Cryptography with Certified Deletion

James Bartusek* Dakshita Khurana†

Abstract

We propose a new, unifying framework that yields an array of cryptographic primitives with certified deletion. These primitives enable a party in possession of a quantum ciphertext to generate a classical certificate that the encrypted plaintext has been information-theoretically deleted, and cannot be recovered even given unbounded computational resources.

- For $X \in \{\mathsf{public\text{-}key}, \mathsf{attribute\text{-}based}, \mathsf{fully\text{-}homomorphic}, \mathsf{witness}, \mathsf{timed\text{-}release}\}$, our compiler converts any (post-quantum) X encryption to X encryption with certified deletion.

  In addition, we compile statistically-binding commitments to statistically-binding commitments with certified everlasting hiding. As a corollary, we also obtain statistically-sound zero-knowledge proofs for QMA with certified everlasting zero-knowledge assuming statistically-binding commitments.

- We also obtain a strong form of everlasting security for two-party and multi-party computation in the dishonest majority setting. While simultaneously achieving everlasting security against *all* parties in this setting is known to be impossible, we introduce *everlasting security transfer (EST)*. This enables *any one* party (or a subset of parties) to dynamically and certifiably information-theoretically delete other participants' data after protocol execution.

  We construct general-purpose secure computation with EST assuming statistically-binding commitments, which can be based on one-way functions or pseudorandom quantum states.

We obtain our results by developing a novel proof technique to argue that a bit b has been *information-theoretically deleted* from an adversary's view once they output a valid deletion certificate, despite having been previously *information-theoretically determined* by the ciphertext they held in their view. This technique may be of independent interest.

*UC Berkeley. Email: bartusek.james@gmail.com

†UIUC. Email: dakshita@illinois.edu.

Contents

1 Introduction

Deletion in a classical world. On classical devices, data is stored and exchanged as a string of bits. There is nothing that can prevent an untrusted device with access to such a string from making arbitrarily many copies of it. Thus, it seems hopeless to try to *force* an untrusted device to delete classical data. Even if the string is merely a ciphertext encoding an underlying plaintext, there is no way to prevent a server from keeping that ciphertext around in memory forever. If at some point in the future, the security of the underlying encryption scheme is broken either via brute-force or major scientific advances, or if the key is compromised and makes its way to the server, the server will be able to recover the underlying plaintext. This may be unacceptable in situations where extremely sensitive data is being transmitted or computed upon.

In fact, there has recently been widespread interest in holding data collectors accountable in responding to "data deletion requests" from their clients, as evidenced by data deletion clauses in legal regulations adopted by the European Union [Eur16] and California [Cal18]. Unfortunately, the above discussion shows that these laws cannot be cryptographically enforced against malicious data collectors, though there has been recent work on cryptographically *formalizing* what it means for *honest* data collectors to follow such guidelines [GGV20].

Deletion in a quantum world. The *uncertainty principle* [Hei27], which lies at the foundation of quantum mechanics, completely disrupts the above classical intuition. It asserts the existence of pairs of measurable quantities such that precisely determining one quantity (e.g. the position of an electron) implies the *inability* to determine the other (e.g. the momentum of the electron). While such effects only become noticeable at an extreme microscopic scale, the pioneering work of Wiesner [Wie83] suggested that the peculiar implications of the uncertainty principle could be leveraged to perform seemingly impossible "human-scale" information processing tasks.

Given the inherent "destructive" properties of information guaranteed by the uncertainty principle, provable data deletion appears to be a natural information processing task that, while impossible classically, may become viable quantumly. Surprisingly, the explicit study of data deletion in a quantum world has only begun recently. However, over the last few years, this question has been explored in many different contexts. Initial work studied deletion in the context of time-lock puzzles with revocation [Unr14], non-local games [FM18] and information-theoretic proofs of deletion with partial security [CW19].

In the context of encryption, the work of [BI20] first defined and constructed one-time pad encryption with certified deletion. This led to many recent followup works on deletion in a cryptographic context: device-independent security of one-time pad encryption with certified deletion [KT20], public-key encryption with certified deletion [HMNY21], commitments and zero-knowledge with certified everlasting hiding [HMNY22b], and most recently fully-homomorphic encryption with certified deletion [Por22].

Our work makes new definitional, conceptual and technical contributions. Our key contribution is a new proof technique to show that many natural encryption schemes satisfy security with certified deletion. This improves prior work in many ways.

1. **A unified framework.** We present a simple compiler that relies on conjugate coding/BB84 states [Wie83, BB84] to bootstrap semantically-secure cryptosystems to semantically-secure cryptosystems with certified deletion. For any $X \in \{$public-key encryption, attribute-based encryption, witness encryption, timed-release encryption, statistically-binding commitment$\}$, we immediately obtain "X with certified deletion" by plugging X into our compiler.

2. **Stronger definitions.** We consider a strong definition of security with certified deletion, which stipulates that if an adversary in possession of a quantum ciphertext encrypting bit b issues a certificate of deletion which passes verification, then the bit b must now be *information-theoretically* hidden from the adversary.

   Previous definitions of public-key and fully-homomorphic encryption with certified deletion [HMNY21, Por22] considered a weaker experiment, where after deletion, the adversary is explicitly given the secret key, but is still required to be computationally bounded. We consider this prior definition to capture a (strong) *security against key leakage* property, as opposed to a *certified deletion* property. Thus we propose our definition as the "right" definition of certified deletion in the public-key setting,[1] and we show that our definition implies [HMNY21]'s definition (Appendix A). Intuitively, this is because for public-key schemes, an adversary can sample a secret key on its own given sufficient computational resources. Moreover, in the case of fully-homomorphic encryption (FHE), previous work [Por22] considered definitions (significantly) weaker than semantic security.[2] We obtain the first semantically-secure FHE with certified deletion, and furthermore we only rely on the standard LWE assumption.

3. **Simpler constructions and weaker assumptions.** Our compiler removes the need to rely on complex cryptographic primitives such as non-committing encryption and indistinguishability obfuscation as in [HMNY21], or idealized models such as random oracles as in [Unr14, HMNY22b], or complex quantum states (such as Gaussian coset states) as in [Por22], instead immediately obtaining simple schemes satisfying certified deletion for a range of primitives from BB84 states and minimal assumptions.

   In fact, reliance on non-committing encryption was a key reason that prior techniques did not yield homomorphic encryption schemes with certified deletion, since compact homomorphic encryption schemes cannot simultaneously be non-committing [KTZ13]. Our work builds simple homomorphic encryption schemes that support certified deletion by eliminating the need to rely on non-committing properties, and instead only relying on semantic security of an underlying encryption scheme.

4. **Overcoming barriers to provable security.** How can one prove that a bit b has been *information-theoretically deleted* from an adversary's view once they produce a valid deletion certificate, while it was previously information-theoretically *determined* by the ciphertext they hold in their view?

   Indeed, prior work [Unr14, HMNY21, HMNY22b, Por22] resorted to either idealized models or weaker definitions, and constructions with layers of indirection, in order to get around this barrier. We develop a novel proof technique that resolves this issue by (1) carefully deferring the dependence of the experiment on the plaintext bit, and (2) identifying an efficiently checkable predicate on the adversary's state after producing a valid deletion certificate. We rely on semantic security of encryption to show that this predicate must hold, and we argue that if the predicate holds, the adversary's left-over state is statistically independent of the plaintext bit. This allows us to prove certified deletion security for simple and natural schemes.

[1] In contrast, in the one-time pad encryption setting as considered by [BI20], the original encrypted message is already information-theoretically hidden from the adversary, so to obtain any interesting notion of certified deletion, one must explicitly consider leaking the secret key.

[2] Subsequent to the original posting of our paper on arXiv, an update to [Por22] was posted with somewhat different results. We provide a comparison between our work and the updated version of [Por22] in Section 1.3.

5. **New implications to secure computation: Everlasting Security Transfer (EST).** We introduce the concept of *everlasting security transfer*. Everlasting security guarantees (malicious) security against a participant in a secure two-(or multi-)party computation protocol even if the participant becomes computationally unbounded after protocol execution. We introduce and build secure computation protocols where participants are able to *transfer* everlasting security properties from one party to another, even after the protocol ends.

We elaborate on our results in more detail below, then we provide an overview of our techniques.

1.1 Our results

Warmup: secret sharing with certified deletion. We begin by considering certified deletion in the context of one of the simplest cryptographic primitives: information-theoretic, two-out-of-two secret sharing. Here, a dealer Alice would like to share a classical secret bit b between two parties Bob and Charlie, such that

1. **(Secret sharing.)** The individual views of Bob and Charlie perfectly hide b, while the joint view of Bob and Charlie can be used to reconstruct b, and
2. **(Certified deletion.)** Bob may generate a deletion certificate for Alice, guaranteeing that b has been *information theoretically removed* from the *joint* view of Bob and Charlie.

That is, as long as Bob and Charlie do not collude at the time of generating the certificate of deletion, their joint view upon successful verification of this certificate is guaranteed to become independent of b. As long as the certificate verifies, b will be perfectly hidden from Bob and Charlie *even if* they decide to later collude.

To build such a secret sharing scheme, we start by revisiting the usage of conjugate coding/BB84 states to obtain encryption with certified deletion, which was first explored in [BI20]. While the construction in [BI20] relies on a seeded randomness extractor in combination with BB84 states, we suggest a simpler alternative that replaces the seeded extractor with the XOR function. Looking ahead, this simplification combined with novel proof techniques will help generically lift our secret sharing scheme to obtain several encryption schemes with certified deletion.

Consider a random string $x \leftarrow \{0,1\}^\lambda$, and a random set of bases $\theta \leftarrow \{0,1\}^\lambda$ (where 0 corresponds to the standard basis and 1 corresponds to the Hadamard basis). To obtain a scheme with certifiable deletion, we will build on the intuition that it is impossible to recover x given only BB84 states $|x\rangle_\theta$ without knowledge of the basis θ. Furthermore, measuring $|x\rangle_\theta$ in an incorrect basis θ' will destroy (partial) information about x.

Thus to secret-share a bit b in a way that supports deletion, the dealer will sample $x \leftarrow \{0,1\}^\lambda$ and bases $\theta \leftarrow \{0,1\}^\lambda$. Bob's share is then

$$|x\rangle_\theta$$

and Charlie's share is

$$\theta, b' = b \oplus \bigoplus_{i:\theta_i=0} x_i$$

That is, in Charlie's share, b is masked by the bits of x that are encoded in the standard basis.

We note that Bob's share contains only BB84 states while Charlie's share is entirely classical. Bob can now produce a certificate of deletion by returning the results of measuring all his BB84 states in the Hadamard basis, and Alice will accept as a valid certificate any string x' such that

$x_i = x'_i$ for all i where $\theta_i = 1$. We show that this scheme is indeed a two-out-of-two secret sharing scheme that satisfies certified deletion as defined above.

A conceptually simple and generic compiler. As our key technical contribution, we upgrade the secret sharing with certified deletion scheme to the public-key setting by encrypting Charlie's share. In more detail, to encrypt a bit b with respect to any encryption scheme, we first produce two secret shares of b as described above, and then release a ciphertext that contains (1) Bob's share in the clear and (2) an encryption of Charlie's share. To certifiably delete a ciphertext, one needs to simply measure the quantum part of the ciphertext (i.e., Bob's share) in the Hadamard basis. Intuitively, since information about the bases (Charlie's share) is hidden at the time of producing the certificate of deletion, generating a certificate that verifies must mean information theoretically losing the description of computational basis states.

This method of converting a two-party primitive (i.e. secret sharing with certified deletion) into one-party primitives (i.e. encryption schemes with certified deletion) is reminiscent of other similar compilers in the literature, for instance those converting probabilistically checkable proofs to succinct arguments [BMW98, KR09]. In our case, just like those settings, while the intuition is relatively simple, the proof turns out to be fairly non-trivial.

Our main theorem. In full generality, our main theorem says the following. Consider an arbitrary family of distributions $\{\mathcal{Z}_\lambda(m)\}_{\lambda\in\mathbb{N},m\in\{0,1\}}$, which can be thought of as distributions over ciphertexts encrypting the plaintext $m = 0$ or $m = 1$, and an arbitrary class $\mathscr{A}$ of computationally bounded adversaries $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, such that $\mathcal{A}_\lambda$ can only distinguish between $\mathcal{Z}_\lambda(0)$ and $\mathcal{Z}_\lambda(1)$ with negligible probability. Let $\mathcal{Z}_\lambda(m)$ for a bitstring $m \in \{0,1\}^n$ denote $\mathcal{Z}_\lambda(m_1), \ldots, \mathcal{Z}_\lambda(m_n)$. Then, consider the following distribution $\widetilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(b)$ over quantum states, parameterized by a bit $b \in \{0,1\}$.

