_q, e_{b1}, e_{b2} \leftarrow D_{R,\sigma}$ is indistinguishable from $(B_{b1}, B_{b2}) \leftarrow R_q^2$ for $b = 1, 2, \cdots, 2\log n$. This concludes our proof. □

Next, we consider the compatible function families $\mathcal{F}_0$ and $\mathcal{F}_1$ for Σ_1.

Lemma A4. *Assume that $\ell = \log n$. Let $\mathcal{F}_0$ and $\mathcal{F}_1$ be the two families of function distributions with respect to $a_1|a_2|\{u_i|\lambda_i\}_{i=1}^t$ in Σ_1 defined as follows:*

$$\mathcal{F}_0 = \{f \mid f(o_1, o_2, o_3) = \lfloor(\mathbf{s}(a_1, a_2, -\bar{u}) + \mathbf{e})(o_1, o_2, o_3)^T + \mathbf{r}\rceil_\theta, \mathbf{s} \leftarrow R_q^{3\ell\times 1}, \mathbf{e} \leftarrow D_{R,\sigma}^{3\ell\times 3}, \mathbf{r} \leftarrow R_q^{3\ell}\}$$
$$\mathcal{F}_1 = \{f \mid f(o_1, o_2, o_3) = \lfloor\mathbf{B}(o_1, o_2, o_3)^T + \mathbf{r}\rceil_\theta, \mathbf{B} \leftarrow R_q^{3\ell\times 3}, \mathbf{r} \leftarrow R_q^{3\ell}\},$$

where $12\sigma n \eta_t'^{1.5} \log n \leq \theta \leq \frac{q}{n \log n}$, and $\eta_t' = 2\eta_t$. Then, $\mathcal{F}_0$ and $\mathcal{F}_1$ are $(\frac{2^9}{3^9}, 1)$-compatible with respect to Σ_1.

Proof. The proof is very similar to Lemma A3. We only sketch the main changes: (1) we use $(a_1, a_2, -\bar{u})(o_1, o_2, o_3)^T = 0$ (fixed) instead of $(a_1, a_2)(z_1, z_2)^T = \sum_i v_i + uc$ (fixed), and hence $\mathcal{F}_0'$ consists only of a constant function $\mathbf{r}$; (2) η_t is replaced by η_t'. Furthermore, the injective property of $\mathbf{B}(o_1, o_2, o_3) + \mathbf{r}$ is reduced to the invertibility of $B = (a_{ij})_{i,j=1,2,3}$ (instead of an order 2 matrix) when a_{ij} is random in R_q. By Gaussian elimination, if a_{11} is invertible, then we make the entries $(1, 2)$ and $(1, 3)$ in B as zero. This updates a_{22} to a_{22}' and a_{33} to a_{33}', while a_{22}' and a_{33}' are still uniformly random in R_q. If a_{22}' is invertible, then we can make an a_{23}' zero similarly that updates a_{33}' to a_{33}'' while preserving its uniformity. So, B is invertible if a_{11}, a_{22}' and a_{33}'' are all invertible, which has a probability of at least $1 - 3 * 2q^{-n/2}$—similar to the argument in Lemma A3. So, for $\mathbf{B} = (\mathbf{B}_i)_{i=1}^{\ell}$, $\mathbf{B}(o_1, o_2, o_3) + \mathbf{r}$ is invertible if some $\mathbf{B}_i$ is invertible. This is violated with the negligible probability only. □

From Lemmas A2, A3, and A4, we can immediately conclude the following corollary.

Corollary A1. *Σ_2 is $\frac{2^6}{3^6}$-weakly collapsing and Σ_1 is $\frac{2^9}{3^9}$-weakly collapsing.*

Proof of Theorem 6. From Corollary A1, we know that Σ_1 and Σ_2 are both weakly collapsing. Then, Lemma A1 gives our desired result. □

Appendix B. Encoding of *CStO* or *CStO$_s$* and Efficient Operations on Oracle State

In this section, we detail how to efficiently encode $CStO$ (or $CStO_s$) and efficiently implement operations (such as U_R and projective measurements) on the oracle register. Since $CStO$ is a special case of $CStO_s$, we only need to consider $CStO_s$. Let q be a polynomial upper bound on the number of random oracle queries to $CStO_s$. Let $\mathcal{X} = \{x_1, \cdots, x_N\}$ be an ordered set, with $x_1 < \cdots < x_N$ and $|\mathcal{X}| = N$ and $0 \notin \mathcal{X}$. Let $\mathcal{D}_q$ be the set of $\mathbf{y} \in \bar{\mathcal{Y}}^{\mathcal{X}}$ that contains at most q non-$\perp$ entries, where $\bar{\mathcal{Y}} = \mathcal{Y} \cup \{\perp\}$. For $\mathbf{y} \in \mathcal{D}_q$, $|\mathbf{y}\rangle_D$ represents $|y_1\rangle_{D_{x_1}} \cdots |y\rangle_{D_{x_N}}$. We can encode it as $|x_1'\rangle|y_1'\rangle \cdots |x_\ell'\rangle|y_\ell'\rangle(|0\rangle|\perp\rangle)^{q-\ell}$ (denoted by $|(\mathbf{x}', \mathbf{y}')\rangle$, and—in this case—the number of records in the encoded D is denoted as $|D| := \ell$), where $x'_1 < x'_2 < \cdots < x'_\ell$ are all the indices in $\mathbf{y}$ with $D(x'_i) = y'_i \neq \perp$. Denote this encoding by *enc*. Let $\mathcal{L}_q \subset \mathcal{X} \times \mathcal{Y}$ be the set of all the possible pairs $(\mathbf{x}', \mathbf{y}')$ of cardinality at most q (sorted according to the first coordinate). Since $|(\mathbf{x}', \mathbf{y}')\rangle$ represents $|x'_1\rangle|y'_1\rangle \cdots |x'_\ell\rangle|y'_\ell\rangle(|0\rangle|\perp\rangle)^{q-\ell}$ for $(\mathbf{x}', \mathbf{y}') = \{(x'_i, y'_i)\}_{i=1}^{\ell}$, with $x'_1 < x'_2 < \cdots < x'_\ell$ and $\ell \leq q$, *enc* is a unitary between $\mathcal{H}(\mathcal{D}_q)$ and $\mathcal{H}(\mathcal{L}_q)$ (because *enc* is one–one and maps between the two sets of orthonormal basis states).

With *enc* in mind, we claim that our results in this paper hold when the quantum state in D is encoded (via *enc*). Specifically, if originally an operator O is applied (with the state on D not encoded), it now applies $enc \cdot O \cdot enc^\dagger$ (with the state on D encoded), where *enc* operates on D. Since $enc^\dagger \cdot enc = I$, the final (adversary–oracle) state with or without encoding on D is related by an *enc* unitary. This will not change the final *adversary* output (from, say, measurement say $M = \{M_t\}_t$), as $\langle\psi| \cdot enc^\dagger \cdot M_t^\dagger M_t \cdot enc|\psi\rangle = \langle\psi|M_t^\dagger M_t|\psi\rangle$ (recall that an adversary does not operate on D, and so *enc* and M_t operate on disjoint registers and commute; as well *enc* is unitary).

