Secret Sharing with Publicly Verifiable Deletion*

Jonathan Katz[1] and Ben Sela[2]

[1] Google and University of Maryland
jkatz2@gmail.com
[2] University of Maryland
benjsela@cs.umd.edu

Abstract. Certified deletion, an inherently quantum capability, allows a party holding a quantum state to prove that they have deleted the information contained in that state. Bartusek and Raizes recently studied certified deletion in the context of secret sharing schemes, and showed constructions with *privately* verifiable proofs of deletion that can be verified only by the dealer who generated the shares. We give two constructions of secret sharing schemes with *publicly* verifiable certified deletion. Our first construction is based on the post-quantum security of the LWE problem, and each share requires a number of qubits that is linear in the size of an underlying classical secret sharing scheme for the same set of authorized parties. Our second construction is based on a more general assumption—the existence of post-quantum one-way functions— but requires an asymptotically larger number of qubits relative to the share size of the underlying classical scheme.

1 Introduction

Secret-sharing schemes [15] allow a dealer to split a secret s into shares such that only certain authorized subsets of those shares (as defined by some monotone access structure) can recover s, while all other subsets of the shares reveal no information about s. Secret sharing has been studied extensively in both the computational and information-theoretic settings, and constructions in both settings are known for various access structures.

Recent work of Bartusek and Raizes [5] initiated the study of secret sharing with *certified deletion*. Such schemes consider shares that are quantum states, and allow a party given a share to generate a (classical) proof that they have deleted their share. Bartusek and Raizes put forth two (incomparable) notions of security in this setting, and show information-theoretic constructions of schemes for threshold access structures that allow for *privately verifiable* proofs of deletion, where the proofs are verifiable only by the dealer who generated the initial shares. They leave open the questions of whether it is possible to construct secret-sharing schemes with *publicly verifiable* proofs of deletion, and whether can one construct secret-sharing schemes with (privately or publicly verifiable) certified deletion for arbitrary monotone access structures.

* Work supported in part by NSF award CNS-2154705.

We resolve both of these questions in the affirmative. We give two constructions of secret-sharing schemes with publicly verifiable proofs of deletion; both constructions inherit the access structure of some underlying (classical) secret-sharing scheme and hence can support any monotone access structure. Our first construction relies on the post-quantum hardness of the LWE problem, and each share requires a number of qubits that is linear in the size of the share size of the underlying classical secret-sharing scheme. Additionally this scheme only has computational security. Our second construction relies only on the existence of a post-quantum one-way function (OWF), but the number of qubits used to encode each share is quadratic in the number of parties.

Although our first construction offers only computational secrecy, we show that any such scheme can be upgraded to satisfy *certified everlasting security*; this roughly means that once a share is deleted the information contained in that share is inaccessible even to a computationally unbounded adversary. Note that this implies the standard notion of information-theoretic secrecy.

1.1 Related Work

The first example of certified deletion is due to Unruh [16] in the context of revocable encryption. Certified deletion was first formalized by Broadbent and Islam [7], who develop a symmetric-key encryption scheme based on BB84 states [18] whose ciphertexts can be certifiably deleted. Certified deletion has since been explored in many other cryptographic settings [2, 3, 5, 10–12, 14].

Public verifiability for proofs of deletion has also been considered [4, 13, 14]. Of particular relevance to our work are the results of Bartusek et al. [4]. Their technique allow one to encode a classical string in a quantum state such that an appropriate preimage of a one-way function serves as proof that the encoded string was deleted. A roughly equivalent result using different techniques was shown concurrently by Kitigawa et al. [13].

1.2 Open Problems and Future Work

Our work leaves open several questions. Our constructions both rely on a classical secret-sharing scheme as a building block, and require the scheme to have the (non-standard) property that the shares in any unauthorized subset are uniform. While schemes satisfying this property are known for both threshold and general monotone access structures [8], it would be interesting to extend our results to work for arbitrary (perfectly secret) secret-sharing schemes, which may potentially allow for smaller share size.

In our work we consider only adaptive certified deletion, which is a simple and intuitively appealing definition. Bartusek and Raizes also propose an alternate notion of security called no-signaling certified deletion. The construction by Bartusek and Raizes satisfying that notion can be easily lifted to achieve public verifiability based on any post-quantum one-way function with sub-exponential security (see Appendix D). However it is an open problem to construct schemes that are provably secure in this setting from milder assumptions.

2 Technical Overview

We now give a more detailed overview of our techniques and results.

2.1 Background

Notions of security. Bartusek and Raizes [5] introduce two notions of security for secret sharing with certified deletion, which we informally recall here:

No-signaling certified deletion: Let $\mathcal{A} = (\mathcal{A}_1, ..., \mathcal{A}_\ell)$ be a set of ℓ non-communicating adversaries, with each adversary $\mathcal{A}_i$ associated with some unauthorized set $A_i \subset [n]$ and all $\{A_i\}$ disjoint.
- The challenger creates shares $|\mathsf{qsh}_1\rangle, ..., |\mathsf{qsh}_n\rangle \leftarrow \mathsf{Share}(s)$, and each $\mathcal{A}_i$ receives $\{|\mathsf{qsh}_j\rangle\}_{j \in A_i}$.
- Each $\mathcal{A}_i$ deletes some subset of its shares subject to the constraint that the set of non-deleted shares across all $\{\mathcal{A}_i\}$ does not form an authorized set. Each $\mathcal{A}_i$ then outputs its state.

A secret-sharing scheme has no-signaling certified deletion if the combined outputs of the $\{\mathcal{A}_i\}$ are (almost) independent of the secret that was shared.

Adaptive certified deletion: Let $\mathcal{A}$ be an adversary.
- The challenger creates shares $|\mathsf{qsh}_1\rangle, ..., |\mathsf{qsh}_n\rangle \leftarrow \mathsf{Share}(s)$. The adversary $\mathcal{A}$ can then adaptively obtain and delete shares subject to the constraint that, at any point in the experiment, the shares that have been obtained but not deleted never form an authorized set.
- $\mathcal{A}$ outputs its state.

A secret-sharing scheme has adaptive certified deletion if the output of $\mathcal{A}$ is (almost) independent of the secret being shared.

Bartusek and Raizes [5] prove that no-signaling security does not imply adaptive security by showing an explicit construction of a secret-sharing scheme that satisfies the former but not the latter. They leave open the other direction, though they note certain difficulties in trying to prove that adaptive security implies no-signalling security. We prove in Appendix C that the definitions are in fact incomparable, and there exists a secret-sharing scheme satisfying adaptive security but not no-signalling security.

In the remainder of our work, we consider adaptive certified deletion only.

Prior work. Bartusek and Raizes show two constructions of secret-sharing schemes with (privately verifiable) certified deletion. Their first scheme, which can be based on any underlying (classical) secret-sharing scheme, satisfies no-signalling security but not adaptive security. As noted earlier, we believe this construction can be adapted fairly easily to achieve public verifiability based on post-quantum OWFs with sub-exponential security; this still leaves open the question of public verifiability for schemes achieving adaptive security. Their second construction, which achieves adaptive security, is based on a specific (classical) secret-sharing scheme for threshold access structures, and it is not clear how to extend the scheme for general access structures. In summary, for the adaptive security definition in which we are interested, there is no prior construction of a secret-sharing scheme for general access structures, or achieving public verifiability.

2.2 The Starting Point of Our Approach

Our approach for constructing an adaptively secure secret-sharing scheme differs from the approaches taken by Bartusek and Raizes in constructing their schemes. We provide a high-level overview here.

We begin by recalling a technique for publicly verifiable deletion introduced by Bartusek et al. [4]. They provide a way to encode a bit b in a quantum state $|\psi\rangle$ such that an adversary $\mathcal{A}$ can perform a measurement on $|\psi\rangle$ that "deletes" b and produces a publicly verifiable proof of that fact. The bit is encoded by choosing $x_0, x_1 \leftarrow \{0,1\}^\kappa$ and encoding b as

$$|\psi\rangle = |x_0\rangle + (-1)^b |x_1\rangle;$$

additionally, $y_0 = f(x_0)$ and $y_1 = f(x_1)$ are published, where f is a one-way function. A deletion certificate is a preimage of either y_0 or y_1. To delete the bit and obtain such a certificate, $\mathcal{A}$ simply measures $|\psi\rangle$ in the computational basis. On the other hand, if $\mathcal{A}$ does not delete the bit and is given $x_0 \oplus x_1$, then it can perform a measurement of $|\psi\rangle$ in the Hadamard basis to learn a string d such that $d \cdot (x_0 \oplus x_1) = b$; i.e., given $x_0 \oplus x_1$ it can learn b.

Bartusek et al. [4] show that it is infeasible for a computationally bounded adversary who does not know $x_0 \oplus x_1$ to generate a deletion certificate and still learn b (even if it is given $x_0 \oplus x_1$ after generating the deletion certificate). On the other hand, if $\mathcal{A}$ is given $x_0 \oplus x_1$ *before* being asked to produce a deletion certificate, the Gentle Measurement Lemma (Lemma 1) implies that $\mathcal{A}$ can learn b without disturbing the state $|\psi_b\rangle$ too much. It can then perform a measurement in the computational basis to produce a (false) proof of deletion.

This suggests the following approach for constructing a secret-sharing scheme with certified deletion. For a secret s, the dealer begins by generating classical shares $\mathsf{csh}_1, ..., \mathsf{csh}_n \leftarrow \mathsf{Share}(s)$; assume each share is an m-bit string. Then for each $i \in [n]$ the dealer encodes csh_i by creating states of the form

$$|\mathsf{qsh}_i\rangle = \bigotimes_{k \in [m]} \left(|x_0^{i,k}\rangle + (-1)^{\mathsf{csh}_{i,k}} |x_1^{i,k}\rangle \right),$$

where $x_b^{i,k}$ are uniform and independent and $\mathsf{csh}_{i,k}$ denotes the kth bit of csh_i. The dealer then publishes $y_b^{i,k} := f(x_b^{i,k})$ for all $i \in [n]$, $k \in [m]$, and $b \in \{0,1\}$.

To delete a quantum share $|\mathsf{qsh}_i\rangle$, a party measures the entire state in the computational basis to produce a sequence of preimages $x_1, ..., x_m$ such that $x_k \in \{x_0^{i,k}, x_1^{i,k}\}$ for all $k \in [m]$. In this way, x_k serves as a proof of deletion for the kth bit of csh_i, which can be publicly verified by checking that $f(x_k)$ yields the appropriate image in the verification key.

While the above allows for certified deletion, we have not yet shown how the secret can be reconstructed from an authorized set of shares! Learning the classical share csh_i (that can be used with other shares to reconstruct the secret s) requires knowledge of the strings $\{x_0^{i,k} \oplus x_1^{i,k}\}_{k\in[m]}$. However, as mentioned earlier, if those strings are revealed at the outset to an adversary then the deletion proof becomes meaningless. Somehow we must allow the preimages to be used by an authorized set of parties, but otherwise remain hidden.

A seemingly natural solution to the above problem is to split the preimages among the parties by using a classical secret sharing scheme so that each share is of the form $(|\mathsf{qsh}\rangle, \mathsf{csh})$. However this approach is vulnerable to the following attack. Note that any portion of each share which is classical cannot be deleted. Thus, an adaptive adversary can alternate corrupting and deleting shares until an authorized set of shares have been corrupted, at which point the adversary holds an authorized set of classical shares $\{\mathsf{csh}_i\}_{i\in A}$. Now the adversary can reconstruct the strings $\{x_0^{i,k} \oplus x_1^{i,k}\}$ which renders any future deletion proofs meaningless. Therefore we need a way of hiding the preimages so that no useful information is left behind after a share is deleted.

We discuss two approaches for achieving exactly this.

2.3 Construction from LWE

Our first approach (Construction 1) is to instantiate a reconstruction oracle obfuscating a program to perform reconstruction. In more detail, define a reconstruction program Rec as follows. Hardcode the strings $\{x_0^{i,k} \oplus x_1^{i,k}\}_{i\in[n],k\in[m]}$. Then, on input a set of the form $\{(d_{i,k}, i, k)\}_{k\in[m],i\in A\subseteq[n]}$, Rec does:

- Compute $\mathsf{csh}'_{i,k} := d_{i,k} \cdot (x_0^{i,k} \oplus x_1^{i,k})$ for all $i \in A, k \in [m]$.
- Set $\mathsf{csh}'_i = \mathsf{csh}'_{i,1} \cdots \mathsf{csh}'_{i,m}$ for $i \in A$, and output $\mathsf{Reconstruct}(\{\mathsf{csh}'_i\}_{i\in A})$, where $\mathsf{Reconstruct}$ is the reconstruction procedure for the underlying classical secret-sharing scheme.

When the $\{d_{i,k}\}_{i\in A,k\in[m]}$ are the results of Hadamard measurements of the corresponding quantum shares, then the above program outputs the original secret s whose classical shares were encoded in the quantum states as discussed in the previous section.

We obfuscate the above program using compute-and-compare obfuscation [17], which can be constructed based on the post-quantum hardness of LWE. Let $P : \{0,1\}^{\ell_{\mathsf{in}}} \to \{0,1\}^{\ell_{\mathsf{out}}}$ be a function, and define the following compute-and-compare program:

$$\mathsf{CC}[P, \mathsf{lock}, z](x) = \begin{cases} z & P(x) = \mathsf{lock} \\ \bot & \text{otherwise.} \end{cases}$$

A compute-and-compare obfuscator takes a program of the above form, and outputs another program $\widetilde{P}$ which is functionally equivalent, and with the security guarantee that if it is computationally infeasible for an adversary given P to compute lock, then $\widetilde{P}$ hides all details of P.

This suggests the following attempt at a secret-sharing scheme with certified deletion. On input s, generate classical shares $\mathsf{csh}_1, ..., \mathsf{csh}_n \leftarrow \mathsf{Share}(s)$ and encode them in quantum states $|\mathsf{qsh}_i\rangle$ as discussed above. Let Rec be the reconstruction program with the $\{x_0^{i,k} \oplus x_1^{i,k}\}_{i\in[n],k\in[m]}$ hardcoded as discussed earlier. Then give the ith party the quantum share $|\mathsf{qsh}_i\rangle$, and give all parties the same obfuscated program $\widetilde{\mathsf{Rec}} \leftarrow \mathsf{CC.Obf}(\mathsf{CC}[\mathsf{Rec}, s, s])$. Now, given an authorized set of shares, parties can measure each $|\mathsf{qsh}_i\rangle$ in the Hadamard basis and evaluate $\widetilde{\mathsf{Rec}}$ on the measurement results to obtain s. Intuitivly, security of the compute-and-compare obfuscator implies that $\widetilde{P}$ hides the details of Rec—and in particular hides the hardcoded preimages—so that our deletion mechanism functions properly.

Note, however, that security of the compute-and-compare obfuscation depends on the unpredictability of the lock value; a problem arises if s is not a high-entropy value! To remedy this issue, we make the following modification: Instead of using s itself as the lock, we sample a uniform value lock, and let the states $\{|\mathsf{qsh}_i\rangle\}_{i\in[n]}$ encode classical shares of lock rather than of s.

We remark that this construction achieves only computational secrecy (even against a static adversary who simply corrupts an unauthorized set and does not delete anything) because the compute-and-compare obfuscation is only computationally hiding. We discuss in Section 2.5 how to upgrade the scheme to achieve information-theoretic secrecy and, in fact, an even stronger notion we call everlasting security.

2.4 Construction from One-Way Functions

Recall that our starting point was to create classical shares $\{\mathsf{csh}_i\}_{i\in[n]} \leftarrow \mathsf{Share}(s)$ of the secret, and then encode these shares in states of the form

$$|\mathsf{qsh}_i\rangle := \bigotimes_{k\in[m]} \left(|x_0^{i,k}\rangle + (-1)^{\mathsf{csh}_{i,k}}|x_1^{i,k}\rangle\right).$$

If we want to avoid compute-and-compare obfuscation then we need some other method of hiding the xor of the preimages in such a way that an authorized set of parties can either recover them directly or otherwise make use of them to recover the classical secret shares.

We first present the following (flawed) construction. In addition to giving each party the state $|\mathsf{qsh}_i\rangle$ defined above, we can also create classical shares of the preimages $\{\mathsf{csh}_i^x\}_{i\in[n]} \leftarrow \mathsf{Share}\left(\{x_0^{i,k}, x_1^{i,k}\}\right)$, and give party i the classical share csh_i^x. Now an authorized set of parties can use the shares for the preimages to recover the preimages, and then use the recovered preimages to learn the classical shares for the secret s. However, this scheme is vulnerable to the following attack:

- Alternate corrupting and deleting shares until an authorized set of classical shares $\{\mathsf{csh}_i^x\}_{i\in S}$ has been recovered, and use them to reconstruct the preimages $\{x_0^{i,k}, x_1^{i,k}\}$.

– Continue to corrupt and delete shares as follows. For each newly corrupted share $(\mathsf{csh}_i, |\mathsf{qsh}_i^x\rangle)$, use the previously recovered preimages to extract csh_i^s from $|\mathsf{qsh}_i^s\rangle$ without disturbing the quantum state (this is possible by the Gentle Measurement Lemma). Then delete the share and continue until an authorized set of shares $\{\mathsf{csh}_i^s\}_{i\in S}$ have been recovered. Reconstruct the secret $s \leftarrow \mathsf{Reconstruct}(\{\mathsf{csh}_i^s\}_{i\in S})$.

While the above attack breaks our construction, we make the following observation. For the adversary to corrupt an authorized set of shares, at least 1 share must be deleted in the preimage extraction step above. While proofs of deletion are meaningless if the adversary knows the preimages, this first deletion proof must have been returned by the adversary prior to learning the preimages. Therefore the classical share csh_i^s encoded in the corresponding state $|\mathsf{qsh}_i^s\rangle$ was truly deleted. As there are at most $n-1$ shares which have not been deleted after the preimage extraction step, the second step of the attack can only succeed if there exists an authorized set of size $n-1$ or smaller.

With the above in mind, our construction seems secure (at least against this particular attack) if we have an (n,n)-access structure. We now modify the construction so that the attack fails for any access structure such that the smallest authorized set is of size at least $n-1$. The sharing procedure for our modified construction is as follows.

– Create secret shares $\{\mathsf{csh}_i^s\}_{i\in[n]} \leftarrow \mathsf{Share}(s)$ of the secret. Sample a set of uniform preimages $\mathsf{Pre}_1 := \{x_0^{i,k}, x_1^{i,k}\}$, and use them to encode the classical shares of s into the corresponding quantum states $|\mathsf{qsh}_i^s\rangle$.
– Create secret shares for the above preimages $\{\mathsf{csh}_i^x\}_{i\in S} \leftarrow \mathsf{Share}(\{x_0^{i,k}, x_1^{i,k}\})$. Sample a set of uniform preimages $\mathsf{Pre}_2 := \{z_0^{i,k}, z_1^{i,k}\}$, and use them to encode the shares $\{\mathsf{csh}_i^x\}_{i\in[n]}$ into the corresponding states $|\mathsf{qsh}_i^x\rangle$.
– Create secret shares $\{\mathsf{csh}_i^z\}_{i\in[n]} \leftarrow \mathsf{Share}(\{z_0^{i,k}, z_1^{i,k}\})$. Finally set the ith share as the tuple $(\mathsf{csh}_i^z, |\mathsf{qsh}_i^x\rangle, |\mathsf{qsh}_i^s\rangle)$.

Consider the analogue of our attack on the above construction. Our attack first corrupts and deletes shares in order to extract the first set of preimages $\{z_0^{i,k}, z_1^{i,k}\}$. Then the adversary continues to corrupt and delete shares, using the previously extracted preimages to extract shares of $\{x_0^{i,k}, x_1^{i,k}\}$ from the states $|\mathsf{qsh}_i^x\rangle$ as it goes. Once the second set of preimages is recovered, the adversary continues to corrupt and delete shares, this time using the second set of preimages to extract shares of the secret.

We claim that at least one deletion must take place for each set of extracted preimages in the above attack. If the preimages in Pre_1 are recovered without the adversary outputting a proof of deletion, then the adversary must hold an authorized set of shares which is not permitted by the experiment. Note that prior to learning Pre_1, any proof of deletion for a share i truly deletes the information encoded in $|\mathsf{qsh}_i^x\rangle$ and $|\mathsf{qsh}_i^s\rangle$. Therefore, after recovering Pre_1, the only shares of Pre_2 which the adversary has access to are the ones currently in its view. It follows that in order to obtain an authorized set of shares for Pre_2, at least one more deletion must be output by the adversary. However this means that by the time Pre_2 is recovered, there are at most $n-2$ shares of s which have not yet been deleted. Thus, assuming all authorized sets are of size at least $n-1$, the above attack fails. Therefore if we iterate the above construction k times, the attack should fail for any access structure all of whose authorized sets have size at least $n-k$.