- Sample $x, \theta \leftarrow \{0,1\}^\lambda$ and initialize $\mathcal{A}_\lambda$ with

$$|x\rangle_\theta, \mathcal{Z}_\lambda\left(\theta, b \oplus \bigoplus_{i:\theta_i=0} x_i\right).$$

- $\mathcal{A}_\lambda$'s output is parsed as a bitstring $x' \in \{0,1\}^\lambda$ and a residual quantum state ρ.
- If $x_i = x'_i$ for all i where $\theta_i = 1$ then output ρ, and otherwise output a special symbol $\perp$.

Then,

Theorem 1.1. *For every $\mathcal{A} \in \mathscr{A}$, the trace distance between $\widetilde{Z}_\lambda^{\mathcal{A}_\lambda}(0)$ and $\widetilde{Z}_\lambda^{\mathcal{A}_\lambda}(1)$ is* $\mathrm{negl}(\lambda)$.

Intuitively, this means that as long as the adversary $\mathcal{A}_\lambda$ is computationally bounded *at the time of producing any deletion certificate* x' that properly verifies (meaning that x'_i is the correct bit encoded at index i for any indices encoded in the Hadamard basis), their left-over state *statistically* contains only negligible information about the original encrypted bit b. That is, once the certificate verifies, information about b cannot be recovered information-theoretically even given unbounded time from the adversary's residual state.

This theorem is both quite simple and extremely general. The quantum part that enables certified deletion only involves simple BB84 states, and we require no additional properties of the

underlying distribution $\mathcal{Z}_\lambda$ except for the fact that $\mathcal{Z}_\lambda(0)$ and $\mathcal{Z}_\lambda(1)$ are indistinguishable to some class of adversaries.[3] We now discuss our applications in more detail.

Public-key, attribute-based and witness encryption. Instantiating the distribution $\mathcal{Z}_\lambda$ with the encryption procedure for any public-key encryption scheme, we obtain a public-key encryption scheme with certified deletion.

We also observe that we can instantiate the distribution $\mathcal{Z}_\lambda$ with the encryption procedure for any *attribute-based* encryption scheme, and immediately obtain an attribute-based encryption scheme with certified deletion. Previously, this notion was only known under the assumption of indistinguishability obfuscation, and also only satisfied the weaker key leakage style definition discussed above [HMNY21]. Finally, instantiating $\mathcal{Z}_\lambda$ with any *witness encryption* scheme implies a witness encryption scheme with certified deletion.

Fully-homomorphic encryption. Next, we consider the question of computing on encrypted data. We observe that, if $\mathcal{Z}_\lambda$ is instantiated with the encryption procedure Enc for a *fully-homomorphic* encryption scheme [Gen09, BV11, GSW13], then given $|x\rangle_\theta, \mathsf{Enc}(\theta, b \oplus \bigoplus_{i:\theta_i=0} x_i)$, one could run a homomorphic evaluation procedure in superposition to recover (a superposition over) $\mathsf{Enc}(b)$. Additionally, given multiple ciphertexts, one can even compute arbitrary functionalities over the encrypted plaintexts. Moreover, if such evaluation is done *coherently* (without performing measurements), then it can be reversed and the deletion procedure can subsequently be run on the original ciphertexts.

This immediately implies what we call a "blind delegation with certified deletion" protocol, which allows a computationally weak client to utilize the resources of a computationally powerful server, while (i) keeping its data hidden from the server during the protocol, and (ii) ensuring that its data is *information-theoretically* deleted from the server afterwards, by requesting a certificate of deletion. We show that, as long as the server behaves honestly[4] during the "function evaluation" phase of the protocol, then even if it is arbitrarily malicious after the function evaluation phase, it cannot both pass deletion verification and maintain any information about the client's original plaintexts.

Recently, Poremba [Por22] also constructed a fully-homomorphic encryption scheme satisfying a weaker notion of certified deletion.[5] In particular, the guarantee in [Por22] is that from the perspective of any server that passes deletion with *sufficiently high probability*, there is significant entropy in the client's original *ciphertext*. This does not necessarily imply anything about the underlying plaintext, since a ciphertext encrypting a fixed bit b may be (and usually will be) highly entropic. Moreover, their construction makes use of relatively complicated and highly entangled *Gaussian coset states* in order to obtain these deletion properties. In summary, our framework

[3] It may seem counter-intuitive that the certified deletion guarantees provided by our theorem hold even when instantiating $\mathcal{Z}_\lambda$ with general semantically secure schemes, such as a fully-homomorphic encryption scheme. In particular, what if an adversary evaluated the FHE to recover a *classical* encryption of b, and then reversed their computation and finally produced a valid deletion certificate? This may seem to contradict everlasting security, since a classical ciphertext could be used to recover b given unbounded time. However, this attack is actually not feasible. After performing FHE evaluation coherently, the adversary would obtain a register holding a superposition over classical ciphertexts encrypting b, but with different random coins. Measuring this superposition to obtain a single classical ciphertext would collapse the state, and prevent the adversary from reversing their computation to eventually produce a valid deletion certificate. Indeed, our Theorem rules out this (and all other) efficient attacks.

[4] Technically, we allow arbitrary specious behavior during the function evaluation phase.

[5] We discuss comparisons with a recently updated version of [Por22] in Section 1.3.

simultaneously strengthens the security (to standard semantic security of the plaintext) and simplifies the construction of fully-homomorphic encryption with certified deletion. We also remark that neither our work nor [Por22] considers security against servers that may be malicious during the function evaluation phase of the blind delegation with certified deletion protocol[6]. We leave obtaining security against fully malicious servers as an interesting direction for future research.

Commitments and zero-knowledge. Next, we consider *commitment schemes*. A fundamental result in quantum cryptography states that one cannot use quantum communication to build a commitment that is simultaneously statistically hiding and statistically binding [May97, LC97]. Intriguingly, [HMNY22b] demonstrated the feasibility of statistically-binding commitments with a *certified everlasting hiding* property, where hiding is computational during the protocol, but becomes information-theoretic after the receiver issues a valid deletion certificate. However, their construction relies on the idealized quantum random oracle model. Using our framework, we show that *any* (post-quantum) statistically-binding computationally-hiding commitment implies a statistically-binding commitment with certified everlasting hiding. Thus, we obtain statistically-binding commitments with certified everlasting hiding in the plain model from post-quantum one-way functions, and even from plausibly weaker assumptions like *pseudorandom quantum states* [AQY22, MY22].

Following implications in [HMNY22b] from commitments with certified deletion to zero-knowledge, we also obtain interactive proofs for NP (and more generally, QMA) with *certified everlasting zero-knowledge*. These are proofs that are statistically sound, and additionally the verifier may issue a classical certificate *after the protocol ends* showing that the verifier has information-theoretically deleted all secrets about the statement being proved. Once a computationally bounded verifier issues a valid certificate, the proof becomes *statistically* zero-knowledge (ZK). Similarly to the case of commitments, while proofs for QMA or NP are unlikely to simultaneously satisfy *statistical soundness* and *statistical ZK*, [HMNY22b] previously introduced and built statistically sound, certified everlasting ZK proofs in the random oracle model. On the other hand, we obtain a construction in the plain model from any statistically-binding commitment.

Timed-release encryption. As another immediate application, we consider the notion of *revocable* timed-release encryption. Timed-release encryption schemes (also known as time-lock puzzles) have the property that, while ciphertexts can eventually be decrypted in some polynomial time, it takes *at least* some (parallel) $T(\lambda)$ time to do so. [Unr14] considered adding a *revocable* property to such schemes, meaning that the recipient of a ciphertext can either eventually decrypt the ciphertext in $\geq T(\lambda)$ time, or issue a certificate of deletion proving that they will *never* be able to obtain the plaintext. [Unr14] constructs semantically-secure revocable timed-release encryption assuming post-quantum timed-release encryption, but with the following drawbacks: the certificate of deletion is a *quantum state*, and the underlying scheme must either be *exponentially* hard or security must be proven in the idealized quantum random oracle model.

We can plug any post-quantum timed-release encryption scheme into our framework, and obtain revocable timed-released encryption from (polynomially-hard) post-quantum timed-released encryption, with a classical deletion certificate. Note that, when applying our main theorem, we simply instantiate the class of adversaries to be those that are $T(\lambda)$-parallel time bounded.

[6] In fact, [Por22] does not even define security in the setting where we allow the server to *first* evaluate the FHE scheme, interacting with the client in the process, and *later* delete the plaintext.

Secure computation with Everlasting Security Transfer (EST). Secure computation allows mutually distrusting participants to compute on joint private inputs while revealing no information beyond the output of the computation. The first templates for secure computation that make use of quantum information were proposed in a combination of works by Crépeau and Kilian [CK88], and Kilian [Kil88]. For a while [MS94, Yao95] it was believed that *unconditionally secure computation* could be realized based on a specific cryptographic building block: an *unconditionally secure quantum bit commitment*. Unfortunately, beliefs that unconditionally secure quantum bit commitments exist [BCJL93] were subsequently proven false [May97, LC97], and the possibility of unconditional secure computation was also ruled out [Lo97].

As such, secure computation protocols must either assume an honest majority or necessarily rely on computational hardness to achieve security against adversaries that are computationally bounded. But this may be troublesome when participants wish to compute on extremely sensitive data, such as medical or government records. In particular, consider a server that computes on highly sensitive data and keeps information from the computation around in memory forever. Such a server may be able to eventually recover data if the underlying hardness assumption breaks down in the future. In this setting, it is natural to ask: Can we use computational assumptions to design "everlasting" secure protocols against an adversary that is computationally bounded during protocol execution but becomes *computationally unbounded* after protocol execution?

Unfortunately, everlasting secure computation against *every participant in a protocol* is *also* impossible [Unr13] for most natural two-party functionalities (or multi-party functionalities against dishonest majority corruptions). For the specific case of two parties, this means that it is impossible to achieve everlasting security against *both* players, without relying on special tools like trusted/ideal hardware. Nevertheless, it is still possible to obtain everlasting (or even the stronger notion of statistical) security against one unbounded participant (see eg., [KM20] and references therein). But in *all existing protocols*, which party may be unbounded and which one must be assumed to be computationally bounded must necessarily be fixed *before protocol execution*. We ask if this is necessary. That is,

Can participants transfer everlasting security from one party to another even after a protocol has already been executed?

We show that the answer is yes, under the weak cryptographic assumption that (post-quantum) statistically-binding computationally-hiding bit commitments exist. These commitments can in turn be based on one-way functions [Nao90] or even pseudo-random quantum states [MY22, AQY22].

We illustrate our novel security property by considering it in the context of Yao's classic millionaire problem [Yao82]. Stated simply, this toy problem requires two millionaires to securely compute who is richer without revealing to each other or anyone else information about their wealth. That is, the goal is to only reveal the bit indicating whether $x_1 > x_2$ where x_1 is Alice's private input and x_2 is Bob's private input. In our extension, the millionaires would also like (certified) everlasting security against the wealthier party, while maintaining standard simulation-based security against the other party. Namely, if $x_1 > x_2$ then the protocol should satisfy certified everlasting security against Alice and standard simulation-based security against computationally bounded Bob; and if it turns out that $x_2 \geq x_1$, then the protocol should satisfy certified everlasting security against Bob and simulation-based security against bounded Alice.

More generally, our goal is to enable any one party (or a subset of parties) to dynamically and certifiably information-theoretically delete other participants' inputs, during or even after a secure

computation protocol completes. At the same time, the process of deletion should not destroy standard simulation-based security.

We build a two-party protocol that is (a) designed to be secure against computationally *unbounded Alice* and computationally *bounded Bob*. In addition, even after the protocol ends, (b) Bob has the capability to generate a proof whose validity certifies that the protocol has now become secure against *unbounded Bob* while remaining secure against *bounded Alice*. In other words, verification of the proof implies that everlasting security roles have switched: this is why we call this property *everlasting security transfer*. This implies zero-knowledge proofs for NP/QMA with certified everlasting ZK as a special case. We also extend this result to obtain *multi-party computation* where even after completion of the protocol, any arbitrary subset of parties can certifiably, information-theoretically remove information about the other party inputs from their view.