However, this is not enough, as we need an efficient implementation of *enc*. Our next step is to deal with this. We first introduce some notations. If D has a state $|(\mathbf{x}, \mathbf{y})\rangle$ with $|D| = \ell < q$, define $|(\mathbf{x}, \mathbf{y}) \cup (x, y)\rangle$ with $x \neq x_i$ for any $i = 1, \cdots \ell$, as sorted pairs $|(\mathbf{x}', \mathbf{y}')\rangle$ (with respect to the first coordinate), which is updated from $(\mathbf{x}, \mathbf{y})$ with (x, y) inserted. This operation is undefined for $\ell \geq q$. Similarly, we can define $|(\mathbf{x}, \mathbf{y}) \backslash (x_i, y_i)\rangle$ by removing (x_i, y_i) from D and sorting the remaining pairs. Next, we introduce the encoding operator COD on XD. For $x \in \mathcal{X}$, COD_x is a unitary from $\mathcal{H}(\bar{\mathcal{L}}_q)$ to $\mathcal{H}(\bar{\mathcal{L}}_q)$, where $\bar{\mathcal{L}}_q \subset \mathcal{X} \times \bar{\mathcal{Y}}$ is similar to $\mathcal{L}_q$, except that $(\mathbf{x}, \mathbf{y}) \in \bar{\mathcal{L}}_q$ means $y_i \in \bar{\mathcal{Y}}$ (instead of $y \in \mathcal{Y}$). For basis state $|(\mathbf{x}, \mathbf{y})\rangle_D$ with $(\mathbf{x}, \mathbf{y}) \in \bar{\mathcal{L}}_q$ and $|D| = \ell$, we use $D(x_i)$ to denote y_i and $D(x) = nil$ if $x \neq x_i$ for any $i = 1, \cdots, \ell$. Essentially, COD_x operates on D_x (by trying to clean up or adding entry $(x, \perp)$) and then sorts the updated $|(\mathbf{x}, \mathbf{y})\rangle$ on D. Specifically, it operates as follows:

- If $D(x) \in \mathcal{Y}$, then $\mathsf{COD}_x|(\mathbf{x}, \mathbf{y})\rangle_D = |(\mathbf{x}, \mathbf{y})\rangle$.
- If $D(x) = \perp$, then $\mathsf{COD}_x|(\mathbf{x}, \mathbf{y})\rangle_D = |(\mathbf{x}, \mathbf{y}) \backslash (x, \perp)\rangle$ (this implies $|D| < q$ after the operation).
- If $D(x) = nil$ (i.e., x is not in D) and $|D| < q$, then $\mathsf{COD}_x|(\mathbf{x}, \mathbf{y})\rangle_D = |(\mathbf{x}, \mathbf{y}) \cup (x, \perp)\rangle$.
- If $D(x) = nil$ and $|D| = q$, then $\mathsf{COD}_x|(\mathbf{x}, \mathbf{y})\rangle_D = |(\mathbf{x}, \mathbf{y})\rangle$.

Note that COD_x is unitary, as it maps from an orthonormal basis to an orthonormal basis in $\mathcal{H}(\bar{\mathcal{L}}_q)$. Furthermore, COD_x is obviously Hermitian. Finally, we define $\mathsf{COD} = \sum_{x \in \mathcal{X}} |x\rangle\langle x|_X \otimes \mathsf{COD}_x$. Note that this COD can be implemented in a polynomial size of quantum gates, as it can be described in a polynomial, and hence, the known techniques (e.g., [37]) can be applied.

We know that without encoding, the initial state of D is $\otimes_x|\perp\rangle_{D_x}$; hence, after encoding, the initial state is $(|0\rangle|\perp\rangle)^q$. In the following, we show that $enc \cdot O \cdot enc^\dagger$ for any original operator O in this paper can be implemented in polynomial time. This can be seen through the following cases:

1. O does not operate on D. For example, attacker's operator and projective measurements on P belong to this category. In this case, this is due to the fact that *enc* and O operate on disjoint registers, $enc \cdot enc^\dagger = I$, and $enc \cdot O \cdot enc^\dagger = O$. So, instead of $enc \cdot O \cdot enc^\dagger$, it suffices to apply O.
2. $CStO_{sXYD}$. Recall that $CStO_{sXYD} = \sum_{x \in \mathcal{X}} |x\rangle\langle x| \otimes CStO_{sYD_x}$ and $CStO_{sYD_x} = F_{D_x} \cdot \mathsf{CNOT}_{YD_x} \cdot F_{D_x}$ for $x \notin \Xi_1$ and $CStO_{sYD_x} = \mathsf{CNOT}_{YD_x}$ for $x \in \Xi_1$. We implement $enc \cdot CStO_s \cdot enc^\dagger$ with $\mathsf{COD} \cdot CStO_s \cdot \mathsf{COD} = \sum_{x \in \mathcal{X}} |x\rangle\langle x| \otimes \mathsf{COD}_x \cdot CStO_{sYD_x} \cdot \mathsf{COD}_x$. The validity of this implementation can be verified through the basis state $|(\mathbf{x}, \mathbf{y})\rangle$. The verification is tedious but straightforward and hence omitted here.
3. U_R. Recall that for $\mathbf{y} \in \mathcal{D}_q$, there exists $x'_1 < x'_2 < \cdots < x'_\ell$ so that $y_{x'_i} \in \mathcal{Y}$ and $y_x = \perp$ for $x \neq x'_i$ for any $i \in [\ell]$. Then, $\mathbf{y}$ is encoded as $(\mathbf{x}', \mathbf{y}')$, where $\mathbf{y}' = (y_{x'_1}, \cdots, y_{x'_\ell})$. Define $\tilde{f}_R((x'_1, y'_1), \cdots, (x'_q, y'_q)) = \sum_i x'_i \cdot \bar{R}(x'_1, y'_1) \cdots \bar{R}(x'_{i-1}, y'_{i-1}) \cdot R(x'_i, y'_i)$, where $x'_i = 0$ and $y'_i = \perp$ for $i > \ell$. We recall that $f_R(\mathbf{y}) = \tilde{f}_R(\mathbf{x}', \mathbf{y}')$. Define unitary $\tilde{U}_R$ so

that $\tilde{U}_R|(\mathbf{x}',\mathbf{y}')\rangle|0\rangle_P = |(\mathbf{x}',\mathbf{y}')\rangle|\tilde{f}_R(\mathbf{x}',\mathbf{y}')\rangle$. Then, $enc \cdot U_R \cdot enc^\dagger$ can be implemented by $\tilde{U}_R$, by directly operating $\tilde{U}_R$ on DP without decoding D.