There is one additional issue to resolve. If the secret-sharing scheme has shares that are larger than the secret (which is the case for many schemes for general access structures), then each iteration of the above construction will have a share size equal to some multiplicative factor of the previous share size, and so iterating n times will result in a share size that is exponential in n. To address this, we generate the preimages using a PRF with a different key k_ℓ at each level ℓ, and then secret-share the key in the next level.

2.5 Everlasting Security

Our first secret-sharing construction does not have information-theoretic secrecy, as the compute-and-compare program contains a (classical) encryption of the secret. It turns out we can upgrade our construction, and more generally any computational secret-sharing scheme, to achieve information-theoretic secrecy, and in fact an even stronger property we call everlasting security.

Let $\mathsf{SS}_{\mathsf{comp}}$ be a computational secret-sharing scheme with adaptive certified deletion, and let $\mathsf{SS}_{\mathsf{classical}}$ be an information-theoretic classical secret-sharing scheme for the same access structure. Recall the intuition behind the deletion mechanism: as long as the preimages are hidden from the adversary, a proof of deletion destroys any information about the underlying bit. With this in mind, rather than creating shares of the secret itself with $\mathsf{SS}_{\mathsf{comp}}$, we generate classical shares $\mathsf{csh}_1, ...\mathsf{csh}_n \leftarrow \mathsf{SS}_{\mathsf{classical}}.\mathsf{Share}(s)$, and we encode each csh_i in a state of the form $|\mathsf{qsh}_i^s\rangle = \bigotimes_{k\in[m]}(|x_0^{i,k}\rangle + (-1)^{\mathsf{csh}_{i,k}}|x_1^{i,k}\rangle)$, where $\mathsf{csh}_{i,k}$ is the kth bit of csh_i, and the preimages $x_b^{i,k}$ are evaluations of a PRF with a uniform key k_0. We then hide k_0 using $\mathsf{SS}_{\mathsf{comp}}$, i.e., we compute shares $|\mathsf{qsh}_1^{\mathsf{PRF}}\rangle, ..., |\mathsf{qsh}_n^{\mathsf{PRF}}\rangle \leftarrow \mathsf{SS}_{\mathsf{comp}}.\mathsf{Share}(k_0)$. Finally we output the set of shares $\{|\mathsf{qsh}_i\rangle \otimes |\mathsf{qsh}_i^{\mathsf{PRF}}\rangle\}_{i\in[n]}$.

Security of $\mathsf{SS}_{\mathsf{comp}}$ implies that a quantum polynomial-time (QPT) adversary outputting a proof of deletion for $|\mathsf{qsh}_i^{\mathsf{PRF}}\rangle$ has indeed deleted any information about the corresponding classical information csh_i in an information theoretic sense. On the other hand, even if an unbounded adversary later breaks the computational scheme $\mathsf{SS}_{\mathsf{comp}}$ to recover the PRF key (and by extension the preimages), one cannot recover the classical shares if they were deleted by a bounded adversary. Provided that $\mathsf{SS}_{\mathsf{classical}}$ is information-theoretic, we have that the secret remains hidden.

3 Preliminaries

We let λ denote the security parameter, and let $\mathsf{negl}(\cdot)$ be an unspecified negligible function. For $n \in \mathbb{N}$, let $[n] = \{1, ..., n\}$. For a finite set S, we write $s \leftarrow S$

to denote that s is sampled uniformly from S. For a distribution $\mathcal{D}$, we write $x \leftarrow \mathcal{D}$ to denote that x is sampled according to $\mathcal{D}$. For two distributions $\mathcal{D}_1, \mathcal{D}_2$ over the same set D, their statistical distance is given by

$$\mathsf{SD}(\mathcal{D}_1, \mathcal{D}_2) = \frac{1}{2} \sum_{x \in D} \left| \Pr_{x' \leftarrow \mathcal{D}_1} [x' = x] - \Pr_{x' \leftarrow \mathcal{D}_2} [x' = x] \right| .$$

QPT stands for "quantum polynomial-time." We use the standard definition of a quantum secure one-way function (OWF).

Definition 1 (One-Way Functions). *An efficiently computable function $f : \{0,1\}^\lambda \mapsto \{0,1\}^{\ell_{\mathsf{out}}(\lambda)}$ is a one-way function if for every QPT adversary $\mathcal{A}$,*

$$\Pr_{x \leftarrow \{0,1\}^\lambda} \left[\mathcal{A}(f(x)) \in f^{-1}(f(x)) \right] \leq \mathsf{negl}(\lambda).$$

We also use the following standard definition of a quantum-secure pseudorandom function (PRF).

Definition 2 (Pseudorandom Function). *A function $F : \mathcal{K} \times \mathcal{X} \mapsto \mathcal{Y}$ is a pseudorandom function family if for any QPT adversary $\mathcal{A}$,*

$$\left| \Pr_{k \leftarrow \mathcal{K}} \left[\mathcal{A}^{|F(k,\cdot)\rangle} = 1 \right] - \Pr_{\mathcal{O} \leftarrow \mathsf{Func}(\mathcal{X}, \mathcal{Y})} \left[\mathcal{A}^{|\mathcal{O}\rangle} = 1 \right] \right| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$ denotes the set of functions from $\mathcal{X}$ to $\mathcal{Y}$, and writing $\mathcal{A}^{|\cdot\rangle}$ denotes giving $\mathcal{A}$ quantum oracle access to the indicated function.

Zhandry [20] showed that quantum-secure PRFs can be constructed from quantum-secure OWFs.

3.1 Quantum Computation

An n-qubit system is a Hilbert space $\mathbb{C}^{2^n}$. A register X is a Hilbert space to which we have assigned a name. A pure state $|\psi\rangle_{\mathsf{X}}$ on register X is a column vector with norm 1. We omit the subscript indicating the register when it is not relevant. The conjugate transpose of $|\psi\rangle$ is denoted by $\langle\psi|$. A distribution over pure states $\{(p_i, |\psi_i\rangle)\}$ is a mixed state which we represent by its density matrix $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$. The trace distance between two mixed states ρ and σ is

$$\mathsf{TD}(\rho, \sigma) = \frac{1}{2} \mathsf{Tr} \left[\sqrt{(\rho - \sigma)^\dagger (\rho - \sigma)} \right] .$$

The trace distance between two mixed states is the optimal distinguishing advantage of an unbounded adversary between the two states.

A projector is a Hermitian operator such that $\Pi^2 = \Pi$, and a projective measurement is a set of projectors $\{\Pi_i\}$ such that $\sum_i \Pi_i = I$. We make use of the following lemma, which states roughly that if a quantum computation acting on some initial mixed state ρ produces a deterministic output, then the same output can be produced without disturbing the state ρ.

Lemma 1 (Gentle Measurement Lemma [19]). *Let ρ be a quantum state on some register X, and let $\{\Pi, \mathbb{1} - \Pi\}$ be a projective measurement on X such that $\mathsf{Tr}(\Pi\rho) \geq 1 - \delta$. Let*

$$\rho' = \frac{\Pi\rho\Pi}{\mathsf{Tr}(\Pi\rho)}$$

be the post-measurement state that results from obtaining the outcome corresponding to Π. Then $\mathsf{TD}(\rho, \rho') < 2\sqrt{\delta}$.

We say two families of distributions $\mathcal{D}_0 = \{\mathcal{D}_{0,\lambda}\}_{\lambda\in\mathbb{N}}$ and $\mathcal{D}_1 = \{\mathcal{D}_{1,\lambda}\}_{\lambda\in\mathbb{N}}$ are computationally indistinguishable if for any QPT distinguisher $\mathcal{A}$, we have

$$\left| \Pr_{x\leftarrow\mathcal{D}_{0,\lambda}} [\mathcal{A}_\lambda(x) = 1] - \Pr_{x\leftarrow\mathcal{D}_{1,\lambda}} [\mathcal{A}_\lambda(x) = 1] \right| \leq \mathsf{negl}(\lambda),$$

in which case we write $\mathcal{D}_0 \approx_c \mathcal{D}_1$. Similarly, we say two families of (possibly mixed) states $\{\rho_{0,\lambda}\}_{\lambda\in\mathbb{N}}$ and $\{\rho_{1,\lambda}\}_{\lambda\in\mathbb{N}}$ are computationally indistinguishable if for any QPT adversary $\mathcal{A}$,

$$|\Pr[1 \leftarrow \mathcal{A}(\rho_{0,\lambda})] - \Pr[1 \leftarrow \mathcal{A}(\rho_{1,\lambda})]| \leq \mathsf{negl}(\lambda).$$

3.2 Compute-and-Compare Obfuscation

The following definitions are taken (almost verbatim) from [9]. Note that we only require security in the presence of classical auxiliary input.

Definition 3 (Compute-and-Compare Program). *Given a function $P : \{0,1\}^{\ell_{\mathsf{in}}} \mapsto \{0,1\}^{\ell_{\mathsf{out}}}$ along with a target value $\mathsf{lock} \in \{0,1\}^{\ell_{\mathsf{out}}}$ and a message $z \in \{0,1\}^{\ell_{\mathsf{msg}}}$, we define the compute-and-compare program:*

$$\mathsf{CC}[P, \mathsf{lock}, z](x) = \begin{cases} z & P(x) = \mathsf{lock} \\ \perp & \textit{otherwise.} \end{cases}$$

Definition 4 (Unpredictable Distributions). *Let $\mathcal{D} = \{\mathcal{D}_\lambda\}$ be family of distributions such that $\mathcal{D}_\lambda$ is a distribution over pairs of the form $(\mathsf{CC}[P, y, z], \mathsf{aux})$, where aux is a classical value. $\mathcal{D}$ is unpredictable if for all QPT algorithms $\mathcal{A}$,*

$$\Pr_{(\mathsf{CC}[P,y,z],\mathsf{aux})\leftarrow\mathcal{D}_\lambda} \left[\mathcal{A}(1^\lambda, f, \mathsf{aux}) = y\right] \leq \mathsf{negl}(\lambda).$$

Definition 5 (Compute-and-Compare Obfuscation). *A PPT algorithm $\mathsf{CC.Obf}$ is an obfuscator for the class of unpredictable distributions if for any family of distributions $\mathcal{D} = \{\mathcal{D}_\lambda\}$ belonging to the class, the following holds:*

Functionality Preserving: *there exists a negligible function negl such that for all λ, and every program P in the support of D_λ,*

$$\Pr[P \leftarrow \mathsf{CC.Obf}(1^\lambda, P) : \forall x, P(x) = P(x)] \geq 1 - \mathsf{negl}(\lambda).$$

Distributional Indistinguishability: *there exists an efficient simulator* Sim *such that:*

$$(\mathsf{CC.Obf}(1^\lambda, P), \mathsf{aux}) \approx_c (\mathsf{Sim}(1^\lambda, P.\mathsf{param}), \mathsf{aux})$$

where $(P, \mathsf{aux}) \leftarrow \mathcal{D}_\lambda$*, and* $P.\mathsf{param}$ *denotes the input size, output size, and circuit size of* P*, which are not required to be obfsucated.*

Wichs and Zirdelis [17] construct a compute-and-compare obfuscator for unpredictable distributions assuming the quantum hardness of LWE.

3.3 Secret Sharing

We now present the standard definitions of (classical) secret sharing. An access structure $\mathbb{A}$ over n parties is a collection of subsets of $[n]$. If $A \in \mathbb{A}$, we say A is *authorized*; $A \in [n]$ is unauthorized otherwise. $\mathbb{A}$ is *monotone* if $A \in \mathbb{A}$ and $A \subseteq A'$ implies $A' \in \mathbb{A}$.

Definition 6 (Secret sharing scheme). *A secret-sharing sharing scheme with message space* $\mathcal{S}$ *for monotone access structure* $\mathbb{A}$ *is a pair of algorithms* $\mathsf{SS} = (\mathsf{Share}, \mathsf{Reconstruct})$ *with the following syntax.*

- $\mathsf{Share}_{\mathbb{A}}(s)$*: A randomized algorithm that on input* $s \in \mathcal{S}$ *outputs shares* $\mathsf{sh}_1, ..., \mathsf{sh}_n$*.*
- $\mathsf{Reconstruct}_{\mathbb{A}}(\{\mathsf{sh}_k\}_{k\in A}, A)$*: On input a set of shares* $\{\mathsf{sh}_k\}_{k\in A}$ *and* $A \subseteq [n]$*, outputs some* $s' \in \mathcal{S}$ *if* $A \in \mathbb{A}$*, and otherwise outputs* $\perp$*.*

Correctness: *For all* $s \in \mathcal{S}$*, and any authorized subsets* $A \in \mathbb{A}$*, we have*

$$\Pr\left[(\mathsf{sh}_1, ..., \mathsf{sh}_n) \leftarrow \mathsf{Share}(s) : \mathsf{Reconstruct}(\{\mathsf{sh}_k\}_{k\in A}) = 1\right] = 1$$

Privacy: *For any unauthorized set* $A \notin \mathbb{A}$*, and for any pair of secrets* $s_0, s_1 \in \mathcal{S}$*, we have*

$$\mathsf{SD}\left(\{\mathsf{sh}_i^0\}_{i\in A}, \{\mathsf{sh}_i^1\}_{i\in A}\right) = 0,$$

where $\mathsf{sh}_1^b, ..., \mathsf{sh}_n^b \leftarrow \mathsf{Share}(s_b)$ *for* $b \in \{0, 1\}$*.*

Uniformity: For our results, we require a secret-sharing scheme with the (non-standard) property that the shares of any unauthorized subset are uniformly distributed. Formally, we require that shares are always m-bit strings for some m, and that for any unauthorized set $A \notin \mathbb{A}$ the distributions

$$\left\{\{\mathsf{sh}_i\}_{i\in[n]} \leftarrow \mathsf{Share}(s) : \{\mathsf{sh}_i\}_{i\in A}\right\} \quad \text{and} \quad \left\{\{\mathsf{sh}_i\}_{i\in[n]} \leftarrow \{0,1\}^m : \{\mathsf{sh}_i\}_{i\in A}\right\}$$

are identical. Chandran et al. [8] proved that Shamir's threshold secret-sharing scheme [15] as well the Benaloh-Leichter scheme [6] for general monotone access structures both satisfy this property.

3.4 Secret Sharing with Verifiable Deletion

We now give definitions of secret sharing with certified deletion. Our definitions are based on those of Bartusek and Raizes [5], modified for publicly verifiable deletion (PVD) and computational secrecy.

Definition 7 (Secret Sharing with PVD). *A secret-sharing scheme with certified deletion for message space $\mathcal{S}$ and a monotone access structure $\mathbb{A}$ over n parties consists of the following four algorithms:*

- $\mathsf{Share}_{\mathbb{A}}(1^\lambda, s)$*: A randomized algorithm that on input a security parameter $\lambda \in \mathbb{N}$ and a secret s, outputs n share registers $\mathsf{Sh}_1, ..., \mathsf{Sh}_n$, and a classical verification key* vk.
- $\mathsf{Reconstruct}_{\mathbb{A}}(\{\mathsf{Sh}_i\}_{i\in A})$*: On input a set of share registers, outputs either s or $\perp$.*
- $\mathsf{Delete}_{\mathbb{A}}(\mathsf{Sh}_i)$*: On input a share register outputs a classical certificate of deletion* cert.
- $\mathsf{Verify}_{\mathbb{A}}(\mathsf{vk}, i, \mathsf{cert})$*: On input a verification key* vk*, an index $i \in [n]$, and a certificate of deletion* cert*, outputs either $\top$ (indicating accept) or $\perp$ (indicating reject).*

Correctness of Reconstruction: *For all $\lambda \in \mathbb{N}$ and all $A \in \mathbb{A}$,*

$$\Pr\left[(\mathsf{Sh}_1, ..., \mathsf{Sh}_n, \mathsf{vk}) \leftarrow \mathsf{Share}_{\mathbb{A}}(1^\lambda, s) : \mathsf{Reconstruct}_{\mathbb{A}}\left(\{\mathsf{Sh}_i\}_{i\in A}\right) = s\right] = 1.$$

Correctness of Deletion: *For all $\lambda \in \mathbb{N}$ and all $i \in [n]$,*

$$\Pr\left[\begin{array}{r}(\mathsf{Sh}_1, \ldots, \mathsf{Sh}_n, \mathsf{vk}) \leftarrow \mathsf{Share}_{\mathbb{A}}(1^\lambda, s)\\ \mathsf{cert} \leftarrow \mathsf{Delete}(\mathsf{Sh}_i)\end{array} : \mathsf{Verify}_{\mathbb{A}}(\mathsf{vk}, i, \mathsf{cert}) = \top\right] = 1.$$

Adaptive certified deletion [5]. The security notion we aim to satisfy involves an adversary who can adaptively learn and delete shares, provided that the set of shares which has been learned but not deleted never forms an authorized set at any point in the experiment. A formal description of the security game $\mathsf{SS\text{-}ACD}$ modeling this type of adversary follows.

Definition 8. *Let $\mathcal{A}$ be an adversary with internal register* State. *Let $\mathbb{A}$ be an access structure, and let s be a secret. Define $\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s)$ as follows:*

- *Generate shares and verification key $(\mathsf{Sh}_1, ..., \mathsf{Sh}_n, \mathsf{vk}) \leftarrow \mathsf{Share}(1^\lambda, s)$. Initialize the corruption set $C = \emptyset$ and the deleted set $D = \emptyset$. Initialize the internal register* State *of the adversary $\mathcal{A}$ with $|\mathsf{vk}\rangle \otimes |\psi\rangle$.*
- *The adversary may then repeatedly do one of three things:*
  - *Request to corrupt share $j \in [n]$. When the adversary chooses this option, add j to C and give $\mathcal{A}$ the corresponding share register Sh_j. If $C \setminus D \in \mathbb{A}$, then immediately abort the experiment and output $\perp$.*
  - *Delete a share by outputting an index $j \in [n]$ and a certificate cert_j. If $\mathsf{Verify}_{\mathbb{A}}(\mathsf{vk}, j, \mathsf{cert}_j) = \top$, add j to D. Otherwise, abort the experiment and output $\perp$.*

- *End the experiment by outputting $\mathcal{A}$'s internal register* State.

– *Output $\mathcal{A}$'s internal register* State*, unless the experiment has already aborted.*

A secret sharing scheme for access structure $\mathbb{A}$ has computational adaptive PVD if for any QPT adversary $\mathcal{A}$, any state $|\psi\rangle$, and any secrets s_0, s_1,

$$\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s_0) \approx_c \mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s_1).$$

We say that a secret sharing scheme for access structure $\mathbb{A}$ has computational adaptive PVD with everlasting security if for any QPT adversary $\mathcal{A}$, any state $|\psi\rangle$, and any pair of secrets (s_0, s_1),

$$\mathsf{TD}\left(\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s_0), \mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s_1)\right) = \mathsf{negl}(\lambda).$$

4 Construction from Compute-and-Compare Obfuscation

In this section we present a construction for a computational secret-sharing scheme with adaptive PVD. Our construction takes as input a secret sharing scheme with the property that any unauthorized set of shares are perfectly uniform, and generates a computational scheme with certified deletion for the same access structure. Our scheme is secure assuming quantum secure compute-and-compare obfuscation, which in turn can be based off the post-quantum hardness of LWE [17]. Our construction does does not have everlasting security. However in Section 4.2, we show how any computational scheme can be upgraded to satisfy this property assuming the existence of a one-way function.