At a high level, we build these protocols by carefully combining Theorem 1.1 with additional techniques to ensure that having one party generate a certificate of deletion does not ruin standard (simulation-based, computational) security against the other party.

In what follows, we provide a detailed overview of our techniques.

1.2 Techniques

We first provide an overview of our proof of Theorem 1.1, noting that we actually prove a stronger version of this theorem, which shows certified deletion security even when the masked value

$$b' = b \oplus \bigoplus_{i:\theta_i=0} x_i$$

is left in plaintext form (i.e., is not encrypted).

Our construction and analysis include a couple of crucial differences from previous work on certified deletion. First, our analysis diverges from recent work [BI20, Por22] that relies on "generalized uncertainty relations" which provide lower bounds on the sum of entropies resulting from two incompatible measurements, and instead builds on the simple but powerful "quantum cut-and-choose" formalism of Bouman and Fehr [BF10]. Next, we make crucial use of an *unseeded* randomness extractor (the XOR function), as opposed to a seeded extractor, as used by [BI20].

Delaying the dependence on b. A key tension that must be resolved when proving a claim like Theorem 1.1 is the following: how to *information-theoretically* remove the bit b from the adversary's view, when it is initially information-theoretically *determined* by the adversary's input. Our first step towards a proof is a simple change in perspective. We will instead imagine sampling the distribution by *guessing* a uniformly random $b' \leftarrow \{0,1\}$, and initializing the adversary with $|x\rangle_\theta, b', \mathcal{Z}_\lambda(\theta)$. Then, we abort the experiment (output $\perp$) if it happens that $b' \neq b \oplus \bigoplus_{i:\theta_i=0} x_i$. Since b' was a uniformly random guess, we always abort with probability exactly $1/2$, and thus the trace distance between the $b = 0$ and $b = 1$ outputs of this experiment is at least half the trace distance between the outputs of the original experiment.[7]

Now, the bit b is only used by the experiment to determine whether or not to output $\perp$. This is not immediately helpful, since the result of this "abort decision" is of course included in the output

[7] One might be concerned that extending this argument to multi-bit messages may eventually reduce the advantage by too much, since the entire message must be guessed. However, it actually suffices to prove Theorem 1.1 for single bit messages and then use a bit-by-bit hybrid argument to obtain security for any polynomial-length message.

of the experiment. However, we can make progress by delaying this abort decision (and thus, the dependence on b) until *after* the adversary outputs (x', ρ). To do so, we will make use of a common strategy in quantum cryptographic proofs: replace the BB84 states $|x\rangle_\theta$ with halves of EPR pairs $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Let C be the register holding the "challenger's" halves of EPR pairs, and A be the register holding the other halves, which is part of the adversary's input. This switch is perfectly indistinguishable from the adversary's perspective, and it allows us to *delay* the measurement of C in the θ-basis (and thus, delay the determination of the string x and subsequent abort decision), until after the adversary outputs (x', ρ).

We still have not shown that when the deletion certificate is accepted, information about b doesn't exist in the output of the experiment. However, note that at this point it suffices to argue that $\bigoplus_{i:\theta_i=0} x_i$ is distributed like a uniformly random bit, even conditioned on the adversary's "side information" ρ (which may be entangled with C). This is because, if $\bigoplus_{i:\theta_i=0} x_i$ is uniformly random, then the outcome of the abort decision, whether $b' = b \oplus \bigoplus_{i:\theta_i=0} x_i$, is also a uniformly random bit, regardless of b.

Identifying an efficiently-checkable predicate. To prove that $\bigoplus_{i:\theta_i=0} x_i$ is uniformly random, we will need to establish that the measured bits $\{x_i\}_{i:\theta_i=0}$ contain sufficient entropy. To do this, we will need to make some claim about the structure of the state on registers $\mathsf{C}_{i:\theta_i=0}$. These registers are measured in the computational basis to produce $\{x_i\}_{i:\theta_i=0}$, so if we could claim that these registers are in a Hadamard basis state, we would be done. We won't quite be able to claim something this strong, but we don't need to. Instead, we will rely on the following claim: consider any (potentially entangled) state on systems A and B, such that the part of the state on system B is in a superposition of Hadamard basis states $|u\rangle_\times$ where each u is a vector of somewhat low Hamming weight.[8] Then, measuring B in the computational basis and computing the XOR of the resulting bits produces a bit that is *uniformly random and independent* of system A. This claim can be viewed as saying that XOR is a good (seedless) randomness extractor for the quantum source of entropy that results from measuring certain structured states in the conjugate basis. Indeed, such a claim was developed to remove the need for *seeded* randomness extraction in applications like quantum oblivious transfer [ABKK22], and it serves a similar purpose here[9].

Thus, it suffices to show that the state on registers $\mathsf{C}_{i:\theta_i=0}$ is only supported on low Hamming weight vectors in the Hadamard basis. A priori, it is not clear why this would even be true, since C, A are initialized with EPR pairs, and the adversary, who has access to A, can simply measure its halves of these EPR pairs in the computational basis. However, recall that the experiment we are interested in only outputs the adversary's final state when its certificate of deletion is valid, and moreover, a valid deletion certificate is a string x' that matches x in all the *Hadamard* basis positions. Moreover, which positions will be checked is semantically hidden from the adversary. Thus, in order to be sure that it passes the verification, an adversary should intuitively be measuring most of its registers A in the Hadamard basis.

[8] It suffices to require that the relative Hamming weight of each u is $< 1/2$.

[9] If we had tried to rely on generic properties of a seeded randomness extractor, as done in [BI20], we would still have had to deal with the fact the adversary's view includes an encryption of the seed, which is required to be *uniform and independent* of the source of entropy. Even if the challenger's state can be shown to produce a sufficient amount of min-entropy when measured in the standard basis, we cannot immediately claim that this source of entropy is perfectly independent of the seed of the extractor. Similar issues with using seeded randomness extraction in a related context are discussed by [Unr14] in their work on revocable timed-release encryption.

Reducing to semantic security. One remaining difficulty in formalizing this intuition is that if the adversary knew θ, it could decide which positions to measure in the Hadamard basis to pass the verification check, and then measure $\mathsf{A}_{i:\theta_i=0}$ in the computational basis in order to thwart the above argument from going through. And in fact, the adversary *does* have information about θ, encoded in the distribution $\mathcal{Z}_\lambda(\theta)$.

This is where the assumption that $\mathcal{A}_\lambda$ cannot distinguish between $\mathcal{Z}_\lambda(0)$ and $\mathcal{Z}_\lambda(1)$ comes into play. We interpret the condition that registers $\mathsf{C}_{i:\theta_i=0}$ must be in a superposition of low Hamming weight vectors in the Hadamard basis (or verification doesn't pass) as an efficient predicate (technically a binary projective measurement) that can be checked by a reduction to the indistinguishability of distributions $\mathcal{Z}_\lambda(\theta)$ and $\mathcal{Z}_\lambda(0^\lambda)$. Thus, this predicate must have roughly the same probability of being true when the adversary receives $\mathcal{Z}_\lambda(0^\lambda)$. But now, since θ is independent of the adversary's view, we can show *information-theoretically* that this predicate must be true with overwhelming probability. This final step reduces to a particular "quantum sample-and-estimate strategy", as defined by [BF10]. Intuitively, since θ can be sampled after the adversary makes its move, it is impossible for the adversary to guess which registers will be measured in the computational basis and which in the Hadamard basis. Thus, in order to be sure that they pass the verification test with reasonable probability, they must have measured "most" of the registers in the Hadamard basis. This completes an overview of our proof.

We note that the broad strategy of identifying an efficiently-checkable predicate which implies the *uncheckable property that some information is random and independent of the adversary's view* has been used in similar (quantum cryptographic) contexts by Gottesman [Got03] in their work on the related concept of *uncloneable* (or perhaps more accurately, *tamper-detectable*) encryption[10] and by Unruh [Unr14] in their work on revocable timed-release encryption.

Application: A variety of encryption schemes with certified deletion. For any $X \in$ {public-key encryption, attribute-based encryption, witness encryption, statistically-binding commitment, timed-release encryption}, we immediately obtain "X with certified deletion" by instantiating the distribution $\mathcal{Z}_\lambda$ with the encryption/encoding procedure for X, and additionally encrypting/encoding the bit $b \oplus \bigoplus_{i:\theta_i=0} x_i$ to ensure that semantic security holds regardless of whether the adversary deletes the ciphertext or not.

Similarly, if $\mathcal{Z}_\lambda$ is instantiated with the encryption procedure for a *fully-homomorphic* encryption scheme [Gen09, BV11, GSW13], then the scheme also allows for arbitrary homomorphic operations over the ciphertext. We also note that such a scheme can be used for blind delegation with certified deletion, allowing a weak client to outsource computations to a powerful server and subsequently verify deletion of the plaintext. In particular, a server may perform homomorphic evaluation coherently (i.e. by not performing any measurements), and return the register containing the output to the client. The client can coherently decrypt this register to obtain a classical outcome, then reverse the decryption operation and return the output register to the server. Finally, the server can use this register to reverse the evaluation operation and recover the original ciphertext. Then, the server can prove deletion of the original plaintext as above, i.e. measure the quantum state associated with this ciphertext in the Hadamard basis, and report the outcomes as their certificate.

[10] In this notion, the adversary is an eavesdropper who sits between a ciphertext generator Alice and a ciphertext receiver Bob (using a symmetric-key encryption scheme), who attempts to learn some information about the ciphertext. The guarantee is that, *either* the eavesdropper gains information-theoretically no information about the underlying plaintext, *or* Bob can detect that the ciphertext was tampered with. While this is peripherally related to our setting, [Got03] does not consider public-key encryption, and moreover Bob's detection procedure is quantum.

Application: Secure computation with Everlasting Security Transfer (EST). Recall that in building two-party computation with EST, the goal is to build protocols (a) secure against *unbounded Alice* and computationally *bounded Bob* such that, during or even after the protocol ends, (b) Bob can generate a proof whose validity certifies that the protocol has now become secure against *unbounded Bob* while remaining secure against *bounded Alice*.

Our goal is to realize two-party secure computation with EST from minimal cryptographic assumptions. We closely inspect a class of protocols for secure computation that do not a-priori have any EST guarantees, and develop techniques to equip them with EST.

In particular, we observe that a key primitive called quantum oblivious transfer (QOT) is known to unconditionally imply secure computation of *all classical (and quantum) circuits* [Kil88, CvT95, DGJ+20]. Namely, given OT with information-theoretic security, it is possible to build secure computation with everlasting (and even unconditional) security against unbounded participants. We recall that information-theoretically secure OT cannot exist in the plain model, even given quantum resources [Lo97]. However, for the case of EST, we establish a general sequential composition theorem (Theorem 5.6) which shows that oblivious transfer with EST can be plugged into the above unconditional protocols to yield secure computation protocols with EST.

Furthermore, a recent line of work [CK88, DFL+09, BF10, BCKM21, GLSV21] establishes *ideal* commitments[11] as the basis for QOT. Intuitively, these are commitments that satisfy the (standard) notion of simulation-based security against computationally bounded quantum committers and receivers. Namely, for every adversarial committer (resp., receiver) that interacts with an honest receiver (resp., committer) in the real protocol, there is a simulator that interacts with the ideal commitment functionality and generates a simulated state that is indistinguishable from the committer's (resp., receiver's) state in the real protocol. Our composition theorem (Theorem 5.6) combined with [BCKM21] also immediately shows that ideal commitments *with EST* imply QOT with EST. Thus, the problem reduces to building ideal commitments with EST.

Constructing Ideal Commitments with EST. An ideal commitment with EST satisfies statistical simulation-based security against unbounded committers, and computational simulation-based security against bounded receivers. Furthermore, after an optional delete/transfer phase succeeds, everlasting security is *transfered*: that is, then the commitment satisfies statistical (simulation-based) security against unbounded receivers, and remains computationally (simulation-based) secure against bounded committers.