4. *Measurement* $\Pi = (\Pi_0, \Pi_1) = (|\bot\rangle\langle\bot|, I - |\bot\rangle\langle\bot|)$ on D_x (in PointReg1 query). In this case, we implement $enc \cdot \Pi_b \cdot enc^*$ as $\mathsf{COD} \cdot \Pi_b \cdot \mathsf{COD}$. For any $(\mathbf{x}',\mathbf{y}') \in \mathcal{L}_q$, let $enc^*|(\mathbf{x}',\mathbf{y}')\rangle = |\mathbf{y}\rangle$. It suffices to verify $\mathsf{COD}_x \cdot \Pi_b \cdot \mathsf{COD}_x|(\mathbf{x}',\mathbf{y}'\rangle = enc \cdot \Pi_b|\mathbf{y}\rangle$. This can be checked for cases $D(x) = nil, \bot, y$ for $y \in \mathcal{Y}$. Tedious details are omitted.
5. Measurement on D. In this paper, the measurement property on D with $|y\rangle$ only depends on the non-$\bot$ entries. That is, the property $f(\mathbf{y})$ equals to $\tilde{f}((\mathbf{x}',\mathbf{y}'))$ for some $\tilde{f}$, where $enc(\mathbf{y}) = (\mathbf{x}',\mathbf{y}')$. Hence, measurement on the uncompressed D for property f can be done on compressed D for property $\tilde{f}$. For example, f is a collision property on $\mathbf{y}$ for non-$\bot$ and is equivalent to the collision property $\tilde{f}$ on encoded $\mathbf{y}$ (i.e., $(\mathbf{x}',\mathbf{y}')$). Since $\tilde{f}$ on the encoded D can be implemented efficiently, measurement of the property f can be done efficiently.

Based on the analysis above, we can conclude that our computation with the oracle state un-encoded can be implemented by applying efficient operations with oracle state encoded, preserving the same adversary success probability and the resulting joint state related only by the unitary encoding on the oracle state.

Appendix C. Proof of Lemma 14

Proof. Our strategy is to relate the collision probabilities before and after *one* oracle query when the abort event does not happen. Since there are at most q queries of either $PointReg1$ or $CStO_s$ to $\mathbf{CStO}_s$, and the initial state $\otimes_x|\bot\rangle_{D_x}$ has no collision, this will allow us to bound the collision probability in the final state. We use μ to represent the collision probability after the next operation and μ' to the collision probability before the query. We will show $\sqrt{\mu} \leq \sqrt{\mu'} + \epsilon$ for some ϵ. We assume that the current state is a pure state $|\psi\rangle = \sum_{xyz\mathbf{y}} \lambda_{xyz\mathbf{y}}|x\rangle|\phi_y\rangle|z\rangle|\mathbf{y}\rangle_D$ (the mixed state will be handled later), where we use basis $\{\phi_y\}_y$ on response register Y for the ease of adapting the phase-oracle-based proof in [31] to $\mathbf{CStO}_s$. If the next query is $PointReg0$, then the state is unchanged and hence $\mu' = \mu$. Then, we consider the other two cases: a random oracle query and a PointReg1 query.

The next operation is a random oracle query. We classify basis $\{|x, \phi_y, z, \mathbf{y}\rangle\}_{xyz\mathbf{y}}$ into four sets: P, Q, R, S:

- P: It consists of the basis states so that $\mathbf{y}$ contains a collision.
- Q: It consists of the basis states satisfying (1) $\mathbf{y}$ has no collision; (2) $y \neq 0$; (3) $y_x = \bot$.
- R: It consists of the basis states satisfying (1) $\mathbf{y}$ has no collision; (2) $y \neq 0$; (3) $y_x \neq \bot$.
- S: It consists of the basis states satisfying (1) $\mathbf{y}$ has no collision; (2) $y = 0$.

We also use P, Q, R, S to denote the projection into the space spanned by the basis states in the respective category. Then, $P + Q + R + S = I$. Since the attacker only makes at most q random oracle queries, D contains at most q non-$\bot$ entries. In this case, the square root of collision probability (when abort does not occur) is $||P \cdot CStO_s \cdot \Lambda_{i0}|\psi\rangle||$, which is at most

$$||P \cdot CStO_s \cdot \Lambda_{i0}P|\psi\rangle|| + ||P \cdot CStO_s \cdot \Lambda_{i0}Q|\psi\rangle|| + ||P \cdot CStO_s \cdot \Lambda_{i0}R|\psi\rangle|| + ||P \cdot CStO_s \cdot \Lambda_{i0}S|\psi\rangle||.$$

Notice that $CStO_s$ has two cases: if $x \in \Xi_1$, then $CStO_{sYD_x} = \mathrm{CNOT}_{YD_x}$; if $x \notin \Xi_1$, then $CStO_{sYD_x} = CStO_{YD_x}$. Let us write $|\psi\rangle = \sum_x |\psi_x\rangle$, where $\psi_x = |x\rangle_X \cdots$.

We first consider the case $x \notin \Xi_1$. In this case, $CStO_s|\psi_x\rangle = CStO|\psi_x\rangle$.

Case $P|\psi_x\rangle$. In this case, $||P \cdot CStO \cdot \Lambda_{i0}P|\psi_x\rangle|| \leq ||CStO \cdot \Lambda_{i0}P|\psi_x\rangle|| = ||\Lambda_{i0} \cdot P|\psi_x\rangle|| \leq ||P|\psi_x\rangle||$.

Case $Q|\psi_x\rangle$. $CStO$ on $|x,z\rangle|\phi_y\rangle \otimes |\mathbf{y}\rangle_D$ (in Q) gives $|x,z\rangle|\phi_y\rangle \otimes \frac{1}{\sqrt{2^n}} \sum_w (-1)^{y \cdot w}|\mathbf{y} \cup (w)_x\rangle$ as $y_x = \bot$. Hence, after operator P, it has a norm of at most $\sqrt{q\Gamma_f/2^n}$, as $|D| \leq q$ and the collision imply that $f(x,w) = f(x', y_{x'})$ for some $x' \neq x$ (recall that $\mathbf{y}$ has no collision), because each $(x', y_{x'})$ collides with (x,w) for at most Γ_f possible ws. Since a distinct

$|x,z\rangle|\phi_y\rangle \otimes |\mathbf{y}\rangle$ (in Q) gives orthogonal images, it follows that $P \cdot CStO \cdot \Lambda_{i0}Q|\psi_x\rangle$ has a norm of at most $\sqrt{q\Gamma_f/2^n}||\Lambda_{i0}Q|\psi_x\rangle|| \leq \sqrt{q\Gamma_f/2^n}||Q|\psi_x\rangle||$ (as Λ_{i0}, Q are projectors on D in the computational basis).

Case $R|\psi_x\rangle$. For category R, consider that D has a state $|\mathbf{y} \cup (w)_x\rangle$ with $y_x = \perp$ and $w \neq \perp$. By a tedious calculation (also in [31, Theorem 1]), we can show that $CStO|x,z\rangle|\phi_y\rangle|\mathbf{y} \cup (w)_x\rangle$ is

$$|x,z\rangle|\phi_y\rangle\left((-1)^{y\cdot w}\left(|\mathbf{y} \cup (w)_x\rangle + \frac{1}{2^{n/2}}|\mathbf{y}\rangle\right) + \frac{1}{2^n}\sum_{y'}(1-(-1)^{y\cdot w}-(-1)^{y\cdot y'})|\mathbf{y} \cup (y')_x\rangle\right).$$

After applying P, since $|x,\phi_y,z\rangle|\mathbf{y} \cup (w)_x\rangle$ is in R, and so $|x,\phi_y,z\rangle|\mathbf{y}\rangle$ is in Q, it becomes

$$|x,z\rangle|\phi_y\rangle \otimes \frac{1}{2^n}\sum_{y':\ \exists x', f(x,y')=f(x',y_{x'})}(1-(-1)^{y\cdot w}-(-1)^{y\cdot y'})P|\mathbf{y} \cup (y')_x\rangle. \tag{A9}$$