Construction 1 *Let $f : \{0,1\}^{\kappa(\lambda)} \mapsto \{0,1\}^{\ell(\lambda)}$ be a one-way function. Let* $\mathsf{SS} = (\mathsf{Share}, \mathsf{Reconstruct})$ *be a secret-sharing scheme for monotone access structure $\mathbb{A}$ with shares in $\{0,1\}^m$ that satisfies the uniformity property. Let* $\mathsf{CC.Obf}$ *be a post-quantum compute-and-compare obfuscator for unpredictable distributions.*

– $\mathsf{Share}_{\mathbb{A}}(1^\lambda, s)$*: On input a secret s, sample* $\mathsf{lock} \leftarrow \{0,1\}^m$*. Generate shares* $\mathsf{csh}_1, ..., \mathsf{csh}_n \leftarrow \mathsf{SS.Share}_{\mathbb{A}}(\mathsf{lock})$*. Sample a PRF key $k_0 \leftarrow \mathcal{K}$.*
  For each $i \in [n]$ do the following:
  - *For each $k \in [m]$ and $b \in \{0,1\}$, compute $x_b^{i,k} = F(k_0, b||i||k)$ and set $y_b^{i,k} = f(x_b^{i,k})$.*
  - *Prepare the quantum state*

$$|\mathsf{qsh}_i\rangle = \bigotimes_{k\in[m]} \left(|x_0^{i,k}\rangle + (-1)^{\mathsf{csh}_{i,k}}|x_1^{i,k}\rangle\right).$$

– *Let* Rec *be the following program:*

$\mathsf{Rec}\left(\{(d_{i,k}, i, k)\}_{i \in A \subseteq [n], k \in [m]}\right)$

1: Hardcode the strings $\{x_0^{i,k} \oplus x_1^{i,k}\}_{i \in [n], k \in [m]}$
2: **for** $i \in A, k \in [m]$ **do**
3: $\quad \mathsf{csh}'_{i,k} := d_{i,k} \cdot (x_0^{i,k} \oplus x_1^{i,k})$
4: **end for**
5: **for** $i \in S$ **do**
6: $\quad \mathsf{csh}'_i := \mathsf{csh}'_{i,1}, ..., \mathsf{csh}_{i,m}$
7: **end for**
8: $\mathsf{lock}' = \mathsf{Reconstruct}(\{\mathsf{csh}_i\}_{i \in A})$
9: return lock'

– *Generate an obfuscated program* $\widetilde{\mathsf{Rec}} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{CC}[\mathsf{Rec}, \mathsf{lock}, s])$ *with* lock *as the target value, and the secret* s *as the hidden value.*

– *Initialize register* Sh_i *to* $\left(\widetilde{\mathsf{Rec}}, |\mathsf{qsh}_i\rangle\right)$.

– *Set the public verification key as* $\mathsf{vk} = \{y_0^{i,k}, y_1^{i,k}\}_{i \in [n], k \in [m]}$.

– $\mathsf{Reconstruct}_{\mathbb{A}}(\{\mathsf{Sh}_i\}_{i \in A \subseteq [n]}, A)$: *For* $i \in A$ *parse the quantum shares* Sh_i *as*

$$|\mathsf{qsh}_i\rangle = \bigotimes_{k \in [m]} \left(|x_0^{i,k}\rangle + (-1)^{\mathsf{csh}_{i,k}} |x_1^{i,k}\rangle\right).$$

Measure each $\left(|x_0^{i,k}\rangle + (-1)^{\mathsf{csh}_{i,k}} |x_1^{i,k}\rangle\right)$ *in the Hadamard basis to obtain measurement result* $d_{i,k}$. *Compute* $\widetilde{\mathsf{Rec}}(\{(d_{i,k}, i, k)\}_{i \in A, k \in [m]})$ *and output whatever it outputs.*

– $\mathsf{Delete}_{\mathbb{A}}(\mathsf{Sh}_i)$: *Parse the quantum share as* $|\mathsf{qsh}_i\rangle = \left(\widetilde{\mathsf{Rec}}, \bigotimes_{k \in [m]} |\mathsf{qsh}_{i,k}\rangle\right)$. *Measure each* $|\mathsf{qsh}_{i,k}\rangle$ *in the computational basis to obtain measurement result* $x^{i,k}$. *Output* $\{x^{i,k}\}_{k \in [m]}$.

– $\mathsf{Verify}_{\mathbb{A}}(\mathsf{vk}, i, \mathsf{cert})$: *Parse* cert *as* $x_1, ..., x_m \in \{0,1\}^\kappa$. *If* $f(x_k) \in \{y_0^{i,k}, y_1^{i,k}\}$ *for all* $k \in [m]$, *output* $\top$, *otherwise output* $\bot$.

4.1 Proof of Security for Construction 1

We give a brief roadmap for our proof. Recall that the security guarantee of the compute-and-compare obfuscator only applies if an adversary $\mathcal{A}$ given the program Rec in the clear, together with some auxiliary input aux, cannot predict the lock value. In our setting the auxiliary input will take the form of some unauthorized set of quantum shares, together with any information that was leftover from additional deleted shares. This poses a problem, since the security definition for the compute-and-compare obfuscator does not allow the auxiliary input to be chosen adaptively based on the obfuscated program.

To get around this problem, we will first appeal to the security of the underlying classical secret sharing scheme to argue that we can replace the classical shares of lock with uniform strings (Lemma 2). The structure of this portion of the proof is based on techniques used in [5], with changes based on our differing deletion mechanism. Now that the auxiliary input is completely independent of lock, we can appeal to the security of the compute-and-compare obsfucator to argue that the secret remains hidden.

Formal proof. We introduce some additional notation. Fix a secret s, and a subset $S \subseteq [n]$. For a classical secret sharing scheme $(\mathsf{Share}, \mathsf{Reconstruct})$, a partial set of shares $\{\mathsf{csh}_i\}_{i \in A \subseteq [n]}$, and an index $j \notin S$, we let $\mathsf{Share}_j(s, \{\mathsf{csh}_i\}_{i \in A})$ denote the distribution over the jth share conditioned on the secret s and the set of shares $\{\mathsf{csh}_i\}_{i \in A}$. If the set of already determined shares are not consistent with the secret s, then the above distribution outputs $\perp$. Similarly, for a subset of indices, $D \subset [n]$, we let $\mathsf{Share}_D(s, \{\mathsf{csh}_i\}_{i \in A \subseteq [n]})$ denote the distribution over the set of shares $\{\mathsf{csh}_i\}_{i \in D}$ conditioned on the secret s and the shares $\{\mathsf{csh}_i\}_{i \in A}$.

We also define the following binary projective measurement, which is parameterized by a proof of deletion cert. Parsing a deletion certificate as $\mathsf{cert} := x_{c_1}, ..., x_{c_m}$, where $c_k \in \{0, 1\}$, we define the following projector:

$$\Pi_{\mathsf{cert}} := \bigotimes_{k \in [m]} H|c_k\rangle\langle c_k|H.$$

The measurement outcome above corresponds to measuring a register C in the Hadamard basis, and then observing $c_1...c_m$ as the measurement outcome.

We now introduce two experiments $\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s)$ and $\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{rand}}(s)$, shown in Figure 1. The first of these denotes experiment $\mathsf{SS\text{-}ACD}$ instantiated with Construction 1 as the secret-sharing scheme, and the second only differs in that the underlying classical shares are replaced with uniform strings. We show that the outputs of these two experiments are indistinguishable.

$\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s)$ $\boxed{\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{rand}}(s)}$

- $\mathsf{lock} \leftarrow \{0, 1\}^m$
- $\mathsf{csh}_1, ..., \mathsf{csh}_n \leftarrow \mathsf{Share}(\mathsf{lock})$ $\boxed{\mathsf{csh}_1, ...\mathsf{csh}_n \leftarrow \{0, 1\}^m}$
- Sample uniform $x_0^{i,k}, x_1^{i,k} \leftarrow \{0, 1\}^\kappa$, hardcode the preimages in Rec according to Construction 1, and generate the obfuscated program $\widetilde{\mathsf{Rec}} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{CC}[\mathsf{Rec}, \mathsf{lock}, s])$.
- For $i \in [n]$ prepare the quantum state

$$|\mathsf{qsh}_i\rangle = \bigotimes_{k \in [m]} \left(|x_0^{i,k}\rangle + (-1)^{\mathsf{csh}_{i,k}}|x_1^{i,k}\rangle\right)$$

- Initialize each share register Sh_i with the state $|\widetilde{\mathsf{Rec}}\rangle \otimes |\mathsf{qsh}_i\rangle$.
- Run and output the result of the experiment $\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s)$ using the above share registers.

Fig. 1: Experiments in the proof.

Lemma 2. *For any secret s,*

$$\mathsf{TD}\left(\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s), \mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{rand}}(s)\right) \leq \mathsf{negl}(\lambda).$$

We first introduce the following hybrids.

- $\mathsf{Hyb}'_0(s)$: This is the same as $\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s)$ except we lazy sample the underlying classical shares as the adversary corrupts them.
  - Sample $\mathsf{lock} \leftarrow \{0,1\}^m$. Sample uniform values $x_0^{i,k}, x_1^{i,k} \leftarrow \{0,1\}^\kappa$ for $i \in [n], k \in [m], b \in \{0,1\}$. Set $y_b^{i,k} = f(x_b^{i,k})$ for $b \in \{0,1\}, i \in [n], k \in [m]$, and set the verification key $\mathsf{vk} = \{y_0^{i,k}, y_1^{i,k}\}_{i\in[n],k\in[m]}$. Generate the obfuscated program $\widetilde{\mathsf{Rec}} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{CC}[\mathsf{Rec}, \mathsf{lock}, s])$.
  - Run the $\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s)$ experiment as follows. Initialize $\mathcal{A}$ with $|\psi\rangle \otimes |\widetilde{\mathsf{Rec}}\rangle$, and initialize the set of corrupted and deleted shares C and D as empty. When $\mathcal{A}$ corrupts a share c, generate the classical share as $\mathsf{csh}_c \leftarrow \mathsf{Share}_c(s, \{\mathsf{csh}_i\}_{i\in C})$, and prepare the following corresponding quantum encoding on register Sh_c:

$$|\mathsf{qsh}_c\rangle_{\mathsf{Sh}_c} := \bigotimes_{k\in m} \left(|x_0^{c,k}\rangle + (-1)^{\mathsf{csh}_{c,k}}|x_1^{c,k}\rangle\right).$$

  Then add c to the set of corrupted share indices C.
- $\mathsf{Hyb}_0(s)$: In this hybrid we purify the classical share generation by introducing a set of registers $\mathsf{C}_1, ..., \mathsf{C}_n$ held by the challenger which will hold superpositions of classical shares. The share registers $\mathsf{Sh}_1, ..., \mathsf{Sh}_n$ will then be generated based on the states on the challengers registers.
  - Sample $\mathsf{lock} \leftarrow \{0,1\}^m$. Sample uniform values $x_b^{i,k} \leftarrow \{0,1\}^\kappa$ for $i \in [n], k \in [m], b \in \{0,1\}$. Set $y_b^{i,k} = f(x_b^{i,k})$ for $b \in \{0,1\}, i \in [n], k \in [m]$, and set the verification key $\mathsf{vk} = \{y_0^{i,k}, y_1^{i,k}\}_{i\in[n],k\in[m]}$. Generate the obfuscated program $\widetilde{\mathsf{Rec}} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{CC}[\mathsf{Rec}, \mathsf{lock}, s])$.
  - Whenever a new share c is corrupted, prepare a state on registers C_c and Sh_c as follows. Run the procedure $\mathsf{Share}_c(s, \{\mathsf{C}_i\}_{i\in C})$ coherently on the superposition of sets of shares defined by the challengers registers to obtain

$$\mathsf{C}_c \leftarrow \mathsf{Share}_j(s, \{\mathsf{C}_i\}_{i\in C}).$$

  Let $\sum_{\mathsf{csh}_c\in\{0,1\}^m} \alpha_{\mathsf{csh}_c}|\mathsf{csh}_c\rangle_{\mathsf{C}_c}$ be the state on register C_c. Prepare the following state by running the quantum share encoding procedure coherently on C_c:

$$\frac{1}{2^{m/2}} \sum_{\mathsf{csh}_c} \alpha_{\mathsf{csh}_c}|\mathsf{csh}_j\rangle_{\mathsf{C}_c} \bigotimes_{k\in[m]} \left(|x_0^{c,k}\rangle + (-1)^{\mathsf{csh}_{c,k}}|x_1^{c,k}\rangle\right)_{\mathsf{Sh}_c}.$$

  Add c to C.
  - Measure each C_i in the computational basis, and then output the result of $\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s)$.

- $\mathsf{Hyb}_i(s)$ for $i \in [n]$: Run Hyb_0 with the following exception. For the first i deletions, after each share j is deleted, measure register C_j with respect to the binary projective measurement $\{\Pi_{\mathsf{cert}_j}, \mathbb{1} - \Pi_{\mathsf{cert}_j}\}$. If the measurement result is "reject" (i.e. has measurement outcome $\mathbb{1} - \Pi_{\mathsf{cert}_j}$), output $\perp$ and abort the experiment.
- $\mathsf{Sim}_i(s)$ for $i \in [0, n]$: Run $\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s)$ as follows.
  - When $\mathcal{A}$ corrupts a share c, prepare the following state on registers C_c and Sh_c:
$$\frac{1}{2^{m/2}} \sum_{\mathsf{csh}_c \in \{0,1\}^m} |\mathsf{csh}_c\rangle_{\mathsf{C}_c} \bigotimes_{k \in [m]} \left(|x_0^{c,k}\rangle + (-1)^{\mathsf{csh}_{c,k}} |x_1^{c,k}\rangle\right)_{\mathsf{Sh}_c}$$
  - For the first i deletions, $d_1, ..., d_i$, after the challenger verifies cert_{d_j} (for $j \in [i]$), perform the binary projective measurement $\{\Pi_{\mathsf{cert}_{d_j}}, \mathbb{1} - \Pi_{\mathsf{cert}_{d_j}}\}$. Abort and output $\perp$ immediately after any measurement that rejects (i.e. has measurement outcome $\mathbb{1} - \Pi_{\mathsf{cert}}$).

Claim 1. For every secret s,
$$\mathsf{TD}\left(\mathsf{Expt}_{\mathsf{real}}^{\mathsf{SS\text{-}ACD}}(s), \mathsf{Hyb}_0(s)\right) = 0.$$

Proof. First, the fact that $\mathsf{TD}(\mathsf{Expt}_{\mathsf{real}}^{\mathsf{SS\text{-}ACD}}(s), \mathsf{Hyb}'_0(s)) = 0$ follows from the definition of the lazy-sampling style sharing procedure used in Hyb'_0. The fact that $\mathsf{TD}(\mathsf{Hyb}'_0(s), \mathsf{Hyb}_0(s)) = 0$ follows from the fact that operations on disjoint sets of registers commute, and in particular measuring the challengers registers at the beginning or at the end of the experiment will not impact the state of the adversary $\mathcal{A}$. Since measuring the challengers registers in the computational basis before giving the share registers to $\mathcal{A}$ induces the same distribution over classical shares as in $\mathsf{Hyb}'_0(s)$, the result follows. □

Recall that in $\mathsf{Hyb}_i(s)$, the state of the challengers share registers that have been deleted (up to the i'th deletion) are measured in the Hadamard basis and are therefore in a uniform superposition immediately after the deletion takes place (assuming the experiment does not abort). Intuitively, if the share registers that are deleted contain the same distribution (uniform) as they did prior to being queried, we might hope that when generating newly corrupted shares we can ignore the shares that have been deleted and condition only on the shares in $C \setminus D$. This intuition is proved formally in the following claim.

Define $\mathsf{Hyb}_i^T(s)$ (resp. $\mathsf{Sim}_i^T(s)$) as the experiment which runs $\mathsf{Hyb}_i(s)$ (resp. $\mathsf{Sim}_i(s)$) but aborts immediately after the ith deletion and outputs the adversaries register State.

Claim 2. For every secret s and for every $i \in [n]$,
$$\mathsf{TD}\left(\mathsf{Hyb}_i^T(s), \mathsf{Sim}_i^T(s)\right) \le \mathsf{negl}(\lambda).$$

Proof. Recall that the only difference between $\mathsf{Hyb}_i^T(s)$ and $\mathsf{Sim}_i^T(s)$ is that in the former experiment shares are generated based on previously corrupted shares, and in the latter they are generated as uniform superposition states. We will prove the claim by induction on i. To see that the claim holds for $i = 0$, note that prior to the first deletion, it follows from the property of the classical secret sharing scheme that any unauthorized set of shares (and in particular whichever subset is queried by the adversary prior to the first deletion) is perfectly indistinguishable from uniformly random strings. Therefore in Hyb_0, each share is a uniform superposition, and $\mathsf{TD}(\mathsf{Hyb}_0^T(s), \mathsf{Sim}_0^T(s)) = 0$.

Now suppose that $\mathsf{TD}(\mathsf{Hyb}_i^T(s), \mathsf{Sim}_i^T(s)) \leq \mathsf{negl}(\lambda)$. We show that the claim holds for $i+1$ by introducing the following sequence of hybrids.

– Hyb_{i+1}^T : Run Hyb_{i+1} but abort and output the adversaries register State as soon as the $(i+1)$th deletion test has been passed.
– $\mathsf{Hyb}_{i+1}'^T$: Run Hyb_{i+1}^T, up until $\mathcal{A}$ outputs its ith deletion. Then, for each subsequent corruption c, prepare the following state on registers C_c and Sh_c which corresponds to encoding a uniform classical share:
$$\frac{1}{2^{m/2}} \sum_{\mathsf{csh}_c \in \{0,1\}^m} |\mathsf{csh}_c\rangle_{\mathsf{C}_c} \bigotimes_{k \in [m]} \left(|x_0^{c,k}\rangle + (-1)^{\mathsf{csh}_{c,k}} |x_1^{c,k}\rangle\right)_{\mathsf{Sh}_c}.$$
Once $\mathcal{A}$ outputs its $(i+1)$th deletion certificate, abort and output the adversary's register State.
– Sim_{i+1}^T : Run Sim_{i+1} but abort and output the adversaries register State as soon as the $(i+1)$th deletion test has been passed.

We first show that
$$\mathsf{TD}(\mathsf{Hyb}_{i+1}'^T, \mathsf{Sim}_{i+1}^T) \leq \mathsf{negl}(\lambda).$$
Note that after the ith deletion the experiments $\mathsf{Hyb}_{i+1}'^T$ and Sim_{i+1}^T are identical. Therefore the trace distance between these two experiments is upper bounded by their distance immediately prior to the ith deletion. By the inductive hypothesis, $\mathsf{TD}(\mathsf{Hyb}_i^T(s), \mathsf{Sim}_i^T(s)) \leq \mathsf{negl}(\lambda)$. Since $\mathsf{Hyb}_{i+1}'^T$ is identical to Hyb_i up until the ith deletion (but prior to the Hadamard measurement on the deleted share), it follows that the state of $\mathsf{Hyb}_{i+1}'^T$ prior to the ith deletion is negligibly close to that of Sim_{i+1}^T prior to the ith deletion, and therefore we have the desired result.

We now show that
$$\mathsf{TD}\left(\mathsf{Hyb}_{i+1}^T, \mathsf{Hyb}_{i+1}'^T\right) \leq \mathsf{negl}(\lambda).$$
To show the above, we will prove that each corrupted share generated after the ith deletion but before the $(i+1)$th deletion in Hyb_{i+1}^T is in a uniform superposition. To do so, we will argue that ignoring the deleted share registers $\{\mathsf{C}_i\}_{i \in D}$ and generating each newly corrupted share based only on the shares in $C \setminus D$ does not change the outcome of the experiment. Since $C \setminus D$ is never authorized, it follows from the uniformity property of the classical secret sharing scheme that generating each newly corrupted share based on $C \setminus D$ results in a uniform superposition.

We introduce the following sequence of hybrids which give different ways of generating the shares corrupted after the ith deletion in Hyb^T_{i+1}.

- Expt_0: Run Hyb^T_{i+1} with no changes. In particular, each newly corrupted is generated as follows based on all shares in C, including those that have been deleted:
$$\mathsf{C}_c \leftarrow \mathsf{Share}_c(s, \{\mathsf{C}_i\}_{i\in C}).$$
- Expt_1: Run Hyb^T_{i+1} but generate each share after the ith deletion as follows:
  - Generate fresh share registers for the deleted shares based on the shares in $C \setminus D$:
  $$\{\mathsf{C}'_i\}_{i\in D} \leftarrow \mathsf{Share}_D(s, \{\mathsf{C}_i\}_{i\in C\setminus D}).$$
  - Generate each newly corrupted share based on the shares in $C \setminus D$ together with the freshly generated share registers $\{\mathsf{C}'_i\}_{i\in D}$:
  $$\mathsf{C}_c \leftarrow \mathsf{Share}_c(s, \{\mathsf{C}_i\}_{i\in C\setminus D} \cup \{\mathsf{C}'_i\}_{i\in D}).$$
- Expt_2: Run Hyb^T_{i+1} but generate each newly corrupted share as follows based only on shares in $C \setminus D$:
$$\mathsf{C}_c \leftarrow \mathsf{Share}_c(s, \{\mathsf{C}_i\}_{i\in C\setminus D}).$$

The only difference between Expt_1 and Expt_2 is that in Expt_2, additional share registers for the indices in D are generated before generating C_j. Since random variables in a joint distribution can be sampled in any order as a sequence of samples from conditional distributions, it is clear that $\mathsf{SD}(\mathsf{Expt}_1, \mathsf{Expt}_2) = 0$.