To build ideal commitments with EST, we start with any commitment that satisfies standard computational hiding, and a strong form of binding: namely, simulation-based security against an unbounded malicious committer. At a high level, this means that there is an efficient extractor that can extract the input committed by an unbounded committer, thereby statistically simulating the view of the adversarial committer in its interaction with the ideal commitment functionality. We call this a *computationally-hiding statistically-efficiently-extractable* (CHSEE) commitment, and observe that prior work ([BCKM21]) builds such commitments from black-box use of any statistically-binding, computationally-hiding commitment. Our construction of ideal commitments with EST starts with CHSEE commitments, and proceeds in two steps, where the first involves new technical insights and the second follows from ideas in prior work [BCKM21].

[11] The term "ideal committment" can sometimes refer to the commitment *ideal funtionality*, but in this work we use the term ideal commitment to refer to a *real-world protocol* that can be shown to securely implement the commitment ideal functionality.

Step 1: One-Sided Ideal Commitments with EST. While CHSEE commitments satisfy simulation-based security against a malicious committer, they do not admit security transfer. Therefore, our first step is to add the EST property to CHSEE commitments, which informally additionally allows receivers to certifiably, information-theoretically, delete the committed input. We call the resulting primitive *one-sided ideal commitments with EST*. The word "one-sided" denotes that these commitments satisfy simulation-based security against any malicious committer, but are not necessarily simulation-secure against malicious receivers. Instead, these commitments semantically hide the committed bit from a malicious receiver and furthermore, support certified everlasting hiding against malicious receivers.

We observe that invoking Theorem 1.1 while instantiating $\mathcal{Z}_\lambda$ with a CHSEE commitments already helps us add the certified everlasting hiding property to any CHSEE commitment. While this ensures the desired certified everlasting security against malicious receivers, the scheme appears to become insecure against malicious committers after certified deletion!

To see why, recall that the resulting commitment is now $|x\rangle_\theta, \mathsf{Com}(\theta, b')$, where Com is a CHSEE commitment and $b' = b \oplus \bigoplus_{i:\theta_i=0} x_i$. In particular, to simulate (i.e., to extract the bit committed by) a malicious committer $\mathcal{C}^*$, a simulator must extract the bases θ and masked bit b' from the CHSEE commitment, measure the accompanying state $|\psi\rangle$ in basis θ to recover x, and then XOR the parity $\bigoplus_{i:\theta_i=0} x_i$ with b' to obtain the committed bit b. Thus, the simulator will have to first measure qubits of $|\psi\rangle$ that correspond to $\theta_i = 0$ in the computational basis to recover x_i values at these positions. If the committer makes a delete request after this point, the simulator must measure *all positions* in the Hadamard basis to generate the certificate of deletion. But consider a cheating committer that (maliciously) generates the qubit at a certain position (say $i = 1$) as a half of an EPR pair, keeping the other half to itself. Next, this committer commits to $\theta_i = 0$ (i.e., computational basis) corresponding to the index $i = 1$. The simulation strategy outlined above will first measure the first qubit of $|\psi\rangle$ in the computational basis, and then later in the Hadamard basis to generate a deletion certificate. On the other hand, an honest receiver will only ever measure this qubit in the Hadamard basis to generate a deletion certificate. This makes it easy for such a committer to distinguish simulation from an honest receiver strategy, simply by measuring its half of the EPR pair in the Hadamard basis, thereby breaking simulation security post-deletion.

To prevent this attack, we modify the scheme so that the committer $\mathcal{C}^*$ *only ever obtains* the receiver's outcomes of Hadamard basis measurements on indices where the committed $\theta_i = 1$. In particular, we make the delete phase interactive: the receiver will first commit to all measurement outcomes in Hadamard bases, $\mathcal{C}^*$ will then decommit to θ, and then finally the receiver will *only* open the committed measurement outcomes on indices i where $\theta_i = 1$. Against malicious receivers, we prove that this scheme is computationally hiding before deletion, and is certified everlasting hiding after deletion. Against a malicious committer, we prove statistical simulation-based security before deletion, and show that computational simulation-based security holds *even after deletion*.

Step 2: Ideal Commitments with EST. Next, we upgrade the one-sided ideal commitments with EST obtained above to build (full-fledged) ideal commitments with EST. Recall that the one-sided ideal commitments with EST do not satisfy simulation-based security against malicious receivers. Intuitively, simulation-based security against malicious receivers requires the existence of a simulator that interacts with a malicious receiver to produce a state in the commit phase, that can later be opened (or *equivocated*) to a bit that is only revealed to the simulator at the end of the

commit phase. We show that this property can be generically obtained (with EST) by relying on a previous compiler, namely an *equivocality compiler* from [BCKM21]. We defer additional details of this step to Section 5.3 since this essentially follows from ideas in prior work [BCKM21]. This also completes an overview of our techniques.

Roadmap. We refer the reader to Section 3 for the proof of our main theorem, Section 4 for a variety of encryption and commitment schemes with everlasting security, and Section 5 for details on building secure computation with everlasting security transfer.

1.3 Concurrent and independent work

Subsequent to the original posting of our paper on arXiv, an updated version of [Por22] was posted with some independent new results on fully-homomorphic encryption with certified deletion. The updated FHE scheme with certified deletion is shown to satisfy standard semantic security, but under a newly introduced conjecture that a particular hash function is "strong Gaussian-collapsing". Proving this conjecture based on a standard assumption such as LWE is left as an open problem in [Por22]. Thus, the FHE scheme presented in our paper remains the only scheme with certified deletion whose security is based on a standard assumption (and in addition satisfies *everlasting hiding*). On the other hand, the updated scheme of [Por22] also satisfies the property of publicly-verifiable deletion, which we do not consider in this work.

We also discuss the concurrent and independent work of Hiroka et al. [HMNY22a] that was posted subsequent to the initial posting of our work. In [HMNY22a], the authors construct public-key encryption schemes satisfying the definition of security that we use in this paper: certified everlasting security. However, their constructions are either in the *quantum random oracle model*, or require a *quantum* certificate of deletion. Thus, our construction of PKE with certified everlasting security, which is simple, in the plain model, and has a classical certificate of deletion, subsumes these results. On the other hand, [HMNY22a] introduce and construct the primitive of (bounded-collusion) *functional encryption* with certified deletion, which we do not consider in this work.

1.4 Followup Work

Finally, a recent work [BGKR22] builds on our techniques to obtain maliciously-secure delegation with certified deletion and other novel applications. They use additional tools like subspace coset states [CLLZ21] and SNARGs (which can be obtained from the Learning with Errors assumption [CJJ21]). These tools, together with our proof technique, also help them obtain the first compilers for obfuscation and unbounded collusion-resistant functional encryption with certified deletion.

2 Preliminaries

Let λ denote the security parameter. We write $\mathsf{negl}(\cdot)$ to denote any *negligible* function, which is a function f such that for every constant $c \in \mathbb{N}$ there exists $N \in \mathbb{N}$ such that for all $n > N$, $f(n) < n^{-c}$.

Given an alphabet A and string $x \in A^n$, let $h(x)$ denote the Hamming weight (number of non-zero indices) of x, and $\omega(x) := h(x)/n$ denote the *relative* Hamming weight of x. Given two strings $x, y \in \{0,1\}^n$, let $\Delta(x,y) := \omega(x \oplus y)$ denote the *relative Hamming distance* between x and y.

2.1 Quantum preliminaries

A register X is a named Hilbert space $\mathbb{C}^{2^n}$. A pure quantum state on register X is a unit vector $|\psi\rangle^{\mathsf{X}} \in \mathbb{C}^{2^n}$, and we say that $|\psi\rangle^{\mathsf{X}}$ consists of n qubits. A mixed state on register X is described by a density matrix $\rho^{\mathsf{X}} \in \mathbb{C}^{2^n \times 2^n}$, which is a positive semi-definite Hermitian operator with trace 1.

A *quantum operation* F is a completely-positive trace-preserving (CPTP) map from a register X to a register Y, which in general may have different dimensions. That is, on input a density matrix ρ^{X}, the operation F produces $F(\rho^{\mathsf{X}}) = \tau^{\mathsf{Y}}$ a mixed state on register Y. We will sometimes write a quantum operation F applied to a state on register X and resulting in a state on register Y as $\mathsf{Y} \leftarrow F(\mathsf{X})$. Note that we have left the actual mixed states on these registers implicit in this notation, and just work with the names of the registers themselves.

A *unitary* $U : \mathsf{X} \to \mathsf{X}$ is a special case of a quantum operation that satisfies $U^\dagger U = UU^\dagger = \mathbb{I}^{\mathsf{X}}$, where $\mathbb{I}^{\mathsf{X}}$ is the identity matrix on register X. A *projector* Π is a Hermitian operator such that $\Pi^2 = \Pi$, and a *projective measurement* is a collection of projectors $\{\Pi_i\}_i$ such that $\sum_i \Pi_i = \mathbb{I}$.

Let Tr denote the trace operator. For registers X, Y, the *partial trace* Tr^{Y} is the unique operation from X, Y to X such that for all $(\rho, \tau)^{\mathsf{X},\mathsf{Y}}$, $\mathsf{Tr}^{\mathsf{Y}}(\rho, \tau) = \mathsf{Tr}(\tau)\rho$. The *trace distance* between states ρ, τ, denoted $\mathsf{TD}(\rho, \tau)$ is defined as

$$\mathsf{TD}(\rho, \tau) \coloneqq \frac{1}{2}\|\rho - \tau\|_1 \coloneqq \frac{1}{2}\mathsf{Tr}\left(\sqrt{(\rho - \tau)^\dagger(\rho - \tau)}\right).$$

We will often use the fact that the trace distance between two states ρ and τ is an upper bound on the probability that any (unbounded) algorithm can distinguish ρ and τ. When clear from context, we will write $\mathsf{TD}(\mathsf{X}, \mathsf{Y})$ to refer to the trace distance between a state on register X and a state on register Y.

Lemma 2.1 (Gentle measurement [Win99])**.** *Let ρ^{X} be a quantum state and let $(\Pi, \mathbb{I} - \Pi)$ 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 state after applying $(\Pi, \mathbb{I} - \Pi)$ to ρ and post-selecting on obtaining the first outcome. Then, $\mathsf{TD}(\rho, \rho') \leq 2\sqrt{\delta}$.

We will make use of the convention that 0 denotes the computational basis $\{|0\rangle, |1\rangle\}$ and 1 denotes the Hadamard basis $\left\{\frac{|0\rangle+|1\rangle}{\sqrt{2}}, \frac{|0\rangle-|1\rangle}{\sqrt{2}}\right\}$. For a bit $r \in \{0,1\}$, we write $|r\rangle_0$ to denote r encoded in the computational basis, and $|r\rangle_1$ to denote r encoded in the Hadamard basis. For strings $x, \theta \in \{0,1\}^\lambda$, we write $|x\rangle_\theta$ to mean $|x_1\rangle_{\theta_1}, \ldots, |x_\lambda\rangle_{\theta_\lambda}$.

A non-uniform quantum polynomial-time (QPT) machine $\{\mathcal{A}_\lambda, |\psi\rangle_\lambda\}_{\lambda\in\mathbb{N}}$ is a family of polynomial-size quantum machines $\mathcal{A}_\lambda$, where each is initialized with a polynomial-size advice state $|\psi_\lambda\rangle$. Each $\mathcal{A}_\lambda$ is in general described by a CPTP map. Similar to above, when we write $\mathsf{Y} \leftarrow \mathcal{A}(\mathsf{X})$, we mean that the machine $\mathcal{A}$ takes as input a state on register X and produces as output a state on register Y, and we leave the actual descripions of these states implicit. Finally, a quantum *interactive* machine is simply a sequence of quantum operations, with designated input, output, and work registers.

2.2 The XOR extractor

We make use of a result from [ABKK22] which shows that the XOR function is a good randomness extractor from certain *quantum* sources of entropy, even given quantum side information. We include a proof here for completeness.