Now, we relate the different states of form $|x,z\rangle|\phi_y\rangle|\mathbf{y} \cup (w)_x\rangle$ in category R . If they have different $(x,z,y,\mathbf{y})$ tuples, then their results in (A9) are orthogonal (as they all have $y_x = \perp$ by definition, and thus their tuples $(x,z,y,\{y_t\}_{t\neq x})$ are different). So, we only need to consider the setting of the same $(x,z,y,\mathbf{y})$ for the norm in this category. In this case, there are at most 2^n choices of w. By the Chauchy–Schwardz inequality, the norm of the superposition of Equation (A9) over w is at most $\sqrt{2^n}$ times its maximum over w. We are left to upper bound the norm of Equation (A9) for a given w. In this case, notice that for each $(x',y_{x'})$ with $y_{x'}$ non-$\perp$, there are at most Γ_f possible y' in Equation (A9) so that $f(x,y') = f(x',y_{x'})$. There are at most q non-$\perp$ $y_{x'}$ in $\mathbf{y}$. Equation (A9) has a norm of at most $3\sqrt{q\Gamma_f} \cdot 2^{-n}$. Hence, the superposition of Equation (A9) has a norm of at most $3\sqrt{q\Gamma_f/2^n}$. Thus,

$$||P \cdot CStO \cdot \Lambda_{i0}R|\psi_x\rangle|| \leq 3\sqrt{q\Gamma_f/2^n}||\Lambda_{i0}R|\psi_x\rangle|| \leq 3\sqrt{q\Gamma_f/2^n}||R|\psi_x\rangle||$$

(as Λ_{i0}, R are projectors on D in the computational basis).

Case $S|\psi_x\rangle$. In this case, $CStO \cdot |x,z\rangle|\phi_0\rangle|\mathbf{y}\rangle = |x,z\rangle|\phi_0\rangle|\mathbf{y}\rangle$, which has no collision.

Summarizing the four cases, we have

$$||P \cdot CStO \cdot \Lambda_{i0}|\psi_x\rangle|| \leq ||P \cdot |\psi_x\rangle|| + 4\sqrt{q\Gamma_f/2^n}\ |||\psi_x\rangle||. \tag{A10}$$

Second, we consider case $x \in \Xi_1$ and so $CStO_s = \text{CNOT}$. In this case, notice that $P \cdot \text{CNOT} \cdot \Lambda_{i0}|\psi_x\rangle = P^2 \cdot \text{CNOT} \cdot \Lambda_{i0}|\psi_x\rangle = P \cdot \text{CNOT} \cdot \Lambda_{i0}P|\psi_x\rangle$, as P commutes with CNOT and Λ_{i0}. Furthermore, $||P \cdot \text{CNOT} \cdot \Lambda_{i0}P|\psi_x\rangle|| \leq ||\text{CNOT} \cdot \Lambda_{i0}P|\psi_x\rangle|| = ||\Lambda_{i0}P|\psi_x\rangle|| \leq ||P|\psi_x\rangle||$, as CNOT is unitary, and Λ_{i0} is a projector in the computational basis (as is the case for P).

Summarizing both $x \in \Xi_1$ and $x \notin \Xi_1$ cases and noticing that their images are orthogonal (as $|x\rangle_X$ will remain unchanged after the operation), we have

$$||P \cdot CStO_s \cdot \Lambda_{i0}|\psi\rangle|| \leq ||P \cdot |\psi\rangle|| + 4\sqrt{q\Gamma_f/2^n} \tag{A11}$$

For the mixed state, suppose that $|\psi\rangle$ has the probability λ_ψ. Then, averaging the square of the above inequality, expanding the right side, and using the Cauchy–Schwarz inequality $\sum_i \lambda_i x_i \leq (\sum_i \lambda_i x_i^2)^{1/2}$ with $\lambda_i, x_i \geq 0$ and $\sum_i \lambda_i = 1$, we have

$$\sqrt{\mu} \leq \sqrt{\mu'} + 4\sqrt{q\Gamma_f/2^n}. \tag{A12}$$

The next operation is PointReg1. Still, we assume that the current adversary–oracle joint state is a pure state $|\psi\rangle$. In this case, under event $\neg$abort, projection Π_0 on $|\psi\rangle$ is applied, and $|\perp\rangle_{D_x}$ is replaced by $|r\rangle_{D_x}$. Since r is random, the resulting state ρ_0 is the mixed state (over r), and so the collision probability is $tr(P \cdot \rho_0 \cdot P)$. We write the current state $|\psi\rangle = \sum_{yz\mathbf{y}} \alpha_{yz\mathbf{y}} |x,z\rangle|\phi_y\rangle|\mathbf{y}\rangle_D$. We classify the basis states $|x,z,\phi_y\rangle|\mathbf{y}\rangle_D$ into three categories P, Q', R', similar to the $CStO_s$ case. But, different from Q, R, here Q', R' respectively removes condition 2 (the restriction on y). It is not hard to show. Let $\rho_0 = \sum_i^n M_i^\dagger |\psi\rangle\langle\psi| M_i$. Let $|a_i\rangle = PM_iP|\psi\rangle, |b_i\rangle = PM_iQ'|\psi\rangle, |c_i\rangle = PM_iR'|\psi\rangle$. Then, Equation (A13) becomes $\sqrt{\sum_{i=1}^n || |a_i\rangle + |b_i\rangle + |c_i\rangle ||^2} \le \sqrt{\sum_{i=1}^n || |a_i\rangle ||^2} + \sqrt{\sum_{i=1}^n || |b_i\rangle ||^2} + \sqrt{\sum_{i=1}^n || |c_i\rangle ||^2}$. Furthermore, define $\mathbf{a}$ as the long vector $(|a_1\rangle, \cdots, |a_n\rangle)$ and $\mathbf{b}$, $\mathbf{c}$ similarly. Then, Equation (A13) becomes $||\mathbf{a}+\mathbf{b}+\mathbf{c}|| \le ||\mathbf{a}|| + ||\mathbf{b}|| + ||\mathbf{c}||$, which is evident. This means that $\sqrt{tr(P \cdot \rho_0 \cdot P)}$ for any mixed state ρ_0 that starts from $|\psi\rangle$ and, through some quantum algorithm, can be upper bounded by

$$\sum_{V \in \{P,Q',R'\}} \sqrt{tr(P \cdot \rho_{0V} \cdot P)}, \tag{A13}$$

where ρ_{0V} is the mixed state ρ_0 with the input state $V|\psi\rangle$ (instead of $|\psi\rangle$).