To prove that Expt_0 and Expt_1 are identical, note that the only difference between these experiments is that each newly corrupted share C_j is generated based on the original deleted share registers $\{\mathsf{C}_i\}_{i\in D}$ in Expt_0, and based on the freshly generated registers $\{\mathsf{C}'_i\}_{i\in D}$ in the case of Expt_1. Since the distribution $\mathsf{Share}_j(\cdot)$ takes classical inputs and is being run coherently on superpositions, it is enough to show that a computational basis measurement of the original registers $\{\mathsf{C}_i\}_{i\in D}$ and the new registers $\{\mathsf{C}'_i\}_{i\in D}$ induce the same distribution. This follows from the fact that each deleted share register C_d is in a Hadamard basis state immediately after being deleted. However by the uniformity property of the underlying classical secret-sharing scheme, if we were to regenerate C_d based on the shares in $C \setminus D$ we would also obtain a uniform superposition.

Therefore $\mathsf{TD}(\mathsf{Hyb}^T_{i+1}(s), \mathsf{Expt}_2) = 0$. However note that in Expt_2, each corrupted share C_j is generated based on a set $C \setminus D$ such that $(C \setminus D) \cup \{j\}$ is not authorized (for otherwise the adversary would obtain an authorized set). Therefore by the uniformity property of the underlying secret sharing scheme, the newly corrupted share registers in Expt_2 contain uniform superpositions. It follows that $\mathsf{TD}(\mathsf{Hyb}^T_{i+}, \mathsf{Hyb}'^T_{i+1}) \leq \mathsf{negl}(\lambda)$ as desired which completes the proof. $\square$

We show that each Hadamard measurement on the deleted registers impacts the state of the experiment by at most a negligible amount.

Claim 3. For every $i \in [0, n]$ and every secret s,

$$\mathsf{TD}(\mathsf{Hyb}_i(s), \mathsf{Hyb}_{i+1}(s)) \leq \mathsf{negl}(\lambda).$$

Proof. The only difference between $\mathsf{Hyb}_i(s)$ and $\mathsf{Hyb}_{i+1}(s)$ is a Hadamard measurement on register $\mathsf{C}_{d_{i+1}}$ in Hyb_{i+1}, where d_{i+1} is the index of the $(i+1)$th share that is deleted. Suppose that the Hadamard measurement rejects with probability at most ϵ. It follows from the Gentle Measurement Lemma that, conditioned on the Hadamard measurement accepting, the trace distance between Hyb_i and Hyb_{i+1} is at most $2\sqrt{\epsilon}$. It follows that

$$\mathsf{TD}(\mathsf{Hyb}_i(s), \mathsf{Hyb}_{i+1}(s)) \leq (1-\epsilon)2\sqrt{\epsilon} + \epsilon.$$

Therefore to prove the claim we will show that the probability that the Hadamard measurement rejects is negligible. We start by observing that the probability of acceptance is almost identical in each of the following hybrids.

- Hyb_{i+1}: Run the identically named hybrid defined at the start of the proof.
- Hyb^T_{i+1}: Run Hyb_{i+1} but abort and output State after the $(i+1)$th deletion.
- Sim^T_{i+1}: Run Sim_{i+1} but abort and output State after the $(i+1)$th deletion.

Since $\mathsf{Hyb}_{i+1}(s)$ and $\mathsf{Hyb}^T_{i+1}(s)$ are identical up to the round where the ith Hadamard test is applied, the acceptance probability is identical in both cases. By Claim 2, we have

$$\mathsf{TD}(\mathsf{Hyb}_{i+1}(s), \mathsf{Sim}_{i+1}(s)) \leq \mathsf{negl}(\lambda).$$

Therefore it suffices to show that the probability that the final deletion test in Sim^T_{i+1} does not pass is negligible.

Recall that the projective measurement Π_{cert} simply measures the classical share register C in the Hadamard basis to obtain a string $c_1...c_m$ and checks if the deletion proof $\mathsf{cert} := x_{b_1}, ..., x_{b_m}$ that was just output by $\mathcal{A}$ is such that $b_k = c_k$ for all $k \in [m]$. Since measurements on disjoint registers commute perfectly, we can instead measure C in the Hadamard basis at the start of the experiment to obtain a string $c_1...c_m$ and then run the experiment until $\mathcal{A}$ deletes the corresponding share and outputs a proof $x_{b_1}...x_{b_m}$. Since the measurements commute, the probability that $c_k = b_k$ for all $i \in [m]$ is identical in each case. With this in mind, we define the following experiment which is essentially identical to Sim_{i+1} except that we perform the Hadamard measurement on C before running the adversary as described above.

Fix some share index $d \in [n]$, and suppose that d has a non-negligible chance of being deleted in the $(i+1)$th round. We will show that conditioned on d being deleted, the Hadamard test passes with high probability. Suppose otherwise. Then $\mathsf{Expt}_0(d)$ given below must output 1 with non-negligible probability[3].

[3] $\mathsf{Expt}_0(d)$ is defined by ignoring any text inside a box, and $\mathsf{Expt}_1(d)$ is defined by running $\mathsf{Expt}_0(d)$, but ignoring text outside a box on lines which contain both boxed and unboxed text.

– $\mathsf{Expt}_0(d)$ $\boxed{\mathsf{Expt}_1(d)}$
  - Sample $\mathsf{lock} \leftarrow \{0,1\}^m$ and uniform values $x_0^{i,k}, x_1^{i,k} \leftarrow \{0,1\}^\kappa$ for $i \in [n], k \in [m]$. Set $y_b^{i,k} = f(x_b^{i,k})$.
  - Instantiate Rec with $\{x_0^{i,k} \oplus x_1^{i,k}\}$ as in Construction 1.
  - $\widetilde{\mathsf{Rec}}_{\mathsf{real}} \leftarrow \mathsf{CC.Obf}\left(1^\lambda, \mathsf{CC}[\mathsf{Rec}, \mathsf{lock}, s]\right)$ $\boxed{\widetilde{\mathsf{Rec}}_{\mathsf{sim}} \leftarrow \mathsf{Sim}(1^\lambda, \mathsf{Rec.param})}$
  - Proceed as in Sim_{i+1} but with the following exception. If $\mathcal{A}$ corrupts the dth share, prepare the state

$$\sum_{\mathsf{csh}} |\mathsf{csh}\rangle_{\mathsf{C}_d} \bigotimes_{k \in [m]} \left(|x_0^{d,k}\rangle + (-1)^{\mathsf{csh}_{d,k}} |x_1^{d,k}\rangle\right)_{\mathsf{Sh}_d}$$

    on registers Sh_d and C_d, and measure C_d in the Hadamard basis to obtain measurement outcome $c_1, ..., c_m$. Note that the residual state on register Sh_d is given by $\bigotimes_{k\in[m]} |x_{c_k}^{d,k}\rangle$.
  - Run Sim_{i+1}, sampling the shares uniformly, up until $\mathcal{A}$ outputs the $(i+1)$th proof of deletion $\mathsf{cert} := (x_{b_1}, ..., x_{b_m})$.
  - If the $(i+1)$th proof of deletion is not for share d, then abort and output $\bot$.
  - If $b_k \neq c_k$ for some $k \in [m]$, output 1, and otherwise output $\bot$.

We first show that

$$\Pr[\mathsf{Expt}_0(d) \text{ outputs } 1] \neq \mathsf{negl}(\lambda) \implies \Pr[\mathsf{Expt}_1(d) \text{ outputs } 1] \neq \mathsf{negl}(\lambda).$$

If the above does not hold, we can construct a distinguisher violating security of the compute-and-compare obfuscator. We present our distinguisher $\mathcal{B}_{\mathsf{CC}}$ below. It takes as input either a simulated program $\widetilde{\mathsf{Rec}}_{\mathsf{sim}} \leftarrow \mathsf{Sim}(1^\lambda, \mathsf{Rec.param})$, or an obfuscated program $\widetilde{\mathsf{Rec}}_{\mathsf{real}} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{CC}[\mathsf{Rec}, \mathsf{lock}, s])$, as well as the preimages $\{x_0^{i,k}, x_1^{i,k}\}$ as auxiliary input.

$$\underline{\mathcal{B}_{\mathsf{CC}}\left(\widetilde{\mathsf{Rec}}, \{x_0^{i,k}, x_1^{i,k}\}_{i\in[n],k\in[m]}\right)}$$

– Hardcode the index d.
– Run the adversary $\mathcal{A}$ initialized with $\widetilde{\mathsf{Rec}}$ and answer the corruption requests as follows. If $\mathcal{A}$ corrupts share q, do the following:
  - If $q \neq d$, prepare a uniform classical share $\mathsf{csh}_q \leftarrow \{0,1\}^m$ and encode it with the appropriate preimages on register Sh_i.
  - If $q = d$, prepare the state $\bigotimes_{i\in[m]} |x_{c_i}\rangle$ on the share register Sh_d.
– If $\mathcal{A}$ outputs a valid proof of deletion $\mathsf{cert}_d := x_{b_1}, ..., x_{b_m}$ for Sh_d as its $(i+1)$th deletion, do the following:
  - If $c_k \neq b_k$ for some $k \in [m]$, output real.
  - Otherwise, output either sim or real with equal probability.
– If $\mathcal{A}$ does not output a proof of deletion for Sh_d, then output either sim or real with equal probability.

Suppose the implication does not hold, and that $\Pr[\mathsf{Expt}_0(d) \text{ outputs } 1]$ is non-negligible but $\Pr[\mathsf{Expt}_1(d) \text{ outputs } 1] \leq \mathsf{negl}(\lambda)$. Let Good denote the event in the execution of $\mathcal{B}_{\mathsf{CC}}$ that $\mathcal{A}$ deletes the dth share as its $(i+1)$th deletion but that $b_k \neq c_k$ for some $k \in [m]$ (note that this is equivalent to $\mathsf{Expt}_0(i_0)/\mathsf{Expt}_1(i_0)$ outputting 1). We abuse notation and write $\widetilde{\mathsf{Rec}} \leftarrow \mathsf{Expt}_b(d)$ to mean that $\widetilde{\mathsf{Rec}}$ is generated according to the first three lines of $\mathsf{Expt}_b(d)$ for $b \in \{0,1\}$. Since $\mathcal{B}_{\mathsf{CC}}$ outputs real when an event in Good occurs, and outputs a random guess otherwise, we have the following:

$$\begin{aligned}&\left|\Pr_{\widetilde{\mathsf{Rec}} \leftarrow \mathsf{Expt}_0(d)}[\mathcal{B}_{\mathsf{CC}} \text{ outputs } \mathsf{real}] - \Pr_{\widetilde{\mathsf{Rec}} \leftarrow \mathsf{Expt}_1(d)}[\mathcal{B}_{\mathsf{CC}} \text{ outputs } \mathsf{real}]\right| \\ &= \frac{1}{2}\Big|\Pr[\mathsf{Expt}_0(d) \text{ outputs } 1] - \Pr[\mathsf{Expt}_1(d) \text{ outputs } 1]\Big|\end{aligned}$$

Since the first term above is assumed to be non-negligible, and the second is assumed to be negligible, it follows that $\mathcal{B}_{\mathsf{CC}}$ has non-negligible advantage against the compute-and-compare obfuscator.

We now show that $\Pr[\mathsf{Expt}_1(d) \text{ outputs } 1] \leq \mathsf{negl}(\lambda)$. If this is not the case, than we can construct an adversary $\mathcal{B}_{\mathsf{OWF}}$ against the one-way function which we present below.

$\underline{\mathcal{B}_{\mathsf{OWF}}(y)}$

- Hardcode the index d, and sample uniform index $k_0 \leftarrow [m]$, and uniform $b_0 \leftarrow \{0,1\}$.
- Set $y_{b_0}^{d,k_0} := y$. For $(i,k,b) \neq (d,k_0,b_0)$, sample uniform $x_b^{i,k} \leftarrow \{0,1\}^\kappa$ and set $y_b^{i,k} = f(x_b^{i,k})$.
- Sample $\widetilde{\mathsf{Rec}}_{\mathsf{sim}} \leftarrow \mathsf{Sim}(1^\lambda, \mathsf{Rec.param})$
- Simulate $\mathsf{Expt}_1(d)$ as follows. Initialize $\mathcal{A}$ with $|\psi\rangle \otimes |\widetilde{\mathsf{Rec}}_{\mathsf{sim}}\rangle$. When $\mathcal{A}$ corrupts some share q, do the following:
  - If $q \neq d$, sample a uniform classical string $\mathsf{csh} \leftarrow \{0,1\}^m$, and encode it with the corresponding preimages.
  - If $q = d$, prepare the state
$$\bigotimes_{k \neq k_0}\left(|x_0^{d,k}\rangle + (-1)^{\mathsf{csh}_{d,k}}|x_1\rangle^{d,k}\right)\bigotimes|x_{1-b}^{d,k_0}\rangle$$
- If $\mathcal{A}$ outputs as part of a certificate of deletion, a preimage of y, then output y. Otherwise output $\perp$.

If y is the evaluation of a uniform preimage, $\mathcal{B}_{\mathsf{OWF}}$ perfectly simulates $\mathsf{Expt}_1(d)$. If $\mathsf{Expt}_1(d)$ has a non-negligible chance of outputting 1, then the above procedure has a non-negligible chance of inverting the one-way function. □

Proof (of Lemma 2). With the above claims in hand the main result easily follows. In more detail, note that $\mathsf{Sim}_n(s)$ is identical to $\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{rand}}(s)$, and therefore

$\mathsf{TD}\left(\mathsf{Sim}_n(s), \mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{rand}}(s)\right) = 0$. Claim 1 implies $\mathsf{TD}(\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s), \mathsf{Hyb}'_0(s)) \leq \mathsf{negl}(\lambda)$. Claim 2 implies $\mathsf{TD}(\mathsf{Hyb}_0(s), \mathsf{Hyb}_n(s)) \leq \mathsf{negl}(\lambda)$, and Claim 3 implies that $\mathsf{TD}(\mathsf{Hyb}_n(s), \mathsf{Sim}_n(s)) \leq \mathsf{negl}(\lambda)$. Putting the above together we obtain the lemma statement. □

We now prove the security of Construction 1.

Theorem 1. *Let $\mathsf{SS}_{\mathsf{classical}}$ be a classical secret sharing scheme such that any unauthorized set of shares is perfectly indistinguishable from uniform. Then instantiating 1 with $\mathsf{SS}_{\mathsf{classical}}$ gives a scheme that has computational adaptive PVD.*

Proof. Let $\mathsf{SS}_{\mathsf{PVD}}$ be the secret sharing scheme that results from Construction 1. We wish to show that for any QPT adversary $\mathcal{A}$ and any two secrets s_0, s_1,

$$\mathsf{SS\text{-}ACD}(1^\lambda, |\psi\rangle, \mathcal{A}, s_0) \approx_c \mathsf{SS\text{-}ACD}(1^\lambda, |\psi\rangle, \mathcal{A}, s_1).$$

Note that for $b \in \{0,1\}$, $\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s_b)$ is identical to $\mathsf{SS\text{-}ACD}(1^\lambda, |\psi\rangle, \mathcal{A}, s_b)$ instantiated with Construction 1. It follows from Lemma 2 that for $b \in \{0,1\}$,

$$\mathsf{TD}\left(\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s_b), \mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{rand}}(s_b)\right) \leq \mathsf{negl}(\lambda).$$

Therefore, it suffices to show that $\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{rand}}(s_0) \approx \mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{rand}}(s_1)$. We prove the claim by appealing to the security of the compute-and-compare obfuscator.

Consider an adversary $\mathcal{B}_{\mathsf{CC}}$ against the compute-and-compare obfuscator which receives uniform preimages $\{x_0^{i,k}, x_1^{i,k}\}_{i\in[n],k\in[m]}$, and an obfuscated program $\widetilde{\mathsf{Rec}} \leftarrow \mathsf{CC.Obf}(\mathsf{CC}[\mathsf{Rec}, \mathsf{lock}, s_b])$ where lock is uniform. Clearly lock is unpredictable, even given Rec in the clear together with the preimages. Therefore it follows from the security of the compute-and-compare obfuscator that there exists a simulator $\mathsf{Sim}(1^\lambda, \mathsf{Rec.param})$ such that

$$\left(\widetilde{\mathsf{Rec}}_{\mathsf{real}}, \{x_0^{i,k}, x_1^{i,k}\}_{i\in[n],k\in[m]}\right) \approx_c \left(\widetilde{\mathsf{Rec}}_{\mathsf{sim}}, \{x_0^{i,k}, x_1^{i,k}\}_{i\in[n],k\in[m]}\right),$$

where the preimages above are uniform and $\widetilde{\mathsf{Rec}}_{\mathsf{sim}} \leftarrow \mathsf{Sim}(1^\lambda, \mathsf{Rec.param})$. Therefore, replacing the obfuscated program $\widetilde{\mathsf{Rec}}_{\mathsf{real}}$ in the experiment $\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{rand}}(s_b)$ with a simulated program $\widetilde{\mathsf{Rec}}_{\mathsf{sim}}$ is undetectable to a computationally bounded adversary. However since this modified experiment no longer depends on the hidden value s_b, the desired claim follows. □

4.2 Certified Everlasting Security

We now present our construction which takes an arbitrary computational secret sharing scheme with adaptive PVD and upgrades it to have everlasting security.

Construction 2 *Let $\mathsf{SS}_{\mathsf{comp}}$ be a computational secret sharing scheme with adaptive publicly verifiable deletion (definition 8), and let $\mathsf{SS}_{\mathsf{classical}}$ be a classical secret sharing scheme for the same access structure with the uniformity property. Let $f : \{0,1\}^\kappa \mapsto \{0,1\}^\beta$ be a one-wway function, and let $F : \mathcal{K} \times \{0,1\}^{\lceil 1+\log(n\cdot m)\rceil} \mapsto \{0,1\}^\kappa$ be a pseudorandom function family.*

– $\mathsf{Share}_{\mathbb{A}}(1^\lambda, s)$: *On input a secret s, sample uniform PRF key $k_0 \leftarrow \mathcal{K}$, and set $x_b^{i,k}, x_1^{i,k} := F(k_0, b||i||k)$ for all $i \in [n], k \in [m], b \in \{0,1\}$. Set $y_b^{i,k} = f(x_b^{i,k})$. Generate the following quantum shares and verification key for the PRF key:*

$$\mathsf{vk}^{\mathsf{PRF}}, \{|\mathsf{qsh}_i^{\mathsf{PRF}}\rangle\}_{i\in[n]} \leftarrow \mathsf{SS}_{\mathsf{comp}}.\mathsf{Share}\left(k_0\right).$$

Generate classical shares $\{\mathsf{csh}_i\}_{i\in[n]} \leftarrow \mathsf{SS}_{\mathsf{classical}}.\mathsf{Share}(s)$. For each $i \in [n]$ do the following:

- *Let $\mathsf{csh}_{i,k}$, be the kth bit of csh_i and prepare the quantum state*

$$|\mathsf{qsh}_i^s\rangle = \bigotimes_{k\in[m]} \left(|x_0^{i,k}\rangle + (-1)^{\mathsf{csh}_{i,k}}|x_1^{i,k}\rangle\right).$$

- *Set $|\mathsf{qsh}_i\rangle := |\mathsf{qsh}_i^s\rangle|\mathsf{qsh}_i^{\mathsf{PRF}}\rangle$.*

Set the verification key as $\mathsf{vk} = \left(\mathsf{vk}^{\mathsf{PRF}}, \{y_0^{(i,k)}, y_1^{i,k}\}_{i\in[n],k\in[m]}\right)$.

– $\mathsf{Reconstruct}_{\mathbb{A}}(\{|\mathsf{qsh}_i\rangle\}_{i\in A})$: *Parse each share as $|\mathsf{qsh}_i\rangle = |\mathsf{qsh}_i^s\rangle|\mathsf{qsh}_i^{\mathsf{PRF}}\rangle$. Reconstruct the PRF key as*

$$k_0 \leftarrow \mathsf{SS}_{\mathsf{comp}}.\mathsf{Reconstruct}(\{|\mathsf{qsh}_i^{\mathsf{PRF}}\rangle\}_{i\in A}).$$

Use the PRF key to reconstruct the xor'ed preimages $\{x_0^{i,k} \oplus x_1^{i,k}\}$. Use the preimages together with the quantum shares $\{|\mathsf{qsh}_i^s\rangle\}_{i\in A}$ to obtain classical shares $\{\mathsf{csh}_i\}_{i\in A}$. Output the secret $s \leftarrow \mathsf{SS}_{\mathsf{classical}}.\mathsf{Reconstruct})(\{\mathsf{csh}_i\}_{i\in A})$.