Imported Theorem 2.2 ([ABKK22])**.** *Let* X *be an* n*-qubit register, and consider any quantum state* $|\gamma\rangle^{\mathsf{A},\mathsf{X}}$ *that can be written as*

$$|\gamma\rangle^{\mathsf{A},\mathsf{X}} = \sum_{u:h(u)<n/2} |\psi_u\rangle^{\mathsf{A}} \otimes |u\rangle^{\mathsf{X}},$$

where $h(\cdot)$ *denotes the Hamming weight. Let* $\rho^{\mathsf{A},\mathsf{P}}$ *be the mixed state that results from measuring* X *in the Hadamard basis to produce a string* $x \in \{0,1\}^n$*, and writing* $\bigoplus_{i\in[n]} x_i$ *into a single qubit register* P*. Then it holds that*

$$\rho^{\mathsf{A},\mathsf{P}} = \mathsf{Tr}^{\mathsf{X}}(|\gamma\rangle\langle\gamma|) \otimes \left(\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|\right).$$

Proof. First, write the state on registers $\mathsf{A},\mathsf{X},\mathsf{P}$ that results from applying Hadamard to X and writing the parity, denoted by $p(x) \coloneqq \bigoplus_{i\in[n]} x_i$, to P:

$$\frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} \left(\sum_{u:h(u)<n/2} (-1)^{u\cdot x} |\psi_u\rangle^{\mathsf{A}} \right) |x\rangle^{\mathsf{X}} |p(x)\rangle^{\mathsf{P}} \coloneqq \frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} |\phi_x\rangle^{\mathsf{A}} |x\rangle^{\mathsf{X}} |p(x)\rangle^{\mathsf{P}}.$$

Then, tracing out the register X, we have that

$$\begin{aligned}
\rho^{\mathsf{A},\mathsf{P}} &= \frac{1}{2^n} \sum_{x\in\{0,1\}^n} |\phi_x\rangle |p(x)\rangle \langle p(x)| \langle\phi_x| \\
&= \frac{1}{2^n} \sum_{x:p(x)=0} |\phi_x\rangle\langle\phi_x| \otimes |0\rangle\langle 0| + \frac{1}{2^n} \sum_{x:p(x)=1} |\phi_x\rangle\langle\phi_x| \otimes |1\rangle\langle 1| \\
&= \frac{1}{2^n} \sum_{x:p(x)=0} \left(\sum_{u_1,u_2:h(u_1),h(u_2)<n/2} (-1)^{(u_1\oplus u_2)\cdot x} |\psi_{u_1}\rangle\langle\psi_{u_2}| \right) \otimes |0\rangle\langle 0| \\
&\quad + \frac{1}{2^n} \sum_{x:p(x)=1} \left(\sum_{u_1,u_2:h(u_1),h(u_2)<n/2} (-1)^{(u_1\oplus u_2)\cdot x} |\psi_{u_1}\rangle\langle\psi_{u_2}| \right) \otimes |1\rangle\langle 1| \\
&= \sum_{u_1,u_2:h(u_1),h(u_2)<n/2} |\psi_{u_1}\rangle\langle\psi_{u_2}| \otimes \left(\frac{1}{2^n} \sum_{x:p(x)=0} (-1)^{(u_1\oplus u_2)\cdot x} |0\rangle\langle 0| + \frac{1}{2^n} \sum_{x:p(x)=1} (-1)^{(u_1\oplus u_2)\cdot x} |1\rangle\langle 1| \right) \\
&= \sum_{u:h(u)<n/2} |\psi_u\rangle\langle\psi_u| \otimes \left(\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|\right) \\
&= \mathsf{Tr}^{\mathsf{X}}(|\gamma\rangle\langle\gamma|) \otimes \left(\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|\right),
\end{aligned}$$

where the 5th equality is due to the following claim, plus the observation that $u_1 \oplus u_2 \neq 1^n$ for any u_1, u_2 such that $h(u_1) < n/2$ and $h(u_2) < n/2$.

Claim 2.3. *For any $u \in \{0,1\}^n$ such that $u \notin \{0^n, 1^n\}$, it holds that*

$$\sum_{x:p(x)=0} (-1)^{u\cdot x} = \sum_{x:p(x)=1} (-1)^{u\cdot x} = 0.$$

Proof. For any such $u \notin \{0^n, 1^n\}$, define $S_0 = \{i : u_i = 0\}$ and $S_1 = \{i : u_i = 1\}$. Then, for any $y_0 \in \{0,1\}^{|S_0|}$ and $y_1 \in \{0,1\}^{|S_1|}$, define $x_{y_0,y_1} \in \{0,1\}^n$ to be the n-bit string that is equal to y_0 when restricted to indices in S_0 and equal to y_1 when restricted to indices in S_1. Then,

$$\begin{aligned}\sum_{x:p(x)=0} (-1)^{u\cdot x} &= \sum_{y_1\in\{0,1\}^{|S_1|}} \sum_{y_0\in\{0,1\}^{|S_0|}:p(x_{y_0,y_1})=0} (-1)^{u\cdot x_{y_0,y_1}} \\ &= \sum_{y_1\in\{0,1\}^{|S_1|}} 2^{|S_0|-1}(-1)^{1^{|S_1|}\cdot y_1} = 2^{|S_0|-1} \sum_{y_1\in\{0,1\}^{|S_1|}} (-1)^{p(y_1)} = 0,\end{aligned}$$

and the same sequence of equalities can be seen to hold for $x : p(x) = 1$. □

□

2.3 Sampling in a quantum population

In this section, we describe a generic framework presented in [BF10] for analyzing cut-and-choose strategies applied to quantum states.

Classical sample-and-estimate strategies. Let A be a set, and let $\mathbf{q} = (q_1, \ldots, q_n) \in A^n$ be a string of length n. We consider the problem of estimating the relative Hamming weight $\omega(\mathbf{q}_{\bar{t}})$ of a substring $\mathbf{q}_{\bar{t}}$ of $\mathbf{q}$ by only looking at the substring $\mathbf{q}_t$, where $t \subset [n]$. We consider "sample-and-estimate" strategies $\Psi = (T, f)$,[12] where T is a distribution over subsets $t \subseteq [n]$ and $f : \{(t, \mathbf{q}_t) : t \subseteq [n], \mathbf{q}_t \in A^{|t|}\} \to \mathbb{R}$ is a function that takes the subset t and the substring $\mathbf{q}_t$, and outputs an estimate for the relative Hamming weight of the remaining string. For a fixed subset t and a parameter δ, define $B_t^\delta(\Psi) \subseteq A^n$ as

$$B_t^\delta(\Psi) \coloneqq \{\mathbf{q} \in A^n : |\omega(\mathbf{q}_{\bar{t}}) - f(t, \mathbf{q}_t)| < \delta\}.$$

Then we define the *classical error probability* of strategy Ψ as follows.

Definition 2.4 (Classical error probability)**.** *The classical error probability of a sample-and-estimate strategy $\Psi = (T, f)$ is defined as the following value, paraterized by $0 < \delta < 1$:*

$$\epsilon^\delta_{\mathsf{classical}}(\Psi) \coloneqq \max_{\mathbf{q}\in A^n} \Pr_{t\leftarrow T}\left[\mathbf{q} \notin B_t^\delta(\Psi)\right].$$

[12] [BF10] consider a more general class of sample-and-estimate strategies that also make use of a random seed, but we will not need such strategies in this work.

Quantum sample-and-estimate strategies. Now, consider an n-partite quantum system on registers $\mathsf{A} = \mathsf{A}_1, \ldots, \mathsf{A}_n$, where each system has dimension d. Let $\{|a\rangle\}_{a \in A}$ be a fixed orthonormal basis for each A_i. A may be entangled with another system E, and we write the purified state on A and E as $|\psi\rangle_{\mathsf{AE}}$. We consider the problem of testing whether the state on A is close to the all-zero reference state $|0\rangle^{\mathsf{A}_1} \ldots |0\rangle^{\mathsf{A}_n}$. There is a natural way to apply any sample-and-estimate strategy $\Psi = (T, f)$ to this setting: sample $t \leftarrow T$, measure subsystems A_i for $i \in [t]$ in basis $\{|a\rangle\}_{a \in A}$ to observe $\mathbf{q}_t \in A^{|t|}$, and compute an estimate $f(t, \mathbf{q}_t)$.

In order to analyze the effect of this strategy, we first consider the mixed state on registers T, A, and E, where T holds the sampled subset t.

$$\rho^{\mathsf{T},\mathsf{A},\mathsf{E}} = \sum_t T(t) \, |t\rangle \langle t| \otimes |\psi\rangle \langle \psi| \, .$$

Next, we compare this state to an *ideal* state, parameterized by $0 < \delta < 1$, of the form

$$\widetilde{\rho}^{\mathsf{T},\mathsf{A},\mathsf{E}} = \sum_t T(t) \, |t\rangle \langle t| \otimes |\widetilde{\psi}^t\rangle \langle \widetilde{\psi}^t| \;\text{ with }\; |\widetilde{\psi}^t\rangle \in \mathsf{span}\left(B_t^\delta(\Psi)\right) \otimes \mathsf{E},$$

where

$$\mathsf{span}\left(B_t^\delta(\Psi)\right) \coloneqq \mathsf{span}\left(\{|\mathbf{q}\rangle : \mathbf{q} \in B_t^\delta(\Psi)\}\right) = \mathsf{span}\left(\{|\mathbf{q}\rangle : |\omega(\mathbf{q}_{\bar{t}}) - f(t, \mathbf{q}_t)| < \delta\}\right).$$

That is, $\widetilde{\rho}^{\mathsf{T},\mathsf{A},\mathsf{E}}$ is a state such that it holds *with certainty* that the state on registers $\mathsf{A}_{\bar{t}}, \mathsf{E}$, after having measured A_t and observed $\mathbf{q}_t$, is in a superposition of states with relative Hamming weight δ-close to $f(t, \mathbf{q}_t)$. This leads us to the definition of the *quantum error probability* of strategy Ψ.

Definition 2.5 (Quantum error probability)**.** *The quantum error probability of a sample-and-estimate strategy $\Psi = (T, f)$ is defined as the following value, parameterized by $0 < \delta < 1$:*

$$\epsilon^\delta_{\mathsf{quantum}}(\Psi) \coloneqq \max_{\mathsf{E}} \max_{|\psi\rangle^{\mathsf{A},\mathsf{E}}} \min_{\widetilde{\rho}^{\mathsf{T},\mathsf{A},\mathsf{E}}} \mathsf{TD}\left(\rho^{\mathsf{T},\mathsf{A},\mathsf{E}}, \widetilde{\rho}^{\mathsf{T},\mathsf{A},\mathsf{E}}\right),$$

where the first max is over all finite-dimensional registers E, the second max is over all states $|\psi\rangle^{\mathsf{A},\mathsf{E}}$ and the min is over all ideal states $\widetilde{\rho}^{\mathsf{T},\mathsf{A},\mathsf{E}}$ of the form described above.

Finally, we relate the classical and quantum error probabilities.

Imported Theorem 2.6 ([BF10])**.** *For any sample-and-estimate strategy Ψ and $\delta > 0$,*

$$\epsilon^\delta_{\mathsf{quantum}}(\Psi) \leq \sqrt{\epsilon^\delta_{\mathsf{classical}}(\Psi)}.$$

In this work, we will only need to analyze one simple sample-and-estimate strategy $\Psi_{\mathsf{uniform}} = (T, f)$, where T is the uniform distribution over subsets $t \subseteq [n]$, and $f(t, \mathbf{q}_t) = \omega(\mathbf{q}_t)$. That is, f receives a uniformly random subset $\mathbf{q}_t$ of $\mathbf{q}$, and outputs the relative Hamming weight of $\mathbf{q}_t$ as its guess for the relative Hamming weight of $\mathbf{q}_{\bar{t}}$. The classical error probability of this strategy can be bound using Hoeffding inequalities, which is done in [BF10, Appendix B.3], where it is shown to be bounded by $4e^{-n\delta^2/32}$ for parameter δ. Thus, we have the following corollary of Imported Theorem 2.6.

Corollary 2.7. *The quantum error probability of Ψ_{uniform} with parameter δ is $\leq 2e^{-n\delta^2/64}$.*

2.4 Quantum rewinding

We will make use of the following lemma from [Wat06].