Case $P|\psi\rangle$. In this case, after applying Π_0, only the basis states $|x,z\rangle|\phi_y\rangle|\mathbf{y}\rangle$ in $P|\psi\rangle$, with $y_x = \perp$ and $\mathbf{y}$ containing a collision, are left and after the query; this state becomes $|x,z\rangle|\phi_y\rangle|\mathbf{y} \cup (r)_x\rangle$ for a uniformly random r. Note that $\mathbf{y} \cup (r)_x$ for any r still contains a collision. Therefore, $tr(P \cdot \rho_{0P} \cdot P) = \sum_r 2^{-n} \langle\psi| P\Pi_0 U_{\perp,r} PP U_{\perp,r} \Pi_0 P |\psi\rangle = \langle\psi| P\Pi_0\Pi_0 P|\psi\rangle = ||\Pi_0 P|\psi\rangle||^2 \le ||P|\psi\rangle||^2$, where $U_{\perp,r} = |r\rangle\langle\perp| + |\perp\rangle\langle r| + \sum_{s \neq r} |s\rangle\langle s|$. Thus, the collision probability of $P|\psi\rangle$ after the query is at most $||P|\psi\rangle||^2$.

Case $Q'|\psi\rangle$. In this case, since D_x in this category always has $\perp$, $\Pi_0 Q'|\psi\rangle = Q'|\psi\rangle$, which, after applying $U_{\perp,r}$ and P, changes the basis state $|x,z\rangle|\phi_y\rangle|\mathbf{y}\rangle$ in $Q'|\psi\rangle$ (where $y_x = \perp$) to $|x,z\rangle|\phi_y\rangle|\mathbf{y} \cup (r)_x\rangle$ (if (x,r) collides with $(x', y_{x'})$ (for some $x' \neq x$)) or 0 (if (x,r) does not collide with any $(x', y_{x'})$). Notice that for different $(x,z,y,\mathbf{y})$, $|x,z\rangle|\phi_y\rangle|\mathbf{y} \cup (r)_x\rangle$ in this category will be orthogonal to each other. Therefore,

$$tr(P \cdot \rho_{0Q'} \cdot P) \le \frac{q\Gamma_f}{2^n} ||Q'|\psi\rangle||^2, \tag{A14}$$

as there are at most q choices of $(x', y_{x'})$ in $\mathbf{y}$ and that $\mathbf{y}$ itself has no collision by definition.

Case $R'|\psi\rangle$. In this case, since $D(x) \neq \perp$, under $\neg$abort event, $\Pi_0 R'|\psi\rangle = 0$ (no collision).

Summarizing the three cases, we have that

$$\sqrt{tr(P \cdot \rho_0 \cdot P)} \le ||P|\psi\rangle|| + \sqrt{\frac{q\Gamma_f}{2^n}} ||\psi||. \tag{A15}$$

If the current state is a mixed state so $|\psi\rangle$ has a probability λ_ψ and ρ_ψ is $P \cdot \rho_0 \cdot P$ from $|\psi\rangle$, then $\sqrt{\sum_\psi \lambda_\psi tr(\rho_\psi)} \le \sqrt{\sum_\psi \lambda_\psi (||P|\psi\rangle|| + \sqrt{\frac{q\Gamma_f}{2^n}} || |\psi\rangle ||)^2}$, which is upper bounded by

$$\sqrt{\sum_\psi \lambda_\psi ||P|\psi\rangle||^2} + \sqrt{\sum_\psi \lambda_\psi \sqrt{\frac{q\Gamma_f}{2^n}} || |\psi\rangle ||^2} = \sqrt{\mu'} + \sqrt{q\Gamma_f/2^n}, \tag{A16}$$

where the first part of Equation (A16) uses $\sqrt{\sum_{i=1}^n (\mathbf{a}_i + \mathbf{b}_i)^2} \le \sqrt{\sum_{i=1}^n ||\mathbf{a}_i||^2} + \sqrt{\sum_{i=1}^n ||\mathbf{b}_i||^2}$. This gives $\sqrt{\mu} \le \sqrt{\mu'} + \sqrt{\frac{q\Gamma_f}{2^n}}$.

Let μ_q be the collision probability of the final state. Since there are at most q queries (either PointReg1 or random oracle query) to $\mathbf{CStO}_s$, $\sqrt{\mu_q} \le 4q\sqrt{\frac{q\Gamma_f}{2^n}}$. This gives our lemma. □

Appendix D. Proof of Theorem 4

For constant $c > 0$, define $\lambda_{i^c,j^c,k^c,\underline{x}^c,w,\mathbf{y}}$ to be the probability that the measurement in the i_tth oracle query in $\mathsf{Exp}_{i^c,j^c,k^c}$ has outcome $\underline{x}_t$ (for $t = 1, \cdots, c$), and the final measurement outcome is $(w, \mathbf{y})$, where $\underline{x}^c = (\underline{x}_1, \cdots, \underline{x}_c)$. For $v \in \mathcal{Y}$, we use $\{v\}_x$ to denote the vector in $\mathcal{Y}^{\mathcal{X}}$ so that the coordinate at index x is v and the remaining coordinates are all 0 (do not confuse this with $(v)_x$, where it is v at coordinate x and $\perp$ otherwise). For $\mathbf{v} \in \mathcal{Y}^{\mathcal{X}}$, we use $|\phi_{\mathbf{v}}\rangle_D$ to denote the oracle state with $|\phi_{v_x}\rangle_{D_x}$. Then, the *CStO* oracle has the following property (which is an alternative description of Fourier oracle's essential property in [31] but in the language of *CStO*).

Fact 1. $|x\rangle_X|\phi_y\rangle_Y F_D|\phi_{\mathbf{v}}\rangle_D$ *under the CStO oracle will be mapped to* $|x\rangle_X|\phi_y\rangle_Y F_D|\phi_{\mathbf{v}+\{y\}_x}\rangle_D$.

The following lemma is extended from (Theorem 9, [33]) through translating its proof on a compressed Fourier oracle using the *CStO* oracle and generalizing it from Exp_{ijk} to $\mathsf{Exp}_{i^c,j^c,k^c}$.

Lemma A5. *For any* $w, \mathbf{y}, x^c$ *with* $D(\underline{x}_t) \neq \perp$ $(t = 1, \cdots, c)$ *and* $\gamma_{w,\mathbf{y}}$ *is the probability in the normal game with output* $(w, \mathbf{y})$. *Then, there exists* (i^c, j^c, k^c) *so that* $\lambda_{i^c,j^c,k^c,\underline{x}^c,w,\mathbf{y}} \geq \gamma_{w,\mathbf{y}}/(q + \binom{q}{3})^{2c}$.

Proof. Let $\sum_{x,y,z} \alpha_{x,y,z}|x, \phi_y, z\rangle$ be the state of the adversary before the first query. Let $U^{(i)}_{x,y,z,x',y',z'}$ be the transition function from $|x, \phi_y, z\rangle$ to $|x', \phi_{y'}, z'\rangle$, starting from the ith query to *CStO* but right before $(i+1)$th query, where the *CStO* is represented under basis $F_D|\phi_{\mathbf{v}}\rangle_D$. By Fact 1 above, this is well defined for a fixed adversary quantum algorithm (as an adversarial algorithm is not acting on D). For any vector $\mathbf{x}, \mathbf{y}, \mathbf{z}$ and w, let

$$\alpha_{\mathbf{x},\mathbf{y},\mathbf{z},w} = \alpha_{x_1,y_1,z_1} U^{(1)}_{x_1,y_1,z_1,x_2,y_2,z_2} \cdots U^{(q)}_{x_q,y_q,z_q,w}. \tag{A17}$$

Then, we can write the final adversary–oracle joint state as

$$\sum_{\mathbf{x},\mathbf{y},\mathbf{z},w} \alpha_{\mathbf{x},\mathbf{y},\mathbf{z},w}|w\rangle \otimes F_D|\phi_{\{y_1\}_{x_1}+\cdots+\{y_q\}_{x_q}}\rangle_D. \tag{A18}$$

(Note: Here, the oracle uses basis $F_D|\phi_{\mathbf{y}}\rangle$ and will switch to $|\mathbf{y}\rangle$ later). For any $\mathbf{v} \in \mathcal{Y}^{\mathcal{X}}$ with at most q non-zero coordinates, define set $S_{\mathbf{v}}$: it contains $\mathbf{x}, \mathbf{y}$ so that $\sum_{i=1}^{q}\{y_i\}_{x_i} = \mathbf{v}$, where the addition is the coordinate-wise addition in group $\mathcal{Y}$.