– $\mathsf{Delete}_{\mathbb{A}}(|\mathsf{qsh}\rangle)$: *Parse $|\mathsf{qsh}\rangle$ as $|\mathsf{qsh}^s\rangle \otimes |\mathsf{qsh}^{\mathsf{PRF}}\rangle$, and run $\mathsf{SS}_{\mathsf{comp}}.\mathsf{Delete}(|\mathsf{qsh}_i^{\mathsf{PRF}}\rangle)$ to obtain $\mathsf{cert}^{\mathsf{PRF}}$. Measure the state $|\mathsf{qsh}_i^s\rangle$ in the computational basis to obtain cert^s. Output $(\mathsf{cert}^s, \mathsf{cert}^{\mathsf{PRF}})$ as the proof of deletion.*

– $\mathsf{Verify}_{\mathbb{A}}(\mathsf{vk}, i, \mathsf{cert})$: *On input an index i, a proof $\mathsf{cert} := (\mathsf{cert}^s, \mathsf{cert}^{\mathsf{PRF}})$ and a verification key $\mathsf{vk} = \left(\mathsf{vk}^{\mathsf{PRF}}, \{(y_0^{i,k}, y_1^{i,k})\}_{i\in[n],k\in[m]}\right)$, parse cert^s as $x_1, ..., x_m \in \{0,1\}^\kappa$ and check that $f(x_k) \in \{y_0^{i,k}, y_1^{i,k}\}$ for all $k \in [m]$. If the above condition is satisfied and $\mathsf{SS}_{\mathsf{comp}}.\mathsf{Delete}(\mathsf{vk}^{\mathsf{PRF}}, i, \mathsf{cert}^{\mathsf{PRF}}) = \top$, then return $\top$, and otherwise return $\bot$.*

Lemma 3. *Construction 2 has adaptive PVD with everlasting security.*

The proof is almost identical to that of Theorem 1, except where we appeal to the security of the computational secret sharing scheme rather than the compute-and-compare obfuscator in order to hide the PRF key. For completeness we include a full proof in Appendix A

5 Construction from One-Way Functions

We now present our secret sharing construction based only on the existence of a post-quantum one-way function.

Construction 3 *Let $f : \{0,1\}^\kappa \mapsto \{0,1\}^\beta$ be a one-way function, and let $\mathsf{SS}_{\mathsf{classical}}$ be a classical secret sharing scheme such that any unauthorized set of shares is indistinguishable from uniform. Let $F : \mathcal{K} \times \{0,1\}^{\lceil 1+\log(n\cdot m)\rceil} \mapsto \{0,1\}^\kappa$ be a pseudorandom function family.*

- $\mathsf{Share}_{\mathbb{A}}(1^\lambda, s)$:
  - *Compute* $\{\mathsf{csh}_i^{-1}\}_{i\in[n]} \leftarrow \mathsf{SS}_{\mathsf{classical}}.\mathsf{Share}(s)$. *Sample* $k^0_{\mathsf{prf}}, \dots, k^n_{\mathsf{prf}} \leftarrow \mathcal{K}$, *and create shares* $\{\mathsf{csh}_i^\ell\} \leftarrow \mathsf{Share}(k^\ell_{\mathsf{prf}})$ *for each* $\ell \in \{0, ..., n-1\}$.
  - *For* $\ell = 0, ..., n$ *do the following:*
    - ∗ *For* $i \in [n], k \in [m]$, *let* $i||k$ *be the concatenations of the binary representation of* i *and* k. *For* $b \in \{0,1\}$, *let* $x^{i,k}_{\ell,b} := F(k^\ell_{\mathsf{prf}}, b||i||k)$. *Prepare the quantum state*

$$|\mathsf{qsh}_i^{\ell-1}\rangle = \bigotimes_{k\in[m]} \left(|x^{i,k}_{\ell,0}\rangle + (-1)^{\mathsf{csh}^{\ell-1}_{i,k}} |x^{i,k}_{\ell,1}\rangle\right)$$

  - *For each* $i \in [n]$, *initialize register* Sh_i *with the state*

$$\left(\bigotimes_{\ell\in[-1,n-1]} |\mathsf{qsh}_i^\ell\rangle, \mathsf{csh}_i^n\right).$$

- $\mathsf{Reconstruct}_{\mathbb{A}}(\{\mathsf{Sh}_i\}_{i\in A}, A)$:
  - *Parse each share as the tuple* $\left(\bigotimes_{\ell\in[-1,n-1]} |\mathsf{qsh}_i^\ell\rangle, \mathsf{csh}_i^n\right)$, *and compute*

$$k^n_{\mathsf{prf}} \leftarrow \mathsf{SS}_{\mathsf{classical}}.\mathsf{Reconstruct}(\{\mathsf{csh}_i^n\}_{i\in A}).$$

  - *For* $\ell = n-1, ..., 0$ *do the following:*
    - ∗ *Compute* $x^{i,k}_{\ell+1,b} = F(k^{\ell+1}_{\mathsf{prf}}, b||i||k)$. *Measure* $|\mathsf{qsh}_i^\ell\rangle$ *in the Hadamard basis to obtain strings* $\{d^{\ell+1}_{i,k}\}$. *Compute* $\mathsf{csh}^\ell_{i,k} := d^{\ell+1}_{i,k} \cdot \left(x^{i,k}_{\ell+1,0} \oplus x^{i,k}_{\ell+1,1}\right)$ *for each* $i \in [n], k \in [m]$, *and set* $\mathsf{csh}_i^\ell := \mathsf{csh}^\ell_{i,1} ... \mathsf{csh}^\ell_{i,m}$.
    - ∗ *Compute* $k^\ell_{\mathsf{prf}} \leftarrow \mathsf{SS}_{\mathsf{classical}}.\mathsf{Reconstruct}(\{\mathsf{csh}_i^\ell\}_{i\in S})$.
  - *Use the PRF key* k^0_{prf} *that was reconstructed at the end of the above loop to recover the secret shares* $\{\mathsf{csh}_i^{-1}\}_{i\in A}$ *and then recover the secret* s.
- $\mathsf{Delete}_{\mathbb{A}}(\mathsf{Sh}_i)$: *Apply a computational basis measurement to register* Sh_i *to obtain strings* $\{x^{i,k}_\ell\}_{i,\ell\in[n],k\in[m]}$. *Output* $\{(i, \ell, k, x^{i,k}_\ell)\}$.
- $\mathsf{Verify}_{\mathbb{A}}(\mathsf{vk}, i, \mathsf{cert})$: *Parse* cert *as* $\{(i, \ell, k, x^{i,k}_\ell)\}$. *If* $f(x^{i,k}_\ell) \in \{y^{i,k}_{0,\ell}, y^{i,k}_{1,\ell}\}$ *for all* i, ℓ, k, *then output* $\top$, *otherwise output* $\bot$.

The proof of security is similar to that of Construction 1 and can be found in Appendix B.

Theorem 2. *Construction 3 has adaptive publicly verifiable deletion security.*

References

1. Ambainis, A., Mosca, M., Tapp, A., Wolf, R.: Private quantum channels. In: 41st Annual Symposium on Foundations of Computer Science. pp. 547–553. IEEE Computer Society Press, Redondo Beach, CA, USA (Nov 12–14, 2000). https://doi.org/10.1109/SFCS.2000.892142
2. Bartusek, J., Goyal, V., Khurana, D., Malavolta, G., Raizes, J., Roberts, B.: Software with certified deletion. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024, Part IV. Lecture Notes in Computer Science, vol. 14654, pp. 85–111. Springer, Cham, Switzerland, Zurich, Switzerland (May 26–30, 2024). https://doi.org/10.1007/978-3-031-58737-5_4
3. Bartusek, J., Khurana, D.: Cryptography with certified deletion. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023, Part V. Lecture Notes in Computer Science, vol. 14085, pp. 192–223. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 20–24, 2023). https://doi.org/10.1007/978-3-031-38554-4_7
4. Bartusek, J., Khurana, D., Malavolta, G., Poremba, A., Walter, M.: Weakening assumptions for publicly-verifiable deletion. In: Rothblum, G.N., Wee, H. (eds.) TCC 2023: 21st Theory of Cryptography Conference, Part IV. Lecture Notes in Computer Science, vol. 14372, pp. 183–197. Springer, Cham, Switzerland, Taipei, Taiwan (Nov 29 – Dec 2, 2023). https://doi.org/10.1007/978-3-031-48624-1_7
5. Bartusek, J., Raizes, J.: Secret sharing with certified deletion. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024, Part VII. Lecture Notes in Computer Science, vol. 14926, pp. 184–214. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 18–22, 2024). https://doi.org/10.1007/978-3-031-68394-7_7
6. Benaloh, J.C., Leichter, J.: Generalized secret sharing and monotone functions. In: Goldwasser, S. (ed.) Advances in Cryptology – CRYPTO'88. Lecture Notes in Computer Science, vol. 403, pp. 27–35. Springer, New York, USA, Santa Barbara, CA, USA (Aug 21–25, 1990). https://doi.org/10.1007/0-387-34799-2_3
7. Broadbent, A., Islam, R.: Quantum encryption with certified deletion. In: Pass, R., Pietrzak, K. (eds.) TCC 2020: 18th Theory of Cryptography Conference, Part III. Lecture Notes in Computer Science, vol. 12552, pp. 92–122. Springer, Cham, Switzerland, Durham, NC, USA (Nov 16–19, 2020). https://doi.org/10.1007/978-3-030-64381-2_4
8. Chandran, N., Kanukurthi, B., Obbattu, S.L.B., Sekar, S.: Adaptive extractors and their application to leakage resilient secret sharing. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021, Part III. Lecture Notes in Computer Science, vol. 12827, pp. 595–624. Springer, Cham, Switzerland, Virtual Event (Aug 16–20, 2021). https://doi.org/10.1007/978-3-030-84252-9_20
9. Coladangelo, A., Liu, J., Liu, Q., Zhandry, M.: Hidden cosets and applications to unclonable cryptography. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021, Part I. Lecture Notes in Computer Science, vol. 12825, pp. 556–584. Springer, Cham, Switzerland, Virtual Event (Aug 16–20, 2021). https://doi.org/10.1007/978-3-030-84242-0_20
10. Hiroka, T., Kitagawa, F., Morimae, T., Nishimaki, R., Pal, T., Yamakawa, T.: Certified everlasting secure collusion-resistant functional encryption, and more. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024, Part III. Lecture Notes in Computer Science, vol. 14653, pp. 434–456. Springer, Cham, Switzerland, Zurich, Switzerland (May 26–30, 2024). https://doi.org/10.1007/978-3-031-58734-4_15

11. Hiroka, T., Morimae, T., Nishimaki, R., Yamakawa, T.: Quantum encryption with certified deletion, revisited: Public key, attribute-based, and classical communication. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2021, Part I. Lecture Notes in Computer Science, vol. 13090, pp. 606–636. Springer, Cham, Switzerland, Singapore (Dec 6–10, 2021). https://doi.org/10.1007/978-3-030-92062-3_21
12. Hiroka, T., Morimae, T., Nishimaki, R., Yamakawa, T.: Certified everlasting zero-knowledge proof for QMA. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology – CRYPTO 2022, Part I. Lecture Notes in Computer Science, vol. 13507, pp. 239–268. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 15–18, 2022). https://doi.org/10.1007/978-3-031-15802-5_9
13. Kitagawa, F., Nishimaki, R., Yamakawa, T.: Publicly verifiable deletion from minimal assumptions. In: Rothblum, G.N., Wee, H. (eds.) TCC 2023: 21st Theory of Cryptography Conference, Part IV. Lecture Notes in Computer Science, vol. 14372, pp. 228–245. Springer, Cham, Switzerland, Taipei, Taiwan (Nov 29 – Dec 2, 2023). https://doi.org/10.1007/978-3-031-48624-1_9
14. Poremba, A.: Quantum proofs of deletion for learning with errors. Cryptology ePrint Archive, Report 2022/295 (2022), https://eprint.iacr.org/2022/295
15. Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery **22**(11), 612–613 (Nov 1979). https://doi.org/10.1145/359168.359176
16. Unruh, D.: Revocable quantum timed-release encryption. In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology – EUROCRYPT 2014. Lecture Notes in Computer Science, vol. 8441, pp. 129–146. Springer, Berlin, Heidelberg, Germany, Copenhagen, Denmark (May 11–15, 2014). https://doi.org/10.1007/978-3-642-55220-5_8
17. Wichs, D., Zirdelis, G.: Obfuscating compute-and-compare programs under LWE. In: Umans, C. (ed.) 58th Annual Symposium on Foundations of Computer Science. pp. 600–611. IEEE Computer Society Press, Berkeley, CA, USA (Oct 15–17, 2017). https://doi.org/10.1109/FOCS.2017.61
18. Wiesner, S.: Conjugate coding. SIGACT News **15**(1), 78–88 (jan 1983). https://doi.org/10.1145/1008908.1008920, https://doi.org/10.1145/1008908.1008920
19. Winter, A.: Coding theorem and strong converse for quantum channels. IEEE Transactions on Information Theory **45**(7), 2481–2485 (1999). https://doi.org/10.1109/18.796385
20. Zhandry, M.: How to construct quantum random functions. In: 53rd Annual Symposium on Foundations of Computer Science. pp. 679–687. IEEE Computer Society Press, New Brunswick, NJ, USA (Oct 20–23, 2012). https://doi.org/10.1109/FOCS.2012.37

A Proof of Security for Construction 2

Theorem 3. *Construction 2 has adaptive publicly verifiable deletion security.*

We will prove the theorem by showing that $\mathsf{TD}(\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s), \mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{sim}}(s)) \leq \mathsf{negl}(\lambda)$, where the two experiments are defined below.

$$\underline{\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s) \quad \boxed{\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{sim}}(s)}}$$

- $\{\mathsf{csh}^s_i\}_{i\in[n]} \leftarrow \mathsf{Share}(s)$ $\boxed{\{\mathsf{csh}^s_i\}_{i\in[n]} \leftarrow \{0,1\}}$
- Run $\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s)$ as normal except use the shares generated above as the underlying classical shares for s in Construction 2.

We first introduce the following hybrids which are almost identical to those introduced in the proof of security for Construction 1.

- $\mathsf{Hyb}'_0(s)$: This is the same as $\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s)$ except we lazy sample the underlying classical shares as the adversary corrupts them.
  - Sample uniform PRF key $k_0 \leftarrow \mathcal{K}$, and set $x^{i,k}_b := F(k_0, b||i||k)$ for $i \in [n], k \in [m], b \in \{0,1\}$. Set $y^{i,k}_b = f(x^{i,k}_b)$. Generate the following quantum shares for the PRF key:
  $$\mathsf{vk}^{\mathsf{PRF}}, \{|\mathsf{qsh}^{\mathsf{PRF}}_i\rangle\} \leftarrow \mathsf{SS}_{\mathsf{comp}}.\mathsf{Share}\left(k_0\right).$$
  - Run the experiment $\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s)$ as follows. Initialize $\mathcal{A}$ with $|\psi\rangle$, and initialize the set of corrupted and deleted shares C and D as empty. When $\mathcal{A}$ corrupts a share c, generate the classical share as $\mathsf{csh}_c \leftarrow \mathsf{Share}_c(s, \{\mathsf{csh}_i\}_{i\in C})$, and prepare the following corresponding quantum encoding on register Sh_c:
  $$|\mathsf{qsh}_c\rangle_{\mathsf{Sh}_c} := \bigotimes_{k\in m} \left(|x^{c,k}_0\rangle + (-1)^{\mathsf{csh}_{c,k}}|x^{c,k}_1\rangle\right).$$
  Then add c to the set of corrupted share indices C.
- $\mathsf{Hyb}_0(s)$: In this hybrid we purify the classical share generation by introducing a set of registers $\mathsf{C}_1, ..., \mathsf{C}_n$ held by the challenger which will hold superpositions of classical shares. The share registers $\mathsf{Sh}_1, ..., \mathsf{Sh}_n$ will then be generated based on the states on the challengers registers.
  - Sample a uniform PRF key $k_0 \leftarrow \mathcal{K}$ and set $x^{i,k}_b = F(k_0, b||i||k)$ for $i \in [n], k \in [m], b \in \{0,1\}$. Set $y^{i,k}_b = f(x^{i,k}_b)$, and set the verification key $\mathsf{vk} = \{y^{i,k}_0, y^{i,k}_1\}_{i\in[n],k\in[m]}$. Generate the following quantum shares for the PRF key:
  $$\mathsf{vk}^{\mathsf{PRF}}, \{|\mathsf{qsh}^{\mathsf{PRF}}_i\rangle\} \leftarrow \mathsf{SS}_{\mathsf{comp}}.\mathsf{Share}\left(k_0\right).$$

- Run the experiment $\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s)$ as follows. Whenever a new share c is corrupted, prepare a state on registers C_c and Sh_c as follows. Run the procedure $\mathsf{Share}_c(s, \{\mathsf{C}_i\}_{i\in C})$ coherently on the superposition of sets of shares defined by the challengers registers to obtain

$$\mathsf{C}_c \leftarrow \mathsf{Share}_c(s, \{\mathsf{C}_i\}_{i\in C}).$$

Let $\sum_{\mathsf{csh}_c\in\{0,1\}^m} \alpha_{\mathsf{csh}_c}|\mathsf{csh}_c\rangle_{\mathsf{C}_c}$ be the state on register C_c. Prepare the following state on register Sh_c by running the quantum share encoding procedure coherently on C_c:

$$\frac{1}{2^{m/2}} \sum_{\mathsf{csh}_c} \alpha_{\mathsf{csh}_c}|\mathsf{csh}_c\rangle_{\mathsf{C}_c} \bigotimes_{k\in[m]} \left(|x_0^{c,k}\rangle + (-1)^{\mathsf{csh}_{c,k}}|x_1^{c,k}\rangle\right)_{\mathsf{Sh}_c}.$$

Additionally add $|\mathsf{qsh}_c^{\mathsf{PRF}}\rangle$ to Sh_c. Add c to C.
- Measure each C_i in the computational basis, and then output the result of $\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s)$.

– $\mathsf{Hyb}_i(s)$ for $i \in [n]$: Run Hyb_0 with the following exception. For the first i deletions, after each share j is deleted, measure register C_j with respect to the binary projective measurement $\{\Pi_{\mathsf{cert}_j}, \mathbb{1} - \Pi_{\mathsf{cert}_j}\}$. If the measurement result is "reject" (i.e. has measurement outcome $\mathbb{1} - \Pi_{\mathsf{cert}_j}$), output $\perp$ and abort the experiment.

– $\mathsf{Sim}_i(s)$ for $i \in [0, n]$: Run $\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s)$ as follows.
- When $\mathcal{A}$ corrupts a share c, prepare the following state on registers C_c and Sh_c:

$$\frac{1}{2^{m/2}} \sum_{\mathsf{csh}_c\in\{0,1\}^m} |\mathsf{csh}_c\rangle_{\mathsf{C}_c} \bigotimes_{k\in[m]} \left(|x_0^{c,k}\rangle + (-1)^{\mathsf{csh}_{c,k}}|x_1^{c,k}\rangle\right)_{\mathsf{Sh}_c}.$$

Add $|\mathsf{qsh}_c^{\mathsf{PRF}}\rangle$ to Sh_c and add c to C.
- For the first i deletions, $d_1, ..., d_i$, after the challenger verifies cert_{d_b}, perform the binary projective measurement $\{\Pi_{\mathsf{cert}_{d_b}}, \mathbb{1} - \Pi_{\mathsf{cert}_{d_b}}\}$. Abort and output $\perp$ immediately after any measurement that rejects (i.e. has measurement outcome $\mathbb{1} - \Pi_{\mathsf{cert}}$).

Claim 4. For every secret s,

$$\mathsf{TD}\left(\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s), \mathsf{Hyb}_0(s)\right) \le \mathsf{negl}(\lambda)$$

Proof. The proof is identical to that Claim 1 □

Claim 5. For every secret s and every $i \in [n]$,

$$\mathsf{TD}\left(\mathsf{Hyb}_i^T(s), \mathsf{Sim}_i^T(s)\right) \le \mathsf{negl}(\lambda)$$

Proof. The proof is identical to that of Claim 2 □

The proof of the following claim is almost identical to that of Claim 3, with several changes due to the use of a computational secret sharing scheme in place of a compute-and-compare obfuscator.