Lemma 2.8. *Let $\mathcal{Q}$ be a quantum circuit that takes n qubits as input and outputs a classical bit b and m qubits. For an n-qubit state $|\psi\rangle$, let $p(|\psi\rangle)$ denote the probability that $b = 0$ when executing $\mathcal{Q}$ on input $|\psi\rangle$. Let $p_0, q \in (0,1)$ and $\epsilon \in (0, 1/2)$ be such that:*

- *For every n-qubit state $|\psi\rangle$, $p_0 \leq p(|\psi\rangle)$,*
- *For every n-qubit state $|\psi\rangle$, $|p(|\psi\rangle) - q| < \epsilon$,*
- *$p_0(1-p_0) \leq q(1-q)$,*

Then, there is a quantum circuit $\widehat{\mathcal{Q}}$ of size $O\left(\frac{\log(1/\epsilon)}{4 \cdot p_0(1-p_0)}|\mathcal{Q}|\right)$, taking as input n qubits, and returning as output m qubits, with the following guarantee. For an n qubit state $|\psi\rangle$, let $\mathcal{Q}_0(|\psi\rangle)$ denote the output of $\mathcal{Q}$ on input $|\psi\rangle$ conditioned on $b = 0$, and let $\widehat{\mathcal{Q}}(|\psi\rangle)$ denote the output of $\widehat{\mathcal{Q}}$ on input $|\psi\rangle$. Then, for any n-qubit state $|\psi\rangle$,

$$\mathsf{TD}\left(\mathcal{Q}_0(|\psi\rangle), \widehat{\mathcal{Q}}(|\psi\rangle)\right) \leq 4\sqrt{\epsilon}\frac{\log(1/\epsilon)}{p_0(1-p_0)}.$$

3 Main theorem

Theorem 3.1. *Let $\{\mathcal{Z}_\lambda(m)\}_{\lambda\in\mathbb{N}, m\in\{0,1\}}$ be a family of distributions over either classical bitstrings or quantum states, and let $\mathscr{A}$ be any class of adversaries[13] such that for any $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathscr{A}$, it holds that*

$$\left|\Pr\left[\mathcal{A}_\lambda\left(\mathcal{Z}_\lambda(0)\right) = 1\right] - \Pr\left[\mathcal{A}_\lambda\left(\mathcal{Z}_\lambda(1)\right) = 1\right]\right| = \mathrm{negl}(\lambda).$$

For $m \in \{0,1\}^n$, let $\mathcal{Z}_\lambda(m)$ denote the distribution $\mathcal{Z}_\lambda(m_1), \ldots, \mathcal{Z}_\lambda(m_n)$.

For any $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathscr{A}$, consider the following distribution $\left\{\widetilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(b)\right\}_{\lambda\in\mathbb{N}, b\in\{0,1\}}$ over quantum states, obtained by running $\mathcal{A}_\lambda$ as follows.

- *Sample $x, \theta \leftarrow \{0,1\}^\lambda$ and initialize $\mathcal{A}_\lambda$ with*

$$|x\rangle_\theta, b \oplus \bigoplus_{i:\theta_i=0} x_i, \mathcal{Z}_\lambda(\theta).$$

- *$\mathcal{A}_\lambda$'s output is parsed as a bitstring $x' \in \{0,1\}^\lambda$ and a residual quantum state ρ.*
- *If $x_i = x'_i$ for all i such that $\theta_i = 1$ then output ρ, and otherwise output a special symbol $\perp$.*

Then,

$$\mathsf{TD}\left(\widetilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(0), \widetilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(1)\right) = \mathrm{negl}(\lambda).$$

[13] Technically, we require that for any $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathscr{A}$, every adversary $\mathcal{B}$ with time and space complexity that is linear in λ more than that of $\mathcal{A}_\lambda$, is also in $\mathscr{A}$.

Remark 3.2. *We note that, in fact, the above theorem is true as long as x, θ are $\omega(\mathsf{log}\lambda)$ bits long.*

Proof. We define a sequence of hybrid distributions.

- $\mathsf{Hyb}_0(b)$: This is the distribution $\left\{\mathcal{Z}_\lambda^{\mathcal{A}_\lambda}(b)\right\}_{\lambda\in\mathbb{N}}$ described above.
- $\mathsf{Hyb}_1(b)$: This distribution is sampled as follows.
  - Prepare λ EPR pairs $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ on registers $(\mathsf{C}_1, \mathsf{A}_1), \ldots, (\mathsf{C}_\lambda, \mathsf{A}_\lambda)$. Define $\mathsf{C} \coloneqq \mathsf{C}_1, \ldots, \mathsf{C}_\lambda$ and $\mathsf{A} \coloneqq \mathsf{A}_1, \ldots, \mathsf{A}_\lambda$.
  - Sample $\theta \leftarrow \{0,1\}, b' \leftarrow \{0,1\}$, measure register C in basis θ to obtain $x \in \{0,1\}^\lambda$, and initialize $\mathcal{A}_\lambda$ with register A along with $b', \mathcal{Z}_\lambda(\theta)$.
  - If $b' = b \oplus \bigoplus_{i:\theta_i=0} x_i$ then proceed as in Hyb_0 and otherwise output $\bot$.
- $\mathsf{Hyb}_2(b)$: This is the same as $\mathsf{Hyb}_1(b)$ except that measurement of register C to obtain x is performed after $\mathcal{A}_\lambda$ outputs x' and ρ.

We define $\mathsf{Advt}(\mathsf{Hyb}_i) \coloneqq \mathsf{TD}\left(\mathsf{Hyb}_i(0), \mathsf{Hyb}_i(1)\right)$. Then, we have that

$$\mathsf{Advt}(\mathsf{Hyb}_1) \geq \mathsf{Advt}(\mathsf{Hyb}_0)/2,$$

which follows because $\mathsf{Hyb}_1(b)$ is identically distributed to the distribution that outputs $\bot$ with probability $1/2$ and otherwise outputs $\mathsf{Hyb}_0(b)$. Next, we have that

$$\mathsf{Advt}(\mathsf{Hyb}_2) = \mathsf{Advt}(\mathsf{Hyb}_1),$$

which follows because the register C is disjoint from the registers that $\mathcal{A}_\lambda$ operates on. Thus, it remains to show that

$$\mathsf{Advt}(\mathsf{Hyb}_2) = \mathrm{negl}(\lambda).$$

To show this, we first define the following hybrid.

- $\mathsf{Hyb}'_2(b)$: This is the same as Hyb_2 except that $\mathcal{A}_\lambda$ is initialized with register A and $b', \mathcal{Z}_\lambda(0^\lambda)$.

Now, for any $b \in \{0,1\}$, consider the state on register C immediately after $\mathcal{A}_\lambda$ outputs (x', ρ) in $\mathsf{Hyb}'_2(b)$. Define the projection

$$\Pi_{x',\theta} \coloneqq |x'_{i:\theta_i=1}\rangle_1^{\mathsf{C}_{i:\theta_i=1}} \langle x'_{i:\theta_i=1}|_1^{\mathsf{C}_{i:\theta_i=1}} \otimes \sum_{y:\Delta\left(y,x'_{i:\theta_i=0}\right)\geq 1/2} |y\rangle_1^{\mathsf{C}_{i:\theta_i=0}} \langle y|_1^{\mathsf{C}_{i:\theta_i=0}},$$

where all the basis vectors are in the Hadamard basis, denoted by $|\cdot\rangle_1$, and $\Delta(\cdot,\cdot)$ denotes relative Hamming distance. Then, let $\Pr[\Pi_{x',\theta}, \mathsf{Hyb}'_2(b)]$ be the probability that a measurement of $\left\{\Pi_{x',\theta}, \mathbb{I} - \Pi_{x',\theta}\right\}$ accepts (returns the outcome associated with $\Pi_{x',\theta}$) in $\mathsf{Hyb}'_2(b)$.

Claim 3.3. *For any $b \in \{0,1\}$, $\Pr[\Pi_{x',\theta}, \mathsf{Hyb}'_2(b)] = \mathrm{negl}(\lambda)$.*

Proof. Since θ is sampled independently of $\mathcal{A}_\lambda$'s input in this experiment, we can consider sampling it after $\mathcal{A}_\lambda$ outputs x' and ρ. Thus, we consider the following setup. The experiment is run until $\mathcal{A}_\lambda$ outputs x' and ρ, where ρ may be entangled with the challenger's register C. Then, $\theta \leftarrow \{0,1\}^\lambda$ is sampled, which defines a uniformly random subset of the registers in C to measure in the Hadamard basis. For each register C_i such that $\theta_i = 1$, it is checked if the measurement result x_i is equal to x'_i.

Now, one can use the final part of this experiment to define a sample-and-estimate strategy (Section 2.3) that is applied to the state on register $\mathsf{C} = \mathsf{C}_1, \ldots, \mathsf{C}_\lambda$. Here, the orthonormal basis each subsystem is measured in is the single-qubit Hadamard basis, and the "reference system"[14] is $|x'_1\rangle_1, \ldots, |x'_\lambda\rangle_1$. The strategy consists of sampling a uniformly random subset of indices of $[\lambda]$, defined by i such that $\theta_i = 1$, measuring $\mathsf{C}_{i:\theta_i=1}$ to obtain a string $\widetilde{x}$, and then outputting the relative Hamming weight of $\widetilde{x} \oplus x'_{i:\theta_i=1}$ as the estimate for "how close"[15] the state on register $\mathsf{C}_{i:\theta_i=0}$ is to $|x'_{i:\theta_i=0}\rangle_1$. Noting that this sample-and-estimate strategy is exactly the Ψ_{uniform} strategy described at the end of Section 2.3, we have by Corollary 2.7 that the quantum error probability of this strategy is bounded by $2e^{-\lambda\delta^2/64} = \mathrm{negl}(\lambda)$, for $\delta = 1/2$. By the definition of quantum error probability (Definition 2.5), this means that, with overwhelming probability over $\theta \leftarrow \{0,1\}^\lambda$, the state on register C is within negligible trace distance of a state in the image of $\mathbb{I} - \Pi_{x',\theta}$. Indeed, $\Pi_{x',\theta}$ is a projection onto states where the computed estimate for closeness of $\mathsf{C}_{i:\theta_i=0}$ to $|x'_{i:\theta_i=0}\rangle_1$ is 0, yet the actual state on $\mathsf{C}_{i:\theta_i=0}$ is supported on vectors with relative Hamming distance $\geq 1/2$ from $x'_{i:\theta_i=0}$. This completes the proof of the claim. □

Now we consider the corresponding event in $\mathsf{Hyb}_2(b)$, denoted $\Pr[\Pi_{x',\theta}, \mathsf{Hyb}_2(b)]$.

Claim 3.4. *For any $b \in \{0,1\}$, $\Pr[\Pi_{x',\theta}, \mathsf{Hyb}_2(b)] = \mathrm{negl}(\lambda)$.*

Proof. This follows by a direct reduction to semantic security of the distribution family $\{\mathcal{Z}_\lambda(\cdot)\}_{\lambda\in\mathbb{N}}$. The reduction samples $\theta \leftarrow \{0,1\}^\lambda$, sends θ to its challenger, and receives either $\mathcal{Z}_\lambda(\theta)$ or $\mathcal{Z}_\lambda(0^\lambda)$ from its challenger. Then, it prepares λ EPR pairs, runs $\mathcal{A}_\lambda$ on halves of the EPR pairs along with the output of the distribution it received from its challenger, and b' for a random $b' \leftarrow \{0,1\}$. After $\mathcal{A}_\lambda$ outputs (x', ρ), it measures $\{\Pi_{x',\theta}, \mathbb{I} - \Pi_{x',\theta}\}$ on the other halves of the EPR pairs. Note that the complexity of this reduction is equal to the complexity of $\mathcal{A}$ plus an extra λ qubits of space and an extra linear time operation, so it is still in $\mathscr{A}$. If $\Pr[\Pi_{x',\theta}, \mathsf{Hyb}_2(b)]$ is non-negligible this can be used to distinguish $\mathcal{Z}_\lambda(\theta)$ from $\mathcal{Z}_\lambda(0^\lambda)$, due to Claim 3.3. □

Finally, we can show the following claim, which completes the proof.

Claim 3.5. $\mathsf{Advt}(\mathsf{Hyb}_2) = \mathrm{negl}(\lambda)$.

Proof. First, we note that for any $b \in \{0,1\}$, the global state of $\mathsf{Hyb}_2(b)$ immediately after $\mathcal{A}_\lambda$ outputs x' is within negligible trace distance of a state $\tau^{\mathsf{C},\mathsf{A}'}_{\mathsf{Ideal}}$ in the image of $\mathbb{I} - \Pi_{x',\theta}$, where A' is $\mathcal{A}_\lambda$'s quantum output register. This follows immediately from Claim 3.4 and Gentle Measurement (Lemma 2.1). Now, consider the measurement of $\tau^{\mathsf{C},\mathsf{A}'}_{\mathsf{Ideal}}$ on $\mathsf{C}_{i:\theta_i=1}$ that determines whether the experiment outputs $\perp$. That is, the procedure that measures $\mathsf{C}_{i:\theta_i=1}$ in the Hadamard basis and checks if the resulting string is equal to $x'_{i:\theta_i=1}$. There are two options.