If we measure D using basis $F_D|\phi_{\mathbf{v}}\rangle$ for $\mathbf{v} \in \mathcal{Y}^{\mathcal{X}}$ and measure w normally, then the measurement outcome $(w, \mathbf{v})$ has a probability $\gamma_{w,\mathbf{v}} = |\gamma'_{w,\mathbf{v}}|^2$, where

$$\gamma'_{w,\mathbf{v}} = \sum_{(\mathbf{x},\mathbf{y},\mathbf{z}):(\mathbf{x},\mathbf{y})\in S_{\mathbf{v}}} \alpha_{\mathbf{x},\mathbf{y},\mathbf{z},w}.$$

Next, starting with $S_{\mathbf{v},i^0,j^0,k^0} := S_{\mathbf{v}}$, we iteratively define $S_{\mathbf{v},i^t,j^t,k^t}$ as a subset of $S_{\mathbf{v},i^{t-1},j^{t-1},k^{t-1}}$. For vector $(\mathbf{x}', \mathbf{y}')$ and x, we say that $\underline{x}$ **is in the database** after the tth query, wherein we mean that $F_D|\phi_{\{y'_1\}_{x'_1}+\cdots+\{y'_t\}_{x'_t}}\rangle$ is orthogonal to $|\perp\rangle_{D_u}$ at *some* coordinate $u \in \underline{x}$ (i.e., at coordinate u, it is $|\phi_y\rangle_{D_u}$ for some $y \neq 0$). We fix x^c with $\mathbf{v}(\underline{x}_t) \neq 0, \forall t \in [c]$. Then, $S_{\mathbf{v},i^t,j^t,k^t}$ is defined as follows:

- **Case** $i_t = j_t = k_t$: It contains all $(\mathbf{x}', \mathbf{y}')$ in $S_{\mathbf{v},i^{t-1},j^{t-1},k^{t-1}}$ so that we have the following:
  1. $\underline{x}_t$ is not in $F_D|\phi_{\{y'_1\}_{x'_1}+\cdots+\{y'_{i_t-1}\}_{x'_{i_t-1}}}\rangle$ (i.e., every index $u \in \underline{x}_t$ has coordinate $|\perp\rangle$).
  2. $\underline{x}_t = \underline{x}'_{i_t}$ and $y'_{i_t} \neq 0$.

- **Case** $i_t < j_t < k_t$: It contains all $(\mathbf{x}', \mathbf{y}')$ in $S_{\mathbf{v}, i^{t-1}, j^{t-1}, k^{t-1}}$ so that we have the following:
  1. $\underline{x}_t$ is not in the database before the i_tth query.
  2. $\underline{x}_t$ is in the database after the i_tth query and befire j_tth query.
  3. $\underline{x}_t$ is not in the database after the j_tth query and befire k_tth query.
  4. $\underline{x}_t$ is in the database after the k_tth query.

Then, we define

$$\gamma'_{i^t, j^t, k^t, w, \mathbf{v}} = \sum_{(\mathbf{x}, \mathbf{y}, \mathbf{z}) : (\mathbf{x}, \mathbf{y}) \in S_{\mathbf{v}, i^t, j^t, k^t}} \alpha_{\mathbf{x}, \mathbf{y}, \mathbf{z}, w}, \tag{A19}$$

where we recall that x^c is fixed and implicit in the γ' and S variables. Then, we have the following claim.

Claim. *For any $x^c, w, \mathbf{v}$ with $\mathbf{v}(\underline{x}_t) \neq 0$ ($t = 1, \cdots, c$), it holds that*

$$\sum_{i_t : i_t = j_t = k_t} \gamma'_{i^t, j^t, k^t, w, \mathbf{v}} - \sum_{i_t < j_t < k_t} \gamma'_{i^t, j^t, k^t, w, \mathbf{v}} = \gamma'_{i^{t-1}, j^{t-1}, k^{t-1}, w, \mathbf{v}} \tag{A20}$$

Proof. Given $(\mathbf{x}, \mathbf{y}) \in S_{\mathbf{v}, i^{t-1}, j^{t-1}, k^{t-1}}$ and $\mathbf{z}$, consider the first i_t queries in the process toward $\alpha_{\mathbf{x}, \mathbf{y}, \mathbf{z}, w} |w\rangle F_D |\phi_{\mathbf{v}}\rangle_D$. Assume that $\underline{x}_t$ is inserted ℓ times into the database (i.e., the change from not in the database from being in the database). Then, $\ell \geq 1$; otherwise, $\mathbf{v}(x_t) = 0$ (contradiction). On the left side, $\alpha_{\mathbf{x}, \mathbf{y}, \mathbf{z}, w}$ will appear in $\sum_{i_t : i_t = j_t = k_t} \gamma'_{i^t, j^t, k^t, w, \mathbf{v}}$ for ℓ times (by the meaning of insertion: before it, it is not in while it is in after it) while appearing in $\sum_{i_t < j_t < k_t} \gamma'_{i^t, j^t, k^t, w, \mathbf{v}}$ for $\ell - 1$ times (as each $(\mathbf{x}, \mathbf{y})$ in $\alpha_{\mathbf{x}, \mathbf{y}, \mathbf{z}, w}$ in this sum requires at least two insertions). This can be seen from the specification of $S_{\mathbf{v}, i^t, j^t, k^t}$. So, $\alpha_{\mathbf{x}, \mathbf{y}, \mathbf{z}, w}$ on the left side appears exactly once. By the definition of $\gamma'_{i^{t-1}, j^{t-1}, k^{t-1}, w, \mathbf{v}}$, it appears on the right side exactly once. Finally, for every $\alpha_{\mathbf{x}, \mathbf{y}, \mathbf{z}, w}$ on the left or right side, it must have $(\mathbf{x}, \mathbf{y}) \in S_{\mathbf{v}, i^{t-1}, j^{t-1}, k^{t-1}}$, by definition of $\gamma'_{i^u, j^u, k^u, w, \mathbf{v}}$ for $u = t, t-1$. The foregoing argument applies again. The claim follows. □

Back to our lemma proof, Equation (A20) for $t = 1, \cdots, c$ can be combined into one equation with the right side $\gamma'_{w, \mathbf{v}}$, while the left side is a sum of $\gamma'_{i^c, j^c, k^c, w, \mathbf{v}}$ over all $(q + \binom{q}{3})^c$ possible (i^c, j^c, k^c). Notice that $\gamma'_{i^t, j^t, k^t, w, \mathbf{v}}$ over (t, i^t, j^t, k^t) has a dependency in a tree structure. Therefore,

$$\gamma'_{w, \mathbf{v}} = \sum_{(i^c, j^c, k^c)} \pm \gamma'_{i^c, j^c, k^c, w, \mathbf{v}}, \tag{A21}$$

where $\pm$ can only be one of $+$ and $-$, but it is not important to be precise here. Either of the two sides of Equation (A21) is the coefficient of $|w\rangle F_D |\phi_{\mathbf{v}}\rangle$.