Claim 6. For every $i \in [0, n]$ and every secret s,

$$\mathsf{TD}(\mathsf{Hyb}_i(s), \mathsf{Hyb}_{i+1}(s)) \leq \mathsf{negl}(\lambda).$$

Proof. The only difference between $\mathsf{Hyb}_i(s)$ and $\mathsf{Hyb}_{i+1}(s)$ is a Hadamard measurement on register $\mathsf{C}_{d_{i+1}}$ in Hyb_{i+1}, where d_{i+1} is the index of the $(i+1)$th share that is deleted. Suppose that the Hadamard measurement rejects with probability at most ϵ. It follows from the Gentle Measurement Lemma that, conditioned on the Hadamard measurement accepting, the trace distance between Hyb_i and Hyb_{i+1} is at most $2\sqrt{\epsilon}$. It follows that

$$\mathsf{TD}(\mathsf{Hyb}_i(s), \mathsf{Hyb}_{i+1}(s)) \leq (1-\epsilon)2\sqrt{\epsilon} + \epsilon.$$

Therefore to prove the claim we will show that the probability that the Hadamard measurement rejects is negligible. We start by observing that the probability of acceptance is almost identical in each of the following hybrids.

- Hyb_{i+1}:
- Hyb^T_{i+1}: Run Hyb_{i+1} but abort and output State after the $(i+1)$st deletion.
- Sim^T_{i+1}: Run Sim_{i+1} but abort and output State after the $(i+1)$st deletion.

Since $\mathsf{Hyb}_{i+1}(s)$ and $\mathsf{Hyb}^T_{i+1}(s)$ are identical up to the round where the ith Hadamard test is applied, the acceptance probability is identical in both cases. By Claim 2, we have

$$\mathsf{TD}(\mathsf{Hyb}_{i+1}(s), \mathsf{Sim}_{i+1}(s)) \leq \mathsf{negl}(\lambda).$$

Therefore it suffices to show that the probability that the final deletion test in Sim^T_{i+1} does not pass is negligible.

Recall that the projective measurement Π_{cert} simply measures the classical share register C in the Hadamard basis to obtain a string $c_1...c_m$ and checks if the deletion proof $\mathsf{cert} := x_{b_1}, ..., x_{b_m}$ that was just output by $\mathcal{A}$ is such that $b_k = c_k$ for all $k \in [m]$. Since measurements on disjoint registers commute perfectly, we can instead measure C in the Hadamard basis at the start of the experiment to obtain a string $c_1...c_m$ and then run the experiment until $\mathcal{A}$ deletes the corresponding share and outputs a proof $x_{b_1}...x_{b_m}$. Since the measurements commute, the probability that $c_k = b_k$ for all $i \in [m]$ is identical in each case. With this in mind, we define the following experiment which is essentially identical to Sim_{i+1} except that we perform the Hadamard measurement on C before running the adversary as described above.

Fix some share index d, and suppose that d has a non-negligible chance of being deleted in the $(i+1)$th round. We will show that conditioned on d being deleted, the Hadamard test passes with high probability. Suppose otherwise. Then experiment $\mathsf{Expt}_0(d)$ shown below must output 1 with non-negligible probability.

– $\mathsf{Expt}_0(d)$ $\boxed{\mathsf{Expt}_1(d)}$ $\boxed{\boxed{\mathsf{Expt}_2(d)}}$

- Sample uniform PRF key k_0 and compute $x_b^{i,k} := F(k_0, b||i||k)$ for $i \in [n], k \in [m], b \in \{0,1\}$. Set $y_b^{i,k} = f(x_b^{i,k})$.
- $\mathsf{vk}^{\mathsf{PRF}}, \{|\mathsf{qsh}_i^{\mathsf{PRF}}\rangle\}_{i \in [n]} \leftarrow \mathsf{SS}_{\mathsf{comp}}.\mathsf{Share}(k_0)$ $\boxed{\mathsf{vk}^{\mathsf{PRF}}, \{|\mathsf{qsh}_i^{\mathsf{PRF}}\rangle\}_{i \in [n]} \leftarrow \mathsf{SS}_{\mathsf{comp}}.\mathsf{Share}(0)}$.
- $\boxed{\boxed{\text{Resample the preimages uniformly as } x_0^{i,k}, x_1^{i,k} \leftarrow \{0,1\}^\kappa}}$
- Proceed as in Sim_{i+1} but with the following exception. If $\mathcal{A}$ corrupts the dth share, prepare the state

$$\frac{1}{2^{m/2}} \sum_{\mathsf{csh}_d \in \{0,1\}^m} |\mathsf{csh}_d\rangle_{\mathsf{C}_d} \bigotimes_{k \in [m]} \left(|x_0^{d,k}\rangle + (-1)^{\mathsf{csh}_{d,k}} |x_1^{d,k}\rangle\right)_{\mathsf{Sh}_d}$$

  on registers Sh_d and C_d, and measure C_d in the Hadamard basis to obtain measurement outcome $c_1, ..., c_m$. Note that the residual state on register Sh_d is given by

$$\bigotimes_{k \in [m]} |x_{c_k}^{d,k}\rangle$$

- Run Sim_{i+1}, sampling the shares uniformly, up until $\mathcal{A}$ outputs the $(i+1)$th proof of deletion $\mathsf{cert} := (x_{b_1}, ..., x_{b_m})$.
- If the $(i+1)$th proof of deletion is not for share d, then abort and output $\perp$.
- If $b_k \neq c_k$ for some $i \in [m]$, output 1, and otherwise output $\perp$.

We first show that

$$\Pr[\mathsf{Expt}_0(d) \text{ outputs } 1] \neq \mathsf{negl}(\lambda) \implies \Pr[\mathsf{Expt}_1(d) \text{ outputs } 1] \neq \mathsf{negl}(\lambda).$$

If the above does not hold, then we can construct a distinguisher that breaks the security of the computational secret sharing scheme $\mathsf{SS}_{\mathsf{comp}}$. Note that $\mathsf{Expt}_0(d)$ and $\mathsf{Expt}_1(d)$ only differ in which secret is being shared by the computational secret sharing scheme. We present our distinguisher $\mathcal{B}_{\mathsf{SS}}$ below.

$\underline{\mathcal{B}_{\mathsf{SS}}}$

– Hardcode the index d.
– Sample a PRF key $k_{\mathsf{prf}} \leftarrow \mathcal{K}$ and give the challenger $(k_{\mathsf{prf}}, 0)$. The Challenger then creates secret shares $\{|\mathsf{qsh}_i^{\mathsf{PRF}}\rangle\}$ of either k_{prf} or 0.
– Compute $x_b^{i,k} := F(k_{\mathsf{prf}}, b||i||k)$ for $i \in [n], k \in [m], b \in \{0,1\}$.
– Simulate the experiment Sim_{i+1} with adversary $\mathcal{A}$ as follows. If $\mathcal{A}$ requests to corrupt share q do the following:
  - If $q \neq d$, prepare a uniform classical share csh_i, and encode it with the appropriate preimages as the following state:

$$\bigotimes_{k \in [m]} \left(|x_0^{q,k}\rangle + (-1)^{\mathsf{csh}_{q,k}} |x_1^{q,k}\rangle\right).$$

  - If $q = d$, sample uniform $c_1, ..., c_m \in \{0,1\}$ and prepare the state $|\mathsf{qsh}_q^s\rangle := \bigotimes_{i\in[m]} |x_{c_i}\rangle$.

  Corrupt the share $|\mathsf{qsh}_q^{\mathsf{PRF}}\rangle$ and initialize register Sh_q to the state $(|\mathsf{qsh}_q^s\rangle, |\mathsf{qsh}_q^{\mathsf{PRF}}\rangle)$. Give Sh_q to $\mathcal{A}$.
- If $\mathcal{A}$ outputs as its $(i+1)$th deletion proof, a valid proof of deletion $\mathsf{cert}_d := x_{b_1}, ..., x_{b_n}$ for Sh_d, do the following:
  - If $c_k \neq b_k$ for some $k \in [m]$, output k_{prf}
  - Otherwise, output either k_{prf} or 0 with equal probability.
- If $\mathcal{A}$ does not output a proof of deletion for the share d, then output either k_{prf} or 0 with equal probability.

Note that the above adversary perfectly simulates either $\mathsf{Expt}_0(d)$ or $\mathsf{Expt}_1(d)$ depending on if the challenger secret shares 0 or the PRF key k_0. Let Good denote the event that in the execution of $\mathcal{B}_{\mathsf{SS}}$ above, $\mathcal{A}$ outputs a valid proof of deletion for d, and $c_k \neq b_k$ for some $k \in [m]$ (note that this is equivalent to $\mathsf{Expt}_0(d)/\mathsf{Expt}_1(d)$ outputting 1). Note that $\mathcal{B}_{\mathsf{SS}}$ guesses randomly unless an event in Good occurs. Therefore the advantage of $\mathcal{B}_{\mathsf{SS}}$ is

$$\begin{aligned} &\Big|\Pr\left[\mathcal{B}_{\mathsf{SS}} \text{ outputs } \mathsf{real} \mid \mathsf{Expt}_0(d)\right] - \Pr[\mathcal{B}_{\mathsf{SS}} \text{ outputs } \mathsf{real} \mid \mathsf{Expt}_1(d)]\Big| \\ &= \frac{1}{2}\Big| \Pr[\mathsf{Expt}_0(d) \text{ outputs } 1] - \Pr[\mathsf{Expt}_1(d) \text{ outputs } 1]\Big| \end{aligned}$$

By assumption the first term above is non-negligible and the second term is negligible, which implies that $\mathcal{B}_{\mathsf{SS}}$ has non-negligible advantage against the computational secret sharing scheme as desired.

We now show that

$$\Pr[\mathsf{Expt}_1(d) \text{ outputs } 1] \neq \mathsf{negl}(\lambda) \implies \Pr[\mathsf{Expt}_2(d) \text{ outputs } 1] \neq \mathsf{negl}(\lambda).$$

If the above does not hold, we can construct an adversary $\mathcal{B}_{\mathsf{PRF}}^{|\mathcal{O}\rangle}$ against the PRF, which we present below. The adversary uses the oracle $\mathcal{O}$ to generate the preimages, and then simulates the remainder of the experiment in $\mathsf{Expt}_1(d)/\mathsf{Expt}_2(d)$.

$$\underline{\mathcal{B}_{\mathsf{PRF}}^{|\mathcal{O}\rangle}}$$

- Hardcode the index d.
- Compute $x_b^{i,k} = \mathcal{O}(b||i||k)$ for $i \in [n], k \in [m], b \in \{0,1\}$. Generate quantum shares $\{|\mathsf{qsh}_i^{\mathsf{PRF}}\rangle\}_{i\in[n]} \leftarrow \mathsf{SS}_{\mathsf{comp}}.\mathsf{Share}(0)$.
- Simulate the experiments $\mathsf{Expt}_1(d)/\mathsf{Expt}_2(d)$ with adversary $\mathcal{A}$ as follows. If $\mathcal{A}$ requests to corrupt share q do the following:
  - If $q \neq d$, prepare a uniform classical share csh, and encode it with the appropriate preimages as the following state:

$$|\mathsf{qsh}_q^s\rangle := \bigotimes_{k\in[m]} \left(|x_0^{q,k}\rangle + (-1)^{\mathsf{csh}_{q,k}} |x_1^{q,k}\rangle\right)$$

  - If $q = d$, sample uniform $c_1, ..., c_m \in \{0,1\}$, and prepare the state $|\mathsf{qsh}^s\rangle := \bigotimes_{i\in[m]} |x_{c_i}\rangle$.

Initialize register Sh_q with the state $(|\mathsf{qsh}_q^s\rangle, |\mathsf{qsh}_q^{\mathsf{PRF}}\rangle)$ and give Sh_q to $\mathcal{A}$.

– If $\mathcal{A}$ outputs a valid proof of deletion $\mathsf{cert}_d := x_{b_1}, ..., x_{b_n}$ for Sh_d, do the following:
  - If $c_k \neq b_k$ for some $k \in [m]$, output PRF.
  - Otherwise, output either PRF or $\mathsf{Uniform}$ with equal probability.
– If $\mathcal{A}$ does not output a proof of deletion for the share d, then output either PRF or $\mathsf{Uniform}$ with equal probability.

If the oracle is a random function, then $\mathcal{B}_{\mathsf{PRF}}^{|\mathcal{O}\rangle}$ perfectly simulates $\mathsf{Expt}_2(d)$. and if the oracle is a PRF then $\mathcal{B}_{\mathsf{PRF}}^{|\mathcal{O}\rangle}$ perfectly simulates $\mathsf{Expt}_1(d)$. Let Good denote the event that the simulation of $\mathsf{Expt}_1(d)/\mathsf{Expt}_2(d)$ in an execution of the adversary $\mathcal{B}_{\mathsf{PRF}}^{|\mathcal{O}\rangle}$ above outputs 1. Note that the adversary guesses that the oracle is a PRF if an event in Good occurs, and otherwise the adversary outputs a random guess. Therefore the advantage of $\mathcal{B}_{\mathsf{PRF}}^{|\mathcal{O}\rangle}$ is given by the following:

$$\left|\Pr\left[\mathcal{B}_{\mathsf{PRF}}^{|\mathcal{O}\rangle} \text{ outputs } \mathsf{real} \mid \mathsf{Expt}_0(d)\right] - \Pr\left[\mathcal{B}_{\mathsf{PRF}}^{|\mathcal{O}\rangle} \text{ outputs } \mathsf{real} \mid \mathsf{Expt}_1(d)\right]\right|$$
$$= \frac{1}{2}\Big| \Pr[\mathsf{Expt}_1(d) \text{ outputs } 1] - \Pr[\mathsf{Expt}_2(d) \text{ outputs } 1]\Big|.$$

Therefore if the first term above is non-negligible but the second term is negligible, then $\mathcal{B}_{\mathsf{PRF}}^{|\mathcal{O}\rangle}$ violates the security of the PRF.

We now show that $\Pr[\mathsf{Expt}_2(d) \text{ outputs } 1] \leq \mathsf{negl}(\lambda)$. If this is not the case, than we can construct an adversary $\mathcal{B}_{\mathsf{OWF}}$ against the one-way function which we present below.

$\underline{\mathcal{B}_{\mathsf{OWF}}(y)}$

– Hardcode the index d, and sample uniform index $k_0 \in [m]$, and uniform $b_0 \in \{0,1\}$.
– Set $y_{b_0}^{d,k_0} := y$. Sample uniform $x_b^{i,k} \leftarrow \{0,1\}^\kappa$ for $(b,i,k) \neq (b_0,d,k_0)$ and set $y_b^{i,k} = f(x_b^{i,k})$.
– Generate quantum shares $\{|\mathsf{qsh}_i^{\mathsf{PRF}}\rangle\}_{i\in[n]} \leftarrow \mathsf{SS}_{\mathsf{comp}}.\mathsf{Share}(0)$.
– Simulate $\mathsf{Expt}_2(d)$ as follows. Initialize $\mathcal{A}$ with $|\psi\rangle$. When $\mathcal{A}$ corrupts some share q, do the following:
  - If $q \neq d$, sample a uniform classical string $\mathsf{csh} \leftarrow \{0,1\}^m$, and encode it with the corresponding preimages.
$$|\mathsf{qsh}_i^s\rangle := \bigotimes_{k\in[m]} \left(|x_0^{q,k}\rangle + (-1)^{\mathsf{csh}_{q,k}}|x_1^{q,k}\rangle\right)$$
  - If $q = d$, prepare the state
$$|\mathsf{qsh}_q^s\rangle := \bigotimes_{k\in[m]\setminus\{k_0\}} \left(|x_0^{d,k}\rangle + (-1)^{\mathsf{csh}_{d,k}}|x_1^{d,k}\rangle\right) \bigotimes |x_{1-b}^{d,k_0}\rangle.$$

Initialize register Sh_q with the state $(|\mathsf{qsh}_q^s\rangle, |\mathsf{qsh}_q^{\mathsf{PRF}}\rangle)$ and give Sh_q to $\mathcal{A}$.

– If $\mathcal{A}$ outputs as part of a certificate of deletion, a preimage of y, then output y. Otherwise output $\perp$.

Note that if the input y is the evaluation of a uniform preimage, then $\mathcal{B}_{\mathsf{OWF}}$ perfectly simulates $\mathsf{Expt}_2(d)$. If $\mathsf{Expt}_2(d)$ has a non-negligible chance of outputting 1, then the above procedure has a non-negligible chance of inverting the one way function.

The above implies that $\mathsf{Expt}_0(d)$ outputs 1 with negligible probability, and therefore the Hadamard test passes with overwhelming probability. $\square$

Proof (of Theorem 3). With the above claims in hand the main result easily follows. In more detail, note that $\mathsf{Sim}_n(s)$ is identical to $\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{rand}}(s)$, and therefore $\mathsf{TD}\left(\mathsf{Sim}_n(s), \mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{rand}}(s)\right) = 0$. Claim 4 implies $\mathsf{TD}(\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s), \mathsf{Hyb}'_0(s)) \leq \mathsf{negl}(\lambda)$. Claim 5 implies $\mathsf{TD}(\mathsf{Hyb}_0(s), \mathsf{Hyb}_n(s)) \leq \mathsf{negl}(\lambda)$, and Claim 6 implies that $\mathsf{TD}(\mathsf{Hyb}_n(s), \mathsf{Sim}_n(s)) \leq \mathsf{negl}(\lambda)$. Putting the above together we obtain the lemma statement. $\square$

B Proof of Security for Construction 3

In this section we prove the security of Construction 3.

Theorem 4. *Construction 3 has adaptive publicly verifiable deletion security.*

We will prove the theorem by showing that $\mathsf{TD}(\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}, \mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{sim}}) \leq \mathsf{negl}(\lambda)$, where the two experiments are defined below.

$$\underline{\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s) \quad \boxed{\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{sim}}(s)}}$$

– $\{\mathsf{csh}^s_i\}_{i\in[n]} \leftarrow \mathsf{Share}(s)$ $\boxed{\{\mathsf{csh}^s_i\}_{i\in[n]} \leftarrow \{0,1\}^m}$
– Run Construction 3 but replace $\{\mathsf{csh}^{-1}_i\}_{i\in[n]}$ (i.e. the classical shares of the secret being shared) with the strings $\{\mathsf{csh}^s_i\}_{i\in[n]}$ generated above.

For notational convenience we write the classical shares of the secret s as csh^{-1} rather than csh^s, and we additionally set $k_{-1} := s$. We define the sets $\mathsf{Good}_i := \{-1, ..., n-i-1\}$ and $\mathsf{Bad}_i := [n] \setminus \mathsf{Good}_i$. Intuitively the set Good_i will be the set of indicies which corresond to shares that are actually deleted

We introduce the following hybrids.

– $\mathsf{Hyb}'_0(s)$: This is the same as $\mathsf{Expt}^{\mathsf{SS\text{-}ACD}}_{\mathsf{real}}(s)$ except we lazy sample the underlying classical shares as the adversary corrupts them.
  - Sample uniform PRF keys $k^0_{\mathsf{prf}}, ..., k^n_{\mathsf{prf}} \leftarrow \mathcal{K}$. Compute $x^{i,k}_{\ell,b} = F(k^\ell_{\mathsf{prf}}, b||i||k)$ for all $i \in [n], k \in [m], \ell \in [n], b \in \{0,1\}$.

- Run the experiment $\mathsf{SS\text{-}ACD}(1^\lambda, |\psi\rangle, \mathcal{A}, s)$ as follows. Initialize $\mathcal{A}$ with $|\psi\rangle$. When $\mathcal{A}$ corrupts a share c, generate the following classical shares for $\ell \in \{-1, 0, ..., n\}$:

$$\mathsf{csh}_c^\ell \leftarrow \mathsf{Share}_c(k_{\mathsf{prf}}^\ell, \{\mathsf{csh}_i^\ell\}_{i \in C}).$$

For $\ell \in \{-1, 0, ..., n-1\}$, prepare the following state on register Sh_c^ℓ:

$$|\mathsf{qsh}_c^\ell\rangle_{\mathsf{Sh}_c^\ell} := \bigotimes_{k \in [m]} \left(|x_{\ell+1,0}^{c,k}\rangle + (-1)^{\mathsf{csh}_{c,k}^\ell} |x_{\ell+1,1}^{c,k}\rangle \right).$$

Initialize the register Sh_c^n with the state csh_c^n. Add c to C.