- If the measurement fails, then the experiment outputs $\perp$, independent of whether $b = 0$ or $b = 1$, so there is 0 advantage in this case.

[14] Section 2.3 describes an all-zeros reference system, but it is easy to see that this generalizes to any fixed string.
[15] The formal definition of closeness is described in Section 2.3.

- If the measurement succeeds, then we know that the state on registers $\mathsf{C}_{i:\theta_i=0}$ is only supported on vectors $|y\rangle_1$ such that $\Delta(y, x'_{i:\theta_i=0}) < 1/2$, since $\tau^{\mathsf{C},\mathsf{A}'}_{\mathsf{Ideal}}$ was in the image of $\mathbb{I} - \Pi_{x',\theta}$. These registers are then measured in the computational basis to produce $\{x_i\}_{i:\theta_i=0}$, and the experiment outputs $\perp$ if $\bigoplus_{i:\theta_i=0} x_i \neq b' \oplus b$ and otherwise outputs the state on register A'. Note that (i) this decision is the *only* part of the experiment that depends on b, and (ii) by Imported Theorem 2.2, the bit $\bigoplus_{i:\theta_i=0} x_i$ is *uniformly random and independent* of the register A', which is disjoint (but possibly entangled with) C. Thus, there is also 0 advantage in this case.

  In slightly more detail, Imported Theorem 2.2 says that making a Hadamard basis measurement of a register that is in a superposition of computational basis vectors with relative Hamming weight $< 1/2$ will produce a set of bits $\{x_i\}_{i:\theta_i=0}$ such that $\bigoplus_{i:\theta_i=0} x_i$ is a uniformly random bit, even given potentially entangled quantum side information. We can apply this lemma to our system on $\mathsf{C}_{i:\theta_i=0}, \mathsf{A}'$ by considering a change of basis that maps $|x'_{i:\theta_i=0}\rangle_1 \to |0\rangle_0$. That is, the change of basis first applies Hadamard, and then an XOR with the fixed string $x'_{i:\theta_i=0}$. Applying such a change of basis maps $\mathsf{C}_{i:\theta_i=0}$ to a state that is supported on vectors $|y\rangle_0$ such that $\omega(y) < 1/2$, and we want to claim that a *Hadamard* basis measurement of the resulting state produces $\{x_i\}_{i:\theta_i=0}$ such that $\bigoplus_{i:\theta_i=0} x_i$ is uniformly random and independent of A. This is exactly the statement of Imported Theorem 2.2.

This completes the proof, since we have shown that there exists a single distribution, defined by $\tau^{\mathsf{C},\mathsf{A}'}_{\mathsf{Ideal}}$, that is negligibly close to both $\mathsf{Hyb}_2(0)$ and $\mathsf{Hyb}_2(1)$. □

□

4 Cryptography with Certified Everlasting Security

4.1 Secret sharing

We give a simple construction of a 2-out-of-2 secret sharing scheme where there exists a designated party that the dealer can ask to produce a certificate of deletion of their share. If this certificate verifies, then the underlying plaintext is information theoretically deleted, even given the other share.

Definition. First, we augment the standard syntax of secret sharing to include a deletion algorithm Del and a verification algorithm Ver. Formally, consider a secret sharing scheme $\mathsf{CDSS} = (\mathsf{Share}, \mathsf{Rec}, \mathsf{Del}, \mathsf{Ver})$ with the following syntax.

- $\mathsf{Share}(m) \to (s_1, s_2, \mathsf{vk})$ is a quantum algorithm that takes as input a classical message m, and outputs a quantum share s_1, a classical share s_2 and a (potentially quantum) verification key vk.
- $\mathsf{Rec}(s_1, s_2) \to \{m, \perp\}$ is a quantum algorithm that takes as input two shares and outputs either a message m or a $\perp$ symbol.
- $\mathsf{Del}(s_1) \to \mathsf{cert}$ is a quantum algorithm that takes as input a quantum share s_1 and outputs a (potentially quantum) deletion certificate cert.

- $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert}) \to \{\top, \bot\}$ is a (potentially quantum) algorithm that takes as input a (potentially quantum) verification key vk and a (potentially quantum) deletion certificate cert and outputs either $\top$ or $\bot$.

We say that $\mathsf{CD\text{-}SS}$ satisfies correctness of deletion if the following holds.

Definition 4.1 (Correctness of deletion)**.** $\mathsf{CD\text{-}SS} = (\mathsf{Share}, \mathsf{Rec}, \mathsf{Del}, \mathsf{Ver})$ *satisfies* correctness of deletion *if for any m, it holds with $1 - \mathrm{negl}(\lambda)$ probability over $(s_1, s_2, \mathsf{vk}) \leftarrow \mathsf{Share}(m), \mathsf{cert} \leftarrow \mathsf{Del}(s_1), \mu \leftarrow \mathsf{Ver}(\mathsf{vk}, \mathsf{cert})$ that $\mu = \top$.*

Next, we define certified deletion security for a secret sharing scheme.

Definition 4.2 (Certified deletion security)**.** *Let $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ denote a computational no-signaling adversary and b denote a classical bit. Consider experiment $\mathsf{EV\text{-}EXP}^{\mathcal{A}}_{\lambda}(b)$ which describes everlasting security given a deletion certificate, and is defined as follows.*

- *Sample $(s_1, s_2, \mathsf{vk}) \leftarrow \mathsf{Share}(b)$.*
- *Initialize $\mathcal{A}_\lambda$ with (s_1, s_2).*
- *Parse $\mathcal{A}_\lambda$'s output as a deletion certificate cert and a left-over quantum state ρ.*
- *If $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert}) = \top$ then output (ρ, s_2), and otherwise output $\bot$.*

Then $\mathsf{CD\text{-}SS} = (\mathsf{Share}, \mathsf{Rec}, \mathsf{Del}, \mathsf{Ver})$ satisfies certified deletion security *if for any computational* no-signaling *adversary $\mathcal{A}$, it holds that*

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}^{\mathcal{A}}_{\lambda}(0), \mathsf{EV\text{-}EXP}^{\mathcal{A}}_{\lambda}(1)\right) = \mathrm{negl}(\lambda),$$

Corollary 4.3. *The scheme $\mathsf{CD\text{-}SS} = (\mathsf{Share}, \mathsf{Rec}, \mathsf{Del}, \mathsf{Ver})$ defined as follows is a secret sharing scheme with certified deletion.*

- $\mathsf{Share}(m)$: *sample $x, \theta \leftarrow \{0,1\}^\lambda$ and output*

$$s_1 := |x\rangle_\theta, s_2 := \left(\theta, b \oplus \bigoplus_{i:\theta_i=0} x_i\right), \quad \mathsf{vk} := (x, \theta).$$

- $\mathsf{Rec}(s_1, s_2)$: *parse $s_1 := |x\rangle_\theta, s_2 := (\theta, b')$, measure $|x\rangle_\theta$ in the θ-basis to obtain x, and output $b = b' \oplus \bigoplus_{i:\theta_i=0} x_i$.*
- $\mathsf{Del}(s_1)$: *parse $s_1 := |x\rangle_\theta$ and measure $|x\rangle_\theta$ in the Hadamard basis to obtain a string x', and output $\mathsf{cert} := x'$.*
- $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert})$: *parse vk as (x, θ) and cert as x' and output $\top$ if and only if $x_i = x'_i$ for all i such that $\theta_i = 1$.*

Proof. Correctness of deletion follows immediately from the description of the scheme. Certified deletion security, i.e.

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}^{\mathcal{A}}_{\lambda}(0), \mathsf{EV\text{-}EXP}^{\mathcal{A}}_{\lambda}(1)\right) = \mathrm{negl}(\lambda)$$

follows by following the proof strategy of Theorem 3.1. This setting is slightly different than the setting considered in the proof of Theorem 3.1 since here we consider unbounded $\mathcal{A}_\lambda$ that are not given access to θ while Theorem 3.1 considers bounded $\mathcal{A}_\lambda$ that are given access to an encryption of θ. However, the proof is almost identical, defining hybrids as follows.

$\mathsf{Hyb}_0(b)$: This is the distribution $\left\{\mathsf{EV\text{-}EXP}_\lambda^{\mathcal{A}_\lambda}(b)\right\}_{\lambda\in\mathbb{N}}$ described above.

$\mathsf{Hyb}_1(b)$: This distribution is sampled as follows.

- Prepare λ EPR pairs $\frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$ on registers $(\mathsf{C}_1,\mathsf{A}_1),\ldots,(\mathsf{C}_\lambda,\mathsf{A}_\lambda)$. Define $\mathsf{C}\coloneqq\mathsf{C}_1,\ldots,\mathsf{C}_\lambda$ and $\mathsf{A}\coloneqq\mathsf{A}_1,\ldots,\mathsf{A}_\lambda$.
- Sample $\theta\leftarrow\{0,1\},b'\leftarrow\{0,1\}$, measure register C in basis θ to obtain $x\in\{0,1\}^\lambda$, and initialize $\mathcal{A}_\lambda$ with register A.
- If $b'=b\oplus\bigoplus_{i:\theta_i=0}x_i$ then proceed as in Hyb_0 and otherwise output $\bot$.

$\mathsf{Hyb}_2(b)$: This is the same as $\mathsf{Hyb}_1(b)$ except that measurement of register C to obtain x is performed after $\mathcal{A}_\lambda$ outputs x' and ρ.

Indistinguishability between these hybrids closely follows the proof of Theorem 3.1. The key difference is that $\mathsf{Hyb}_2'(b)$ is identical to $\mathsf{Hyb}_2(b)$ except that s_2 is set to $(b',0^\lambda)$. Then, $\Pr[\Pi_{x',\theta},\mathsf{Hyb}_2'(b)]=\mathrm{negl}(\lambda)$ follows identically to the proof in Theorem 3.1, whereas $\Pr[\Pi_{x',\theta},\mathsf{Hyb}_2(b)]=\mathrm{negl}(\lambda)$ follows because the view of $\mathcal{A}_\lambda$ is identical in both hybrids. The final claim, that $\mathsf{Advt}(\mathsf{Hyb}_2)=\mathrm{negl}(\lambda)$ follows identically to the proof in Theorem 3.1. □

Remark 4.4 (One-time pad encryption)**.** *We observe that the above proof, which considers unbounded $\mathcal{A}_\lambda$ who don't have access to θ until after they produce a valid deletion certificate, can also be used to establish the security of a simple one-time pad encryption scheme with certified deletion. The encryption of a bit b would be the state $|x\rangle_\theta$ together with a one-time pad encryption $k\oplus b\oplus\bigoplus_{i:\theta_i=0}x_i$ with key $k\leftarrow\{0,1\}$. The secret key would be (k,θ). Semantic security follows from the one-time pad, while certified deletion security follows from the above secret-sharing proof. This somewhat simplifies the construction of one-time pad encryption with certified deletion of [BI20], who required a seeded extractor.*

4.2 Public-key encryption

In this section, we define and construct post-quantum public-key encryption with certified deletion for classical messages, assuming the existence of post-quantum public-key encryption for classical messages.

Public-Key encryption with certified deletion. First, we augment the standard syntax to include a deletion algorithm Del and a verification algorithm Ver. Formally, consider a public-key encryption scheme $\mathsf{CD\text{-}PKE}=(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec},\mathsf{Del},\mathsf{Ver})$ with syntax

- $\mathsf{Gen}(1^\lambda)\to(\mathsf{pk},\mathsf{sk})$ is a classical algorithm that takes as input the security parameter and outputs a public key pk and secret key sk.
- $\mathsf{Enc}(\mathsf{pk},m)\to(\mathsf{ct},\mathsf{vk})$ is a quantum algorithm that takes as input the public key pk and a message m, and outputs a (potentially quantum) verification key vk and a quantum ciphertext ct.
- $\mathsf{Dec}(\mathsf{sk},\mathsf{ct})\to\{m,\bot\}$ is a quantum algorithm that takes as input the secret key sk and a quantum ciphertext ct and outputs either a message m or a $\bot$ symbol.