Let the superposition before the final measurement be $|\psi\rangle = \sum_{w', \mathbf{v}} \gamma'_{w', \mathbf{v}} |w'\rangle F_D |\phi_{\mathbf{v}}\rangle_D$. Let $\mathbf{v}$ be $v_{x'_i}$ at x'_i for $i = 1, \cdots, L$, while it is 0 at any other index. Thus, by definition of the Walsh–Hadamard transform, $|\psi\rangle$ can be expanded as

$$|\psi\rangle = \frac{1}{|\mathcal{Y}|^{L/2}} \sum_{w', \mathbf{v}} \sum_{u_{x'_1}, \cdots, u_{x'_L}} (-1)^{u_{x'_1} v_{x'_1} + \cdots + u_{x'_L} v_{x'_L}} \gamma'_{w', \mathbf{v}} |w'\rangle |\mathbf{u}\rangle_D, \tag{A22}$$

where $u_{x'_j}$ for $j > L$ is $\perp$. Thus, $|w'\rangle |\mathbf{u}\rangle_D$ in $|\psi\rangle$ has coefficient

$$\gamma''_{w', \mathbf{u}} \overset{def}{=} \frac{1}{|\mathcal{Y}|^{L/2}} \sum_{w', \mathbf{v}:\ v_{x'_j} \neq 0, j \in [L]} (-1)^{u_{x'_1} v_{x'_1} + \cdots + u_{x'_L} v_{x'_L}} \gamma'_{w', \mathbf{v}}. \tag{A23}$$

Let $\gamma''_{i^t,j^t,k^t,w',\mathbf{u}}$ be the coefficient of $|w'\rangle|\mathbf{u}\rangle_D$ in $|\psi\rangle$ from $\mathsf{Exp}_{i^c,j^c,k^c}$. Then,

$$\gamma''_{i^t,j^t,k^t,w',\mathbf{u}}=\frac{1}{|\mathcal{Y}|^{L/2}}\sum_{w',\mathbf{v}:\ v_{x'_j}\neq 0, j\in[L]}(-1)^{u_{x'_1}v_{x'_1}+\cdots+u_{x'_L}v_{x'_L}}\gamma'_{i^t,j^t,k^t,w',\mathbf{v}}. \tag{A24}$$

From Equation (A21), we have

$$\gamma''_{w,\mathbf{u}}=\sum_{(i^c,j^c,k^c)}\pm\gamma''_{i^c,j^c,k^c,w,\mathbf{u}}. \tag{A25}$$

Hence, at least one $|\gamma''_{i^c,j^c,k^c,w,\mathbf{u}}|\geq|\gamma''_{w,\mathbf{u}}|/(q+\binom{q}{3})^c$. Since $\lambda_{i^c,j^c,k^c,\underline{x}^c,w,\mathbf{u}}=|\gamma''_{i^c,j^c,k^c,w,\mathbf{u}}|^2$ and $\lambda_{w,\mathbf{u}}=|\gamma''_{w,\mathbf{u}}|^2$, the lemma follows. □

Proof of Theorem 4. We take the implicit $x^c=x_{w,\mathbf{y},1},\cdots,x_{w,\mathbf{y},c}$. Let $\lambda_{\underline{x}^c,w,\mathbf{y}}$ be $\lambda_{i^c,j^c,k^c,\underline{x}^c,w,\mathbf{y}}$ for a random (i^c,j^c,k^c). There are $(q+\binom{q}{3})^c$ possible (i,j,k) in the support of $\mathcal{U}^c_{IJK}$. Then, by Lemma A5, $\lambda_{\underline{x}^c,w,\mathbf{y}}\geq\lambda_{w,\mathbf{y}}/(q+\binom{q}{3})^{3c}$. Hence,

$$\lambda\geq\sum_{(w,\mathbf{y})\in S}\lambda_{\underline{x}^c,w,\mathbf{y}}\geq\sum_{(w,\mathbf{y})\in S}\frac{\gamma_{w,\mathbf{y}}}{(q+\binom{q}{3})^{3c}}=\frac{\gamma}{(q+\binom{q}{3})^{3c}}, \tag{A26}$$

desired! □

References

1. Itakura, K.; Nakamura, K. A public-key cryptosystem suitable for digital multisignatures. *NEC Res. Dev.* **1983**, *71*, 1–8.
2. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: http://bitcoin.org/bitcoin.pdf (accessed on 24 October 2024).
3. Bellare, M.; Neven, G. Multi-signatures in the plain public-Key model and a general forking lemma. In Proceedings of the 13th ACM Conference on COMPUTER and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006; pp. 390–399
4. Shor, P.W. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134
5. Boneh, D.; Gentry, C.; Lynn, B.; Shacham, H. Aggregate and verifiably encrypted signatures from bilinear maps. In *EUROCRYPT 2003*; Biham, E., Ed.; Volume 2656 of LNCS; Springer: Berlin/Heidelberg, Germany, 2003; pp. 416–432.
6. Alper, H.K.; Burdges, J. Two-round trip schnorr multi-signatures via delinearized witnesses. In *CRYPTO 2021, Part I*; Malkin, T., Peikert, C., Eds.; Virtual Event; Springer: Berlin/Heidelberg, Germany, 2021; Volume 12825, pp. 157–188.
7. Bagherzandi, A.; Cheon, J.H.; Jarecki, S. Multisignatures secure under the discrete logarithm assumption and a generalized forking lemma. In Proceedings of the 15th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 27–31 October 2008; pp. 449–458.
8. Boldyreva, A. Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. In *International Workshop on Public Key Cryptography*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 31–46.
9. Lu, S.; Ostrovsky, R.; Sahai, A.; Shacham, H.; Waters, B. Sequential Aggregate Signatures and Multisignatures Without Random Oracles.In *Advances in Cryptology— EUROCRYPT 2006. EUROCRYPT 2006. Lecture Notes in Computer Science*; Springer:Berlin/Heidelberg, Germany, 2006; pp. 465–485.
10. Ma, C.; Weng, J.; Li, Y.; Deng, R.H. Efficient discrete logarithm based multi-signature scheme in the plain public key model. *Des. Codes Cryptogr.* **2010**, *54*, 121–133. [CrossRef]
11. Maxwell, G.; Poelstra, A.; Seurin, Y.; Wuille, P. Simple schnorr multi-signatures with applications to bitcoin. *Des. Codes Cryptogr.* **2019**, *87*, 2139–2164. [CrossRef]
12. Micali, S.; Ohta, K.; Reyzin, L. Accountable-subgroup multisignatures: Extended abstract. In Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, PA, USA, 5–8 November 2001; pp. 245–254.
13. Nick, J.; Ruffing, T.; Seurin, Y. MuSig2: Simple two-round Schnorr multi-signatures. In *CRYPTO 2021*, Part I, LNCS 12825; Springer: Berlin/Heidelberg, Germany, 2021; pp. 189–221.
14. Syta, E.; Tamas, I.; Visher, D.; Wolinsky, D.I.; Jovanovic, P.; Gasser, L.; Gailly, N.; Khoffi, I.; Ford, B. Keeping authorities "honest or bust" with decentralized witness cosigning. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; IEEE Computer Society Press: Piscataway, NJ, USA, 2016; pp. 526–545.
15. He, Q.; Xin, X.; Yang, Q. Security analysis and improvement of a quantum multi-signature protocol. *Quantum Inf. Process.* **2021**, *20*, 26. [CrossRef]