– $\mathsf{Hyb}_0(s)$: In this hybrid we purify the classical share generation by introducing a set of registers $\mathsf{C}_1, ..., \mathsf{C}_n$ held by the challenger which will hold superpositions of classical shares. The share registers $\mathsf{Sh}_1, ..., \mathsf{Sh}_n$ will then be generated based on the states on the challengers registers.
  - Sample uniform PRF keys $k_{\mathsf{prf}}^0 ..., k_{\mathsf{prf}}^n \leftarrow \mathcal{K}$. Compute $x_{\ell,b}^{i,k} = F(k_{\mathsf{prf}}^\ell, b||i||k)$ for all $i \in [n], k \in [m], \ell \in [n], b \in \{0, 1\}$.
  - When $\mathcal{A}$ corrupts a share c, prepare registers $\{\mathsf{C}_c^\ell\}_{\ell \in [-1,n]}$ and $\{\mathsf{Sh}_c^\ell\}_{\ell \in [-1,n]}$ as follows. Run the procedure $\mathsf{Share}_c(k_{\mathsf{prf}}^\ell, \{\mathsf{C}_i^\ell\}_{i \in C})$ coherently on the superposition of sets of shares defined by the challengers registers to obtain

$$\mathsf{C}_c^\ell \leftarrow \mathsf{Share}_c(k_{\mathsf{prf}}^\ell, \{\mathsf{C}_i^\ell\}_{i \in C})$$

for $\ell \in \{-1, ..., n\}$. Let $\sum_{\mathsf{csh}_c \in \{0,1\}^m} \alpha_{\mathsf{csh}_c}^\ell |\mathsf{csh}_c\rangle_{\mathsf{C}_c}$ be the state on C_c^ℓ. For each $\ell \in \{-1, 0, ..., n-1\}$, prepare the following state on register Sh_c^ℓ:

$$\sum_{\mathsf{csh}_c^\ell} \alpha_{\mathsf{csh}_c^\ell} |\mathsf{csh}_c^\ell\rangle_{\mathsf{C}_c^\ell} \bigotimes_{k \in [m]} \left(|x_{\ell+1,0}^{c,k}\rangle + (-1)^{\mathsf{csh}_{c,k}^\ell} |x_{\ell+1,1}^{c,k}\rangle \right)_{\mathsf{Sh}_c^\ell}.$$

Initialize the register Sh_c^n with the same state as C_c^n. Add c to C.
  - For $i \in [n]$, $\ell \in \{-1, ..., n\}$, measure each C_i^ℓ in the computational basis, and then output the result of $\mathsf{SS\text{-}ACD}_{\mathbb{A}}(1^\lambda, |\psi\rangle, \mathcal{A}, s)$.

– $\mathsf{Hyb}_i(s)$ for $i \in [n]$: Run $\mathsf{Hyb}_0(s)$ with the following exception. For the first i deletions, do the following. Immediately after deletion j (for $j \in [i]$), measure register C_j^ℓ with respect to the binary projective measurement $\{\Pi_{\mathsf{cert}_j^\ell}, \mathbb{1} - \Pi_{\mathsf{cert}_j^\ell}\}$ for each $\ell \in \{-1, ..., n-j-1\}$. If any of the measurement results are "reject" (i.e. has measurement outcome $\mathbb{1} - \Pi_{\mathsf{cert}_j^\ell}$), output $\perp$ and abort the experiment.

– $\mathsf{Sim}_i(s)$ for $i \in [0, n]$: Here every classical share corresponding to the secret s and the PRF keys $k_{\mathsf{prf}}^0, ..., k_{\mathsf{prf}}^{n-i}$ is generated as a uniform string, where i is the number of deletions that have taken place. For $\ell \geq n-i$, the classical shares csh_j^ℓ are generated as shares of the corresponding PRF keys $k_{\mathsf{prf}}^{n-i+1}, ..., k_{\mathsf{prf}}^n$.
  - Run $\mathsf{Hyb}_i(s)$ with the following exception. For all corruptions that take place prior to the $(i+1)$th deletion, do the following. When $\mathcal{A}$ corrupts

a share c, prepare the following states on registers C_c^ℓ and Sh_c^ℓ for each $\ell \in \{-1, 0, ..., n-i-1\}$:

$$\frac{1}{2^{m/2}} \sum_{\mathsf{csh}_c^\ell \in \{0,1\}^m} |\mathsf{csh}_c^\ell\rangle_{\mathsf{C}_c^\ell} \bigotimes_{k \in [m]} \left(|x_{\ell+1,0}^{c,k}\rangle + (-1)^{\mathsf{csh}_{c,k}^\ell} |x_{\ell+1,1}^{c,k}\rangle\right)_{\mathsf{S}_c^\ell}.$$

For $\ell \in \{n-i, ..., n\}$ generate the corresponding classical share registers as follows:

$$\mathsf{C}_c^\ell \leftarrow \mathsf{Share}_c(k_{\mathsf{prf}}^\ell, \{\mathsf{C}_i^\ell\}_{i \in C}).$$

Prepare the corresponding registers $\{\mathsf{Sh}_c^\ell\}$ as above.

- For the first i deletions do the following. For deletion $j \in [i]$, let d_j be the index of the share that is deleted. After the challenger verifies cert_{d_j}, perform the binary projective measurement $\left\{\Pi_{\mathsf{cert}_{d_j}^\ell}, \mathbb{1} - \Pi_{\mathsf{cert}_{d_j}^\ell}\right\}$ on register C_j^ℓ for each $\ell \in \{-1, 0, ..., n-i-1\}$ abort and output $\perp$ if any of these measurement results are "reject".

Lemma 4. *For every secret s,*

$$\mathsf{TD}\left(\mathsf{Expt}_{\mathsf{real}}^{\mathsf{SS\text{-}ACD}}(s), \mathsf{Hyb}_0(s)\right) = 0$$

Proof. First, the fact that $\mathsf{TD}(\mathsf{Expt}_{\mathsf{real}}^{\mathsf{SS\text{-}ACD}}(s), \mathsf{Hyb}_0(s)) = 0$ follows from the definition of the lazy-sampling style sharing procedure used in Hyb_0. The fact that $\mathsf{TD}(\mathsf{Hyb}_0(s), \mathsf{Hyb}_0'(s)) = 0$ follows from the fact that operations on disjoint sets of registers commute, and in particular measuring the challengers registers at the beginning or at the end of the experiment will not impact the state of the adversary $\mathcal{A}$. Since measuring the challengers registers in the computational basis before giving the share registers to $\mathcal{A}$ induces the same distribution over classical shares as in $\mathsf{Hyb}_0(s)$, the result follows. □

The proof of the following is almost identical to the proof of Claim 2. The only difference is that we are now applying the argument to a set of classical share registers $\{\mathsf{C}_i^\ell\}_{\ell \in \{-1,...,n-i-1\}}$ rather than a single classical share register.

Lemma 5. *For $i \in [n]$, and for every secret s,*

$$\mathsf{TD}\left(\mathsf{Hyb}_i^T(s), \mathsf{Sim}_i^T(s)\right) \leq \mathsf{negl}(\lambda).$$

Proof. Recall that the only difference between $\mathsf{Hyb}_i^T(s)$ and $\mathsf{Sim}_i^T(s)$ is that in the former experiment shares corresponding to $\ell \in \{-1, ..., n-i-1\}$ are generated based on previously corrupted shares, and in the latter they are generated as uniform superposition states. We will prove the claim by induction on i.

To see that the claim holds for $i = 0$, note that prior to the first deletion, it follows from the property of the classical secret sharing scheme that any unauthorized set of shares (and in particular whichever subset is queried by the adversary prior to the first deletion) is perfectly indistinguishable from uniformly random strings. Therefore in Hyb_0, each share is a uniform superposition, and $\mathsf{TD}(\mathsf{Hyb}_0^T(s), \mathsf{Sim}_0^T(s)) = 0$.

Now suppose that $\mathsf{TD}(\mathsf{Hyb}_i^T(s), \mathsf{Sim}_i^T(s)) \leq \mathsf{negl}(\lambda)$. We show that the claim holds for $i+1$ by introducing the following sequence of hybrids.

- Hyb_{i+1}^T : Run Hyb_{i+1} but abort and output the adversaries register State as soon as the $(i+1)$th deletion measurement has been passed.
- $\mathsf{Hyb}_{i+1}'^T$: Run Hyb_{i+1}^T, up until $\mathcal{A}$ outputs its ith deletion. Then, for each subsequent corruption j, prepare the following state on registers $\{\mathsf{C}_j^\ell\}_{\ell\in[-1,n]}$ and $\{\mathsf{Sh}_j^\ell\}_{\ell\in[-1,n]}$:
  - For $\ell \in \{-1, 0, ..., n-i-1\}$, prepare
$$\frac{1}{2^{m/2}} \sum_{\mathsf{csh}_c^\ell \in \{0,1\}^m} |\mathsf{csh}_c^\ell\rangle_{\mathsf{C}_c^\ell} \bigotimes_{k\in[m]} \left(|x_{\ell,0}^{c,k}\rangle + (-1)^{\mathsf{csh}_{c,k}^\ell} |x_{\ell,1}^{c,k}\rangle\right)_{\mathsf{Sh}_c^\ell}.$$
  - For $\ell \in \{n-i, ..., n\}$, prepare the state
$$\mathsf{C}_c^\ell \leftarrow \mathsf{Share}_c\left(k_{\mathsf{prf}}^\ell, \{\mathsf{C}_i^\ell\}_{i\in C}\right),$$
    and prepare the corresponding state on registers $\{\mathsf{Sh}_j^\ell\}_{\ell\in\{0,...,n-i\}}$ using the corresponding evaluations of the PRF.

  Once $\mathcal{A}$ outputs its $(1+1)$th deletion and the corresponding Hadamard measurement has passed, abort and output the adversaries register State.
- Sim_{i+1}^T :

We will first show that
$$\mathsf{TD}\left(\mathsf{Hyb}_{i+1}'^T(s), \mathsf{Sim}_{i+1}^T(s)\right) \leq \mathsf{negl}(\lambda).$$

First, note that $\mathsf{Hyb}_{i+1}'^T$ is identical to Hyb_i up until the ith deletion (but prior to the Hadamard measurement on the deleted share registers), and identical to Sim_{i+1}^T after the ith deletion. Additionally, by the inductive hypothesis we have $\mathsf{TD}(\mathsf{Hyb}_i^T, \mathsf{Sim}_i^T) \leq \mathsf{negl}(\lambda)$. Since the (mixed) states of the two experiments prior to the ith deletion are negligibly close, and the procedures are identical after this point, the result follows.

We now show that
$$\mathsf{TD}\left(\mathsf{Hyb}_{i+1}^T(s), \mathsf{Hyb}_{i+1}'^T(s)\right) \leq \mathsf{negl}(\lambda).$$

Let $\mathsf{Good}_i := \{-1, ..., n-i-1\}$. To show the above, we will prove that each corrupted share generated after the ith deletion but before the $(i+1)$th deletion in Hyb_{i+1}^T is in a uniform superposition if it corresponds to a PRF key in $\{k_{\mathsf{prf}}^\ell\}_{\ell\in\mathsf{Good}_i}$. To do so, we will argue that ignoring the deleted share registers $\{\mathsf{C}_i^\ell\}_{i\in D,\ell\in\mathsf{Good}_i}$ and generating each newly corrupted share based only on the shares in $C \setminus D$ does not change the outcome of the experiment. Since $C \setminus D$ is never authorized, it follows from the uniformity property of the classical secret sharing scheme that generating each newly corrupted share based on $C \setminus D$ results in a uniform superposition.

We introduce the following sequence of hybrids which give different ways of generating the shares corrupted after the ith deletion in Hyb_{i+1}^T.

- Expt_0: Run Hyb^T_{i+1} with no changes. In particular, each newly corrupted is generated as follows based on all shares in C, including those that have been deleted:

$$\mathsf{C}^\ell_j \leftarrow \mathsf{Share}_j(k^\ell_{\mathsf{prf}}, \{\mathsf{C}^\ell_i\}_{i\in C}).$$

- Expt_1: Run Hyb^T_{i+1} but generate each share after the ith deletion as follows:
  - For $\ell \in \mathsf{Good}_i$, generate fresh share registers for the deleted shares based on the shares in $C \setminus D$:

$$\{\mathsf{C}'^\ell_i\}_{i\in D} \leftarrow \mathsf{Share}_D(k^\ell_{\mathsf{prf}}, \{\mathsf{C}^\ell_i\}_{i\in C\setminus D}).$$

  - For $\ell \in \mathsf{Good}_i$, generate each newly corrupted share based on the shares in $C \setminus D$ together with the freshly generated share registers $\{\mathsf{C}'^\ell_i\}_{i\in D}$:

$$\mathsf{C}^\ell_j \leftarrow \mathsf{Share}_j(k^\ell_{\mathsf{prf}}, \{\mathsf{C}^\ell_i\}_{i\in C\setminus D} \cup \{\mathsf{C}'^\ell_i\}_{i\in D}).$$

- Expt_2: Run Hyb^T_{i+1} but generate each newly corrupted share as follows based only on shares in $C \setminus D$:

$$\mathsf{C}^\ell_j \leftarrow \mathsf{Share}_j(k^\ell_{\mathsf{prf}}, \{\mathsf{C}^\ell_i\}_{i\in C\setminus D}).$$

The only difference between Expt_1 and Expt_2 is that in Expt_2, additional share registers for the indices in D are generated before generating C^ℓ_j. Since random variables in a joint distribution can be sampled in any order as a sequence of samples from conditional distributions, it is clear that $\mathsf{SD}(\mathsf{Expt}_1, \mathsf{Expt}_2) = 0$.

To prove that Expt_0 and Expt_1 are identical, note that the only difference between these experiments is that each newly corrupted share C^ℓ_j is generated based on the original deleted share registers $\{\mathsf{C}^\ell_i\}_{i\in D}$ in Expt_0, and based on the freshly generated registers $\{\mathsf{C}'^\ell_i\}_{i\in D}$ in the case of Expt_1. Since the distribution $\mathsf{Share}_j(\cdot)$ takes classical inputs and is being run coherently on superpositions, it is enough to show that a computational basis measurement of the original registers $\{\mathsf{C}^\ell_i\}_{i\in D}$ and the new registers $\{\mathsf{C}'^\ell_i\}_{i\in D}$ induce the same distribution. This follows from the fact that each deleted share register C^ℓ_d is in a Hadamard basis state immediately after being deleted. However by the uniformity property of the underlying classical secret-sharing scheme, if we were to regenerate C^ℓ_d based on the shares in $C \setminus D$ we would obtain a uniform superposition.

Therefore $\mathsf{TD}(\mathsf{Hyb}^T_{i+1}(s), \mathsf{Expt}_2) = 0$. However note that in Expt_2, each corrupted share C^ℓ_j is generated based on a set $C \setminus D$ such that $(C \setminus D) \cup \{j\}$ is not authorized (for otherwise the adversary would obtain an authorized set). Therefore by the uniformity property of the underlying secret sharing scheme, the newly corrupted share registers contain uniform superpositions. It follows that $\mathsf{TD}(\mathsf{Hyb}^T_{i+}, \mathsf{Hyb}'^T_{i+1}) \leq \mathsf{negl}(\lambda)$ as desired which completes the proof. $\square$

Lemma 6. *For $i \in [0, n]$ and every secret s,*

$$\mathsf{TD}\left(\mathsf{Hyb}_i(s), \mathsf{Hyb}_{i+1}(s)\right) \leq \mathsf{negl}(\lambda).$$

Proof. The only difference between Hyb_i and Hyb_{i+1} is an additional Hadamard measurement on registers $\{\mathsf{C}^\ell_{d_{i+1}}\}_{\ell\in\mathsf{Good}_i}$ in Hyb_{i+1}, where d_{i+1} is the index of the $(i+1)$th deleted share. We will show that the probability that the Hadamard measurements corresponding to the $(i+1)$th deletion in Hyb_{i+1} reject is negligible. We start by observing that the probability of acceptance is almost identical in each of the following hybrids.

- Hyb_{i+1} :
- Hyb^T_{i+1} : Run Hyb_{i+1} but abort and output State after the $(i+1)$st deletion.
- Sim^T_{i+1} : Run Sim_{i+1} but abort and output State after the $(i+1)$st deletion.

$\square$

Since Hyb_{i+1} and Hyb^T_{i+1} are identical up to the round where the $(i+1)$th Hadamard test is applied, the acceptance probability is identical in both cases. By Lemma 5, we have

$$\mathsf{TD}(\mathsf{Hyb}^T_{i+1}, \mathsf{Sim}^T_{i+1}) \leq \mathsf{negl}(\lambda).$$

Therefore it suffices to show that the probability that the final deletion test in Sim^T_{i+1} does not pass is negligible.

Recall that the projective measurement $\Pi_{\mathsf{cert}_{d_{i+1}}} = \left\{\Pi_{\mathsf{cert}^\ell_{d_{i+1}}}\right\}$ simply measures the classical share registers $\{\mathsf{C}^\ell_{d_{i+1}}\}$ in the Hadamard basis to obtain strings $c^\ell_1...c^\ell_m$ and checks if the deletion proof $\mathsf{cert}^\ell_{d_{i+1}} := x^\ell_{b_1}, ..., x^\ell_{b_m}$ that was just output by $\mathcal{A}$ is such that $b^\ell_k = c^\ell_k$ for all $k \in [m]$ and $\ell \in \{-1, ..., n-i-1\}$. Since measurements on disjoint registers commute perfectly, we can instead measure $\{\mathsf{C}^\ell_{d_{i+1}}\}$ in the Hadamard basis at the start of the experiment to obtain string $c^\ell_1...c^\ell_m$ and then run the experiment until $\mathcal{A}$ deletes the corresponding share and outputs a proof $x^\ell_{b_1}...x^\ell_{b_m}$. Since the measurements commute, the probability that $c^\ell_k = b^\ell_k$ for all $i \in [m]$ and $\ell \in \{-1, ..., n-i-1\}$ is identical in each case. With this in mind, we define the following experiment which is essentially identical to Sim_{i+1} except that we perform the Hadamard measurement on $\{\mathsf{C}^\ell_{d_{i+1}}\}_\ell$ before running the adversary as described above.

Fix some share index d, and suppose that d has a non-negligible probability of being deleted in the $(i+1)$th round. We show that conditioned on share d being deleted, the Hadamard test passes with high probability. Suppose otherwise. Then the following experiment $\mathsf{Expt}_0(d)$ must output 1 with non-negligible probability.

- $\mathsf{Expt}_0(d)$ $\boxed{\mathsf{Expt}_1(i_0)}$
  - Fix a share index d and set $k^{-1}_{\mathsf{prf}} = s$.
  - Sample PRF keys $k^0_{\mathsf{prf}}, ..., k^n_{\mathsf{prf}} \leftarrow \mathcal{K}$.
  - For $\ell \in \{n-i, ..., n\}$, sample shares $\{\mathsf{csh}^\ell_i\} \leftarrow \mathsf{Share}(k^\ell_{\mathsf{prf}})$, and for $\ell \in \{0, ..., n-i\}$, sample uniform strings $\{\mathsf{csh}^\ell_i\}_{i\in[n]} \leftarrow \{0,1\}^m$.
  - For $\ell \in [n] \setminus [n-i-1]$, set $x^{i,k}_{\ell,b} = F(k^\ell_{\mathsf{prf}}, b||i||k)$.

- For $\ell \in \{-1, ..., n-i-1\}$, set $x^{i,k}_{\ell,b} = F(k^{\ell}_{\mathsf{prf}}, b||i||k)$.
$\boxed{\text{For } \ell \in \{-1, ..., n-i-1\}, \text{ sample } x^{i,k}_{\ell,b} \leftarrow \{0,1\}^{\kappa}.}$
- Proceed as in Sim_{i+1} but with the following exception. If $\mathcal{A}$ corrupts the dth share, prepare the states

$$\sum_{\mathsf{csh}} |\mathsf{csh}\rangle_{\mathsf{C}^{\ell}_{d}} \bigotimes_{k \in [m]} \left(|x^{d,k}_{\ell+1,0}\rangle + (-1)^{\mathsf{csh}_{d,k}} |x^{d,k}_{\ell+1,1}\rangle \right)_{\mathsf{Sh}^{\ell}_{d}}$$

on registers S^{ℓ}_{d} and C^{ℓ}_{d} for $\ell \in \{-1, ..., n-i-1\}$. Measure C^{ℓ}_{d} in the Hadamard basis to obtain measurement outcome $c^{\ell}_{1}, ..., c^{\ell}_{m}$. Note that the residual state on register S^{ℓ}_{d} is given by

$$\bigotimes_{k \in [m]} |x^{d,k}_{c^{\ell}_{k},\ell}\rangle$$

Continue to run Sim_{i+1}, sampling the shares uniformly, up until $\mathcal{A}$ outputs the $(i+1)$th proof of deletion $\{\mathsf{cert}_j := (x_{b^{\ell}_{1}}, ..., x_{b^{\ell}_{m}})\}_{\ell \in \{-1,...,n-i-1\}}$.
- If the $(i+1)$th proof of deletion is not for share d, then abort and output $\perp$.
- If $b^{\ell}_{k} \neq c^{\ell}_{k}$ for some $k \in [m], \ell \in \{-1, ..., n-i-1\}$, output 1, and otherwise output $\perp$.