- $\mathsf{Del}(\mathsf{ct}) \to \mathsf{cert}$ is a quantum algorithm that takes as input a quantum ciphertext ct and outputs a (potentially quantum) deletion certificate cert.
- $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert}) \to \{\top, \bot\}$ is a (potentially quantum) algorithm that takes as input a (potentially quantum) verification key vk and a (potentially quantum) deletion certificate cert and outputs either $\top$ or $\bot$.

We say that $\mathsf{CD\text{-}PKE}$ satisfies correctness of deletion if the following holds.

Definition 4.5 (Correctness of deletion)**.** $\mathsf{CD\text{-}PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Ver})$ *satisfies* correctness of deletion *if for any m, it holds with $1 - \mathrm{negl}(\lambda)$ probability over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda), (\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(\mathsf{pk}, m), \mathsf{cert} \leftarrow \mathsf{Del}(\mathsf{ct}), \mu \leftarrow \mathsf{Ver}(\mathsf{vk}, \mathsf{cert})$ that $\mu = \top$.*

Next, we define certified deletion security. Our definition has multiple parts, which we motivate as follows. The first experiment is the everlasting security experiment, which requires that conditioned on the (computationally bounded) adversary producing a valid deletion certificate, their left-over state is information-theoretically independent of b. However, we still want to obtain meaningful guarantees against adversaries that do not produce a valid deletion certificate. That is, we hope for standard semantic security against arbitrarily malicious but computationally bounded adversaries. Since such an adversary can query the ciphertext generator with an arbitrarily computed deletion certificate, we should include this potential interaction in the definition, and require that the response from the ciphertext generator still does not leak any information about b.[16] Note that, in our constructions, the verification key vk is actually completely independent of the plaintext b, and thus for our schemes this property follows automatically from semantic security.

Definition 4.6 (Certified deletion security)**.** $\mathsf{CD\text{-}PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Ver})$ *satisfies* certified deletion security *if for any non-uniform QPT adversary $\mathcal{A} = \{\mathcal{A}_\lambda, |\psi\rangle_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}_\lambda^{\mathcal{A}}(0), \mathsf{EV\text{-}EXP}_\lambda^{\mathcal{A}}(1)\right) = \mathrm{negl}(\lambda),$$

and

$$\left|\Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{A}}(0) = 1\right] - \Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{A}}(1) = 1\right]\right| = \mathrm{negl}(\lambda),$$

where the experiment $\mathsf{EV\text{-}EXP}_\lambda^{\mathcal{A}}(b)$ considers everlasting security given a deletion certificate, and is defined as follows.

- *Sample $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(\mathsf{pk}, b)$.*
- *Initialize $\mathcal{A}_\lambda(|\psi_\lambda\rangle)$ with pk and ct.*
- *Parse $\mathcal{A}_\lambda$'s output as a deletion certificate cert and a left-over quantum state ρ.*
- *If $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert}) = \top$ then output ρ, and otherwise output $\bot$.*

and the experiment $\mathsf{C\text{-}EXP}_\lambda^{\mathcal{A}}(b)$ is a strengthening of semantic security, defined as follows.

[16] One might expect that the everlasting security definition described above already captures this property, since whether the certificate accepts or rejects is included in the output of the experiment. However, this experiment does not include the output of the adversary in the case that the certificate is rejected. So we still need to capture the fact that the *joint* distribution of the final adversarial state and the bit indicating whether the verification passes semantically hides b.

- *Sample* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ *and* $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(\mathsf{pk}, b)$.
- *Initialize* $\mathcal{A}_\lambda(|\psi_\lambda\rangle)$ *with* pk *and* ct.
- *Parse* $\mathcal{A}_\lambda$*'s output as a deletion certificate* cert *and a left-over quantum state* ρ.
- *Output* $\mathcal{A}_\lambda$ $(\rho, \mathsf{Ver}(\mathsf{vk}, \mathsf{cert}))$.

Now we can formally define the notion of public-key encryption with certified deletion.

Definition 4.7 (Public-key encryption with certified deletion)**.** $\mathsf{CD\text{-}PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Ver})$ *is a secure* public-key encryption scheme with certified deletion *if it satisfies (i) correctness of deletion (Definition 4.5), and (ii) certified deletion security (Definition 4.6).*

Then, we have the following corollary of Theorem 3.1.

Corollary 4.8. *Given any post-quantum semantically-secure public-key encryption scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, the scheme* $\mathsf{CD\text{-}PKE} = (\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}', \mathsf{Del}, \mathsf{Ver})$ *defined as follows is a public-key encryption scheme with certified deletion.*

- $\mathsf{Enc}'(\mathsf{pk}, m)$: *sample* $x, \theta \leftarrow \{0,1\}^\lambda$ *and output*

$$\mathsf{ct} := \left(|x\rangle_\theta, \mathsf{Enc}\left(\mathsf{pk}, \left(\theta, b \oplus \bigoplus_{i:\theta_i=0} x_i\right)\right)\right), \quad \mathsf{vk} := (x, \theta).$$

- $\mathsf{Dec}'(\mathsf{sk}, \mathsf{ct})$: *parse* $\mathsf{ct} := (|x\rangle_\theta, \mathsf{ct}')$*, compute* $(\theta, b') \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}')$*, measure* $|x\rangle_\theta$ *in the* θ*-basis to obtain* x*, and output* $b = b' \oplus \bigoplus_{i:\theta_i=0} x_i$.
- $\mathsf{Del}(\mathsf{ct})$: *parse* $\mathsf{ct} := (|x\rangle_\theta, \mathsf{ct}')$ *and measure* $|x\rangle_\theta$ *in the Hadamard basis to obtain a string* x'*, and output* $\mathsf{cert} := x'$.
- $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert})$: *parse* vk *as* (x, θ) *and* cert *as* x' *and output* $\top$ *if and only if* $x_i = x'_i$ *for all* i *such that* $\theta_i = 1$.

Proof. Correctness of deletion follows immediately from the description of the scheme. For certified deletion security, we consider the following:

- First, we observe that

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(0), \mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(1)\right) = \mathrm{negl}(\lambda)$$

  follows from Theorem 3.1 and the semantic security of PKE by setting the distribution $\mathcal{Z}_\lambda(b)$ to sample $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, b)$ and output $(\mathsf{pk}, \mathsf{ct})$, and setting the class of adversaries $\mathscr{A}$ to be all non-uniform families of QPT adversaries $\{\mathcal{A}_\lambda, |\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$.

- Next, we observe that

$$\left| \Pr\left[\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(0) = 1\right] - \Pr\left[\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(1) = 1\right] \right| = \mathrm{negl}(\lambda)$$

  follows from the fact that the encryption scheme remains (computationally) semantically secure even when the adversary is given the verification key x corresponding to the challenge ciphertext, since the bit b remains encrypted with Enc.

This completes our proof. □

The notion of certified deletion security can be naturally generalized to consider multi-bit messages, as follows.

Definition 4.9 (Certified deletion security for multi-bit messages)**.** $\mathsf{CD\text{-}PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Ver})$ *satisfies* certified deletion security *if for any non-uniform QPT adversary* $\mathcal{A} = \{\mathcal{A}_\lambda, |\psi\rangle_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}_\lambda^{\mathcal{A}}(0), \mathsf{EV\text{-}EXP}_\lambda^{\mathcal{A}}(1)\right) = \mathrm{negl}(\lambda),$$

and

$$\left|\Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{A}}(0) = 1\right] - \Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{A}}(1) = 1\right]\right| = \mathrm{negl}(\lambda),$$

where the experiment $\mathsf{EV\text{-}EXP}_\lambda^{\mathcal{A}}(b)$ *considers everlasting security given a deletion certificate, and is defined as follows.*

- *Sample* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$*. Initialize* $\mathcal{A}_\lambda(|\psi_\lambda\rangle)$ *with* pk *and parse its output as* (m_0, m_1)*.*
- *Sample* $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$*.*
- *Run* $\mathcal{A}_\lambda$ *on input* ct *and parse* $\mathcal{A}_\lambda$*'s output as a deletion certificate* cert*, and a left-over quantum state* ρ*.*
- *If* $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert}) = \top$ *then output* ρ*, and otherwise output* $\bot$*.*

and the experiment $\mathsf{C\text{-}EXP}_\lambda^{\mathcal{A}}(b)$ *is a strengthening of semantic security, defined as follows.*

- *Sample* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$*. Initialize* $\mathcal{A}_\lambda(|\psi_\lambda\rangle)$ *with* pk *and parse its output as* (m_0, m_1)*.*
- *Sample* $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$*.*
- *Run* $\mathcal{A}_\lambda$ *on input* ct *and parse* $\mathcal{A}_\lambda$*'s output as a deletion certificate* cert*, and a left-over quantum state* ρ*.*
- *Output* $\mathcal{A}_\lambda\left(\rho, \mathsf{Ver}(\mathsf{vk}, \mathsf{cert})\right)$*.*

A folklore method converts any public-key bit encryption scheme to a public-key string encryption scheme, by separately encrypting each bit in the underlying string one-by-one and appending all resulting ciphertexts. Semantic security of the resulting public-key encryption scheme follows by a hybrid argument, where one considers intermediate hybrid experiments that only modify one bit of the underlying plaintext at a time. We observe that the same transformation from bit encryption to string encryption also preserves certified deletion security, and this follows by a similar hybrid argument. That is, as long as the encryption scheme for bits satisfies certified deletion security for single-bit messages per Definition 4.6, the resulting scheme for multi-bit messages satisfies certified deletion security according to Definition 4.9.

Attribute-based encryption with certified deletion. We observe that if the underlying scheme PKE is an *attribute-based encryption* scheme, then the scheme with certified deletion that results from the above compiler inherits these properties. Thus, we obtain an attribute-based encryption scheme with certified deletion, assuming any standard (post-quantum) attribute-based encryption. The previous work of [HMNY21] also constructs an attribute-based encryption scheme with certified deletion, but under the assumption of (post-quantum) indistinguishability obfuscation. We formalize our construction below.

We first describe the syntax of an attribute-based encryption scheme with certified deletion $\mathsf{CD\text{-}ABE} = (\mathsf{Gen}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Ver})$. This augments the syntax of an ABE scheme $\mathsf{ABE} = (\mathsf{Gen}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ by adding the Del and Ver algorithms. Let $p = p(\lambda)$ denote a polynomial.

- $\mathsf{Gen}(1^\lambda) \to (\mathsf{pk}, \mathsf{msk})$ is a classical algorithm that takes as input the security parameter and outputs a public key pk and master secret key msk.
- $\mathsf{KeyGen}(\mathsf{msk}, P) \to \mathsf{sk}_P$ is a classical key generation algorithm that on input the master secret key and a predicate $P : \{0,1\}^{p(\lambda)} \to \{0,1\}$, outputs a secret key sk_P.
- $\mathsf{Enc}(\mathsf{pk}, X, m) \to (\mathsf{ct}_X, \mathsf{vk})$ is a quantum algorithm that on input a message m and an attribute X outputs a (potentially quantum) verification key vk and quantum ciphertext ct_X.
- $\mathsf{Dec}(\mathsf{sk}_P, \mathsf{ct}_X) \to \{m', \bot\}$ on input a secret key sk_P and a quantum ciphertext ct_X outputs either a message m' or a $\bot$ symbol.
- $\mathsf{Del}(\mathsf{ct}) \to \mathsf{cert}$ is a quantum algorithm that takes as input a quantum ciphertext ct and outputs a (potentially quantum) deletion certificate cert.
- $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert}) \to \{\top, \bot\}$ is a (potentially quantum) algorithm that takes as input a (potentially quantum) verification key vk and a (potentially quantum) deletion certificate cert and outputs either $\top$ or $\bot$.

Correctness of decryption for $\mathsf{CD\text{-}ABE}$ is the same as that for ABE. We define correctness of deletion, and certified deletion security for $\mathsf{CD\text{-}ABE}$ below.

Definition 4.10 (Correctness of deletion)**.** $\mathsf{CD\text{-}ABE} = (\mathsf{Gen}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Ver})$ *satisfies* correctness of deletion *if for any m, X, it holds with $1 - \mathrm{negl}(\lambda)$ probability over $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Gen}(1^\lambda), (\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(\mathsf{pk}, X, m), \mathsf{cert} \leftarrow \mathsf{Del}(\mathsf{ct}), \mu \leftarrow \mathsf