16. Jiang, D.H.; Hu, Q.Z.; Liang, X.Q.; Xu, G.B. A novel quantum multi-signature protocol based on locally indistinguishable orthogonal product states. *Quantum Inf. Process.* **2019**, *18*, 268. [CrossRef]
17. Boschini, C.; Takahashi, A.; Tibouchi, M. Musig-L: Lattice-based multi-signature with single-round online phase. In *Advances in Cryptology—CRYPTO 2022. CRYPTO 2022. Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2022.
18. Fukumitsu, M.; Hasegawa, S. A tightly-secure lattice-based multisignature. In Proceedings of the Asia CCS '19: ACM Asia Conference on Computer and Communications Security, Auckland, New Zealand, 8 July 2019; pp. 3–11.
19. Kansal, M.; Singh, A.K.; Dutta, R. Efficient Multi-Signature Scheme Using Lattice. *Comput. J.* **2022**, *65*, 2421–2429. [CrossRef]
20. Kansal, M.; Dutta, R. Round Optimal Secure Multisignature Schemes from Lattice with Public Key Aggregation and Signature Compression. In *Progress in Cryptology–AFRICACRYPT 2020. AFRICACRYPT 2020. Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2020; pp. 281–300.
21. Liu, Z.-Y.; Tseng, Y.-F.; Tso, R. Cryptanalysis of a round optimal lattice-based multisignature scheme. *Inf. Process. Lett.* **2020**, *182*, 106364. [CrossRef]
22. Ma, C.; Jiang, M. Practical Lattice-Based Multisignature Schemes for Blockchains. *IEEE Access* **2019**, *7*, 179765–179778. [CrossRef]
23. Jiang, S.; Alhadidi, D.; Khojir, H.F. Key-and-Signature Compact Multi-Signatures for Blockchain: A Compiler with Realizations. *IEEE Trans. Dependable Secur. Comput.* **2024**, 1–18. [CrossRef]
24. Damg, I.; Orlandi, C.; Takahashi, A.; Tibouchi, M. Two-round n-out-of-n and multisignatures and trapdoor commitment from lattices. *J. Cryptol.* **2022**, *35*, 14.
25. El Bansarkhani, R.; Sturm, J. An efficient lattice-based multisignature scheme with applications to bitcoins. In *Cryptology and Network Security. CANS 2016. Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2016; pp. 140–155.
26. Fleischhacker, N.; Simkin, M.; Zhang, Z. Squirrel: Efficient Synchronized Multi-Signatures from Lattices. In Proceedings of the CCS '22: 2022 ACM SIGSAC Conference on Computer and Communications Security, Los Angeles, CA, USA, 7–11 November 2022; pp. 1109–1123.
27. Fukumitsu, M.; Hasegawa, S. A lattice-based provably secure multisignature scheme in quantum random oracle model. In *Provable and Practical Security. ProvSec 2020. Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2020.
28. Kiltz, E.; Lyubashevsky, V.; Schaffner, C. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In *EUROCRYPT 2018*; Nielsen, J.B., Rijmen, V., Eds.; Springer: Berling, Germany, 2018; pp. 552–586.
29. Bellare, M.; Rogaway, P. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In Proceedings of the CCS93: 1st ACM Conference on Communications and Computing Security, Fairfax, VA, USA, 3–5 November 1993; pp. 62–73.
30. Canetti, R.; Goldreich, O.; Halevi, S. The Random Oracle Methodology, Revisited. *J. ACM* **1998**, *51*, 209–218. [CrossRef]
31. Zhandry, M. How to Record Quantum Queries, and Applications to Quantum Indifferentiability. In *CRYPTO 2019*; Part II; Springer: Cham, Switzerland, 2019; pp. 239–268.
32. Don, J.; Fehr, S.; Majenz, C.; Schaffner, C. Online-Extractability in the Quantum Random-Oracle Model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Cham, Switzerland, 2002.
33. Liu, Q.; Zhandry, M. Revisiting Post-quantum Fiat-Shamir. In *Advances in Cryptology—CRYPTO 2019. CRYPTO 2019. Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2019; pp. 326–355
34. Unruh, D. Quantum Proofs of Knowledge. In *EUROCRYPT 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 135–152.
35. Lang, S. *Algebra, GTM 211*; Springer: Berlin, Germany, 2002.
36. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: New York, NY, USA, 2010.
37. Watrous, J. Quantum Computing, *Lecture Notes*. 2006. Available online: https://cs.uwaterloo.ca/~watrous/QC-notes/ (accessed on 22 October 2024).
38. Boneh, D.; Zhandry, M. Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World. In *Advances in Cryptology – CRYPTO 2013. CRYPTO 2013. Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 361–379.
39. Lyubashevsky, V.; Micciancio, D. Generalized Compact Knapsacks Are Collision Resistant. In *ICALP 2006*; part 2; Springer: Berlin/Heidelberg, Germany, 2006; pp. 144–155.
40. Lyubashevsky, V.; Peikert, C.; Regev, O. A toolkit for Ring-LWE cryptography. In *EUROCRYPT 2013*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7881, pp. 35–54.
41. Peikert, C.; Rosen, A. Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices. In *TCC 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 145–166.
42. Abdalla, M.; Fouque, P.A.; Lyubashevsky, V.; Tibouchi, M. Tightly-Secure Signatures from Lossy Identification Schemes. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, 15–19 April 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 572–590.
43. Ducas, L.; Durmus, A. Ring-lwe in polynomial rings. In *PKC 2012*; LNCS 7293; Springer: Cham, Switzerland, 2012; pp. 34–51.
44. Lyubashevsky, V.; Peikert, C.; Regev, O. On ideal lattices and learning with errors over rings. *J. ACM* **2013**, *60*, 43:1–43:35. [CrossRef]
45. Blake, I.F.; Gao, S.; Mullin, R.C. Explicit Factorization of $x^{2^k}+1$ over F_p with Prime $p \equiv 3 \bmod 4$. *Appl. Algebra Eng. Commun. Comput.* **1993**, *4*, 89–94. [CrossRef]

46. Léo Ducas, B.C.; Wesolowski, B. Short stickelberger class relations and application to ideal-svp. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, 30 April–4 May 2017; Springer: Cham, Switzerland, 2017.
47. Micciancio, D.; Regev, O. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.* **2007**, *37*, 267–302. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.