We first show that

$$\Pr[\mathsf{Expt}_0(d) \text{ outputs } 1] \neq \mathsf{negl}(\lambda) \implies \Pr[\mathsf{Expt}_1(d) \text{ outputs } 1] \neq \mathsf{negl}(\lambda).$$

If this is not the case, then we can construct an adversary against the PRF. Note that the only difference between Expt_0 and Expt_1 is that in the latter experiment, the preimages corresponding to $\ell \in \{0, ..., n-i\}$ are uniform, and in the former they are the evaluations of a PRF. However in both cases the PRF keys are not in the adversaries view.

$$\underline{\mathcal{B}^{|\mathcal{O}\rangle}_{\mathsf{PRF}}}$$

– Hardcode the index d.
– Parse the oracle $\mathcal{O}$ as a list of $n-i$ oracles $\mathcal{O}_0, ..., \mathcal{O}_{n-i}$.
– Sample uniform PRF keys $k^{n-i+1}_{\mathsf{prf}}, ..., k^{n}_{\mathsf{prf}}$.
– For $\ell \in [n] \setminus [n-i]$, compute $x^{i,k}_{\ell,b} := F(k^{\ell}_{\mathsf{prf}}, b||i||k)$, and for $\ell \in \{-1, ..., n-i\}$, compute $x^{i,k}_{\ell,b} = \mathcal{O}_{\ell}(b||i||k)$.
– For $\ell \in [n] \setminus [n-i]$ run $\{\mathsf{csh}^{\ell}_{i}\}_{i \in [n]} \leftarrow \mathsf{Share}(k^{\ell}_{\mathsf{prf}})$, and for $i \in \{-1, 0, ..., n-i\}$, sample uniform strings $\{\mathsf{csh}^{\ell}_{i}\}_{i \in [n]} \leftarrow \{0,1\}^{m}$.
– Using the preimages and classical shares computed above, simulate $\mathsf{Expt}_0(d)/\mathsf{Expt}_1(d)$. Let $c^{\ell}_{1}, ..., c^{\ell}_{m}$ be the result of the Hadamard measurement on C_d in $\mathsf{Expt}_0(d)/\mathsf{Expt}_1(d)$.
– If $\mathcal{A}$ outputs a proof of deletion $\mathsf{cert}_d = x_{b_1} ... x_{b_m}$ for share d such that $b_k \neq c_k$ for some $k \in [m]$, then guess that the oracle is a PRF, and otherwise output a uniform guess.

A straightforward hybrid argument shows that an adversary winning the above game implies an adversary winning the standard PRF security game with a single oracle. In the above, if the oracles are for uniform functions, then $\mathcal{B}_{\mathsf{PRF}}$ perfectly simulates Expt_1, and if the oracles are for PRFs with uniform keys, then $\mathcal{B}_{\mathsf{PRF}}$ perfectly simulates Expt_0. Note that $\mathcal{B}_{\mathsf{PRF}}$ outputs a random guess except in the situation that corresponds to $\mathsf{Expt}_0(d)/\mathsf{Expt}_1(d)$ outputting 1. It follows that the distinguishing advanatage of $\mathcal{B}_{\mathsf{PRF}}$ is given by the following expression:

$$\frac{1}{2}\Big|\Pr[\mathsf{Expt}_0(d) \text{ outputs } 1] - \Pr[\mathsf{Expt}_1(d) \text{ outputs } 1]\Big|.$$

By assumption the first term above is non-negligible, and the second term is negligible,. which implies $\mathcal{B}_{\mathsf{PRF}}$ has a non-negligible distinguishing advantage.

We now show that $\Pr[\mathsf{Expt}_1(d) \text{ outputs } 1] \leq \mathsf{negl}(\lambda)$. If this is not the case, then we can construct an adversary against the PRF. Our adversary on input a uniform y will simulate $\mathsf{Expt}_1(d)$, and try to get the adversary $\mathcal{A}$ to output of a preimage of y by putting y in the verification key for share d. The adversary is presented below.

$\underline{\mathcal{B}_{\mathsf{OWF}}(y)}$

- Hardcode the index d, and sample uniform $k_0 \leftarrow [m]$, $b_0 \leftarrow \{0,1\}$, and $\ell_0 \leftarrow \{-1, 0, ..., n-i\}$.
- Set $y^{d,k_0}_{\ell_0,b_0} := y$. For $(i,k,\ell,b) \neq (d,k_0,\ell_0,b_0)$, sample uniform $x^{i,k}_{\ell,b} \leftarrow \{0,1\}^\kappa$ and set $y^{i,k}_{\ell,b} = f(x^{i,k}_{\ell,b})$.
- For $\ell \in \{i, ..., n\}$, set
- Simulate Expt_2 as follows. Initialize $\mathcal{A}$ with $|\psi\rangle$. When $\mathcal{A}$ corrupts some share $q \in [n]$, do the following:
  - If $q \neq d$, sample uniform $\mathsf{csh}^\ell_i \leftarrow \{0,1\}^m$ for $\ell \in \{-1, 0, ..., n-i\}$, and encode them with the corresponding preimages as the following state:
  $$\bigotimes_{k\in[m]} \left(|x^{q,k}_0\rangle + (-1)^{\mathsf{csh}_{q,k}}|x^{q,k}_1\rangle\right)$$
  - If $q = d$, prepare prepare the states as above for $\ell \neq \ell_0$, but encode $\mathsf{csh}^{\ell_0}_i$ as the following state:
  $$\bigotimes_{k\neq k_0} \left(|x^{q,k}_{\ell,0}\rangle + (-1)^{\mathsf{csh}_{q,k}}|x_{\ell,1}\rangle^{q,k}\right) \bigotimes |x^{q,k_0}_{\ell,1-b}\rangle$$
- If $\mathcal{A}$ outputs as part of a certificate of deletion, a preimage of y, then output y. Otherwise output $\perp$.

If the input y to the adversary is the evaluation of f on a uniform preimage, then the above adversary perfectly simulates $\mathsf{Expt}_2(d)$. If the simulation of $\mathsf{Expt}_2(d)$ outputs 1, then $\mathcal{B}_{\mathsf{OWF}}$ succeeds in outputting a preimage of y. Therefore $\Pr[\mathsf{Expt}_1(d) \text{ outputs } 1] \leq \mathsf{negl}(\lambda)$.

Proof (of Theorem 4). First, by Lemma 4 we have $\mathsf{TD}(\mathsf{Expt}_{\mathsf{real}}(s), \mathsf{Hyb}'_0(s)) \leq \mathsf{negl}(\lambda)$. Lemma 6 implies that $\mathsf{TD}(\mathsf{Hyb}_1(s), \mathsf{Hyb}_n(s)) \leq \mathsf{negl}(\lambda)$ and Lemma 5 implies that $\mathsf{TD}(\mathsf{Hyb}_n(s), \mathsf{Sim_n(s)}) \leq \mathsf{negl}(\lambda)$. Putting the above together we have $\mathsf{TD}(\mathsf{Expt}_{\mathsf{real}}(s), \mathsf{Sim}_n(s)) \leq \mathsf{negl}(\lambda)$. □

C Adaptive Certified Deletion $\not\Longrightarrow$ No-Signaling Certified Deletion

In this section we construct a secret sharing scheme which has adaptive certified deletion but does not satisfy no-signaling certified deletion. First, we recall the no-signaling certified deletion definition given by Bartusek and Raizes [5].

Definition 9. *Let $P = (P_1, ..., P_\ell)$ be a partition of $[n]$, let $|\psi\rangle$ be an ℓ-part state on registers $\mathsf{State}_1, ..., \mathsf{State}_\ell$, and let $\mathcal{A} = (\mathcal{A}_1, ..., \mathcal{A}_\ell)$ be an ℓ-part QPT adversary. Define the experiment $\mathsf{SS\text{-}NSCD}_{\mathbb{A}}(1^\lambda, P, |\psi\rangle, \mathcal{A}, s)$ as follows:*

- *Sample $(\mathsf{Sh}_1, ..., \mathsf{Sh}_n) \leftarrow \mathsf{Share}_{\mathbb{A}}(1^\lambda, s)$.*
- *For each $t \in [\ell]$, run $(\{\mathsf{cert}_i\}_{i \in P_t}, \mathsf{State}'_t) \leftarrow \mathcal{A}(\{\mathsf{S}_i\}_{i \in P_t}, \mathsf{State}_t)$, where State'_t is an arbitrary output register.*
- *If for all $S \in \mathbb{A}$, there exists $i \in S$ such that $\mathsf{Verify}(\mathsf{vk}, i, \mathsf{cert}_i) = \top$, then output $(\mathsf{State}'_1, ..., \mathsf{State}'_\ell)$, and otherwise output $\bot$.*

A secret sharing scheme has no-signaling certified deletion security if for any partition $P = (P_1, ..., P_\ell)$, any ℓ-part state $|\psi\rangle$, any ℓ-part QPT adversary $\mathcal{A}$, and any pair of secrets s_0, s_1,

$$\mathsf{TD}\left(\mathsf{SS\text{-}NSCD}_{\mathbb{A}}(1^\lambda, P, |\psi\rangle, \mathcal{A}, s_0), \mathsf{SS\text{-}NSCD}_{\mathbb{A}}(1^\lambda, P, |\psi\rangle, \mathcal{A}, s_1)\right) \leq \mathsf{negl}(\lambda).$$

Our construction starts with an arbitrary secret-sharing scheme with adaptive certified deletion for an access structure $\mathbb{A}$. We assume there are two disjoint (unauthorized) subsets $P_1, P_2 \subset [n]$ along with two indices $i_1 \in P_1$ and $i_2 \in P_2$ such that (1) $\{i_b\} \cup P_{1-b} \in \mathbb{A}$ for $b \in \{0, 1\}$, and (2) that $\{i_0, i_1\} \notin \mathbb{A}$. We note that this condition is satisfied for any threshold scheme with $t < n/2$ as we can set P_1 and P_2 to be two disjoint subsets of size $t - 1$.

Our construction will make use of a quantum one-time pad, first introduced by Ambainis et al. [1] which allows us to encrypt a quantum state with a classical key. We present the syntax and security properties of a quantum one-time pad below. Ambainis et al. gave a concrete construction of such an encryption scheme.

Definition 10 (Quantum one-time pad encryption). *Let $\mathcal{K}$ be a key space, and let $\mathcal{M} := (\mathbb{C}^2)^{\otimes n}$ be a quantum message space. The quantum one-time pad encryption scheme is defined by the following pair of algorithms which have identical syntax:*

- $\mathsf{OTP.Enc}(k, \rho)$*: On input a quantum state $\rho \in \mathcal{M}$ and a classical key $k \in \mathcal{K}$, output a state $\sigma \in \mathcal{M}$.*
- $\mathsf{OTP.Dec}(k, \rho)$*: On input a quantum state $\rho \in \mathcal{M}$ and a classical key $k \in \mathcal{K}$, output a state $\sigma \in \mathcal{M}$.*

Correctness: *For all keys $k \in \mathcal{K}$ and any state ρ,*

$$\mathsf{OTP.Dec}\left(k, \mathsf{OTP.Enc}(k, \rho)\right) = \rho.$$

Security: *For any two states ρ and σ,*

$$\sum_{k \in \mathcal{K}} \frac{1}{\sqrt{|\mathcal{K}|}} \mathsf{OTP.Enc}(k, \rho) = \sum_{k \in \mathcal{K}} \frac{1}{\sqrt{|\mathcal{K}|}} \mathsf{OTP.Enc}(k, \sigma).$$

We now present our construction.

Construction 4 *Let* $\mathsf{SS}_{\mathsf{adaptive}}$ *be a secret sharing scheme with adaptive certified deletion for a monotone access structure* $\mathbb{A}$*, and let* $\mathsf{SS}_{(t,n)}$ *be a* (t,n)*-threshold secret sharing scheme with adaptive certified deletion. Let* P_0 *and* P_1 *be two disjoint unauthorized subsets such that there exist indices* $i_0 \in P_0$ *and* $i_1 \in P_0$ *with the property that* $P_0 \cup \{i_1\}, P_1 \cup \{i_0\} \in \mathbb{A}$*, and* $\{i_0, i_1\} \notin \mathbb{A}$*. Let* $\mathsf{OTP}(\mathsf{pad}, |\psi\rangle)$ *denote the procedure that applies a quantum pad to its second argument, using its first argument as a classical key.*

- $\mathsf{Share}(1^\lambda, s)$:
  - *Generate shares* $\{|\mathsf{qsh}_i\rangle\}_{i\in[n]} \leftarrow \mathsf{SS}_{\mathsf{adaptive}}.\mathsf{Share}(s)$.
  - *Create a* $(4,4)$ *sharing* $\{|\mathsf{qsh}_0^x\rangle, |\mathsf{qsh}_1^x\rangle, |\mathsf{qsh}_0^y\rangle, |\mathsf{qsh}_1^y\rangle\} \leftarrow \mathsf{SS}_{(4,4)}.\mathsf{Share}(s)$.
  - *Sample random strings* pad_0 *and* pad_1*,*
  - *Create the following* $(|P_b|, |P_b|)$*-sharing of* pad_b *for* $b \in \{0,1\}$*:*

$$\{|\mathsf{qsh}_i^{\mathsf{pad}_b}\rangle\}_{i\in P_b} \leftarrow \mathsf{Share}_{(|P_b|,|P_b|)}(\mathsf{pad}_b).$$

  - *Using* pad_b *as a quantum one-time pad, encrypt* $|\mathsf{qsh}_1^x\rangle$ *and* $|\mathsf{qsh}_1^b\rangle$ *to obtain*

$$|\mathsf{ct}_0\rangle \leftarrow \mathsf{OTP}.\mathsf{Enc}(\mathsf{pad}_0, |\mathsf{qsh}_0^x\rangle) \quad \textit{and} \quad |\mathsf{ct}_1\rangle \leftarrow \mathsf{OTP}.\mathsf{Enc}(\mathsf{pad}_1, |\mathsf{qsh}_1^x\rangle)$$

  - *For* $i \in [n]$*, initialize register* Sh_i *to the state* $|\mathsf{qsh}_i\rangle$*.*
  - *For* $b \in \{0,1\}$*, for* $i \in P_b$*, add state* $|\mathsf{qsh}_i^{\mathsf{pad}_b}\rangle|\mathsf{ct}_b\rangle$ *to share register* Sh_i*.*
  - *For* $b \in \{0,1\}$*, add the state* $|\mathsf{qsh}_b^y\rangle$ *to the share register* Sh_{i_b}*.*
- $\mathsf{Reconstruct}(\{\mathsf{Sh}_i\}_{i\in P})$: *Run the reconstruction algorithm for the underlying scheme* $\mathsf{SS}_{\mathsf{adaptive}}$ *and ignore any of the additional shares.*
- $\mathsf{Delete}(\mathsf{Sh}_i)$: *If* $i \notin P_0 \cup P_1$*, run* $\mathsf{SS}_{\mathsf{adaptive}}.\mathsf{Delete}(\mathsf{Sh})$*. if* $i \in P_b$*, run any additional deletion algorithms for the corresponding state.*
- $\mathsf{Verify}(i, \mathsf{vk}, \mathsf{cert}_i)$: *Run the corresponding verification algorithms for any quantum shares contained on register* Sh_i*. If any verification fails then output* $\bot$*, otherwise output* $\top$*.*

Lemma 7. *Construction 4 has adaptive certified deletion but is insecure against a no-signaling adversary.*

Proof. The scheme is clearly insecure against a no-signaling adversary by construction. For any partition which includes P_0 and P_1 from the construction, $\mathcal{A}_b$ does the following for $b \in \{0,1\}$. Recover pad_b by computing

$$\mathsf{pad}_b \leftarrow \mathsf{Reconstruct}\left(\{|\mathsf{qsh}_i^{\mathsf{pad}_b}\rangle\}_{i\in P_b}\right).$$

The above is deterministic and therefore can be done without disturbing any shares. Then compute $|\mathsf{qsh}_b^x\rangle \leftarrow \mathsf{OTP}.\mathsf{Dec}(\mathsf{pad}_b, |\mathsf{ct}_b\rangle)$. Finally delete all shares except $|\mathsf{qsh}_b^y\rangle$, and output $(|\mathsf{qsh}_b^x\rangle, |\mathsf{qsh}_b^y\rangle)$. The combined views of $\mathcal{A}_0$ and $\mathcal{A}_1$ can then be used to reconstruct the secret using $\{|\mathsf{qsh}_0^x\rangle, |\mathsf{qsh}_1^x\rangle, |\mathsf{qsh}_0^y\rangle, |\mathsf{qsh}_1^y\rangle\}$.

We now prove that the scheme is adaptively secure. Let $\mathcal{A}$ be an adaptive adversary. First note that we can replace the shares $\{|\mathsf{qsh}_i\rangle\}_{i\in[n]} \leftarrow \mathsf{SS}_{\mathsf{adaptive}}.\mathsf{Share}(s)$ with shares of 0 by appealing to the security of $\mathsf{SS}_{\mathsf{adaptive}}$. Therefore it remains to show that the additional shares do not allow an adaptive adversary to break the scheme.

Let SS' denote the modified secret sharing where $\mathsf{SS}_{\mathsf{adaptive}}$ is used to generate shares of 0 rather than of the secret. If $\mathcal{A}$ breaks SS', we can assume that with overwhelming probability, P_0 and P_1 are each contained in $C \setminus D$ at some point (though not necessarily at the same time) in the experiment. If this is not the case, then at least one of pad_0 or pad_1 remain hidden from $\mathcal{A}$. Therefore it follows from the security of the quantum one-time pad that the shares $\{|\mathsf{qsh}^x_{i_b}\rangle\}_{b\in\{0,1\}}$ remain hidden from $\mathcal{A}$.

With the above in mind, assume that P_0 is contained in $C \setminus D$ before P_1 is. Recall that by assumption $P_b \cup \{i_{1-b}\}$ is an authorized set for $b \in \{0,1\}$. Therefore $\mathcal{A}$ must delete Sh_{i_1} before corrupting all shares in P_1, and in particular must delete the share $|\mathsf{qsh}^y_{i_0}\rangle$ prior to obtaining P_1. This means that an adversary against the $(4,4)$-secret sharing can simulate $\mathcal{A}$ without ever being forced to obtain an authorized set of shares, thus violating the security of the threshold secret-sharing scheme if $\mathcal{A}$ is able to guess the secret that was shared. $\square$

D Secret Sharing with No-Signaling PVD

In this section we provide a sketch of the secret sharing construction with no-signaling certified deletion based on the construction in [5]. Their secret sharing construction makes black-box use of a $(2,2)$-secret sharing scheme for a single bit in which one share is classical, and the other is a quantum state that can be certifiably deleted. They additionally require that the deletion security is sub-exponential. More precisely, they require that the post deletion states in the cases that the secret was 0 or 1 have trace distance at most $1/\mathsf{subexp}(\lambda)$. Therefore if we can construct a $(2,2)$-scheme with the above properties that also has publicly verifiable deletion, we can simply plug the scheme into the secret-sharing construction of Bartusek and Raizes to obtain the desired result.

We present such a $(2,2)$-secret sharing scheme below, which is analogous to the scheme used by Bartusek and Raizes. Since the security in the deletion proof is lower bounded by the security of the one-way function, we will require sub-exponential security. We omit the proof of deletion security for the following construction, which is identical to the proof of [4, Theorem 3].

Construction 5 *Let $f : \{0,1\}^{\ell_{\mathsf{in}}} \mapsto \{0,1\}^{\ell_{\mathsf{out}}}$ be a one-way function.*

- $\mathsf{Share}(b)$*: Sample uniform $x_0, x_1 \leftarrow \{0,1\}^\kappa$. The quantum and classical shares are defined as follows:*

$$|\mathsf{qsh}\rangle := |x_0\rangle + (-1)^b|x_1\rangle, \qquad \mathsf{csh} := x_0 \oplus x_1$$

- $\mathsf{Rec}(|\mathsf{qsh}\rangle, \mathsf{csh})$*: Measure $|\mathsf{qsh}\rangle$ in the Hadamard basis to obtain a string d, and compute $b = d \cdot \mathsf{csh}$.*

– $\mathsf{Del}(|\mathsf{qsh}\rangle)$*: Measure* $|\mathsf{qsh}\rangle$ *in the computational basis and output the result.*
– $\mathsf{Ver}(x)$*: If* $f(x) \in \{y_0, y_1\}$ *output* $\top$*, and otherwise output* $\perp$*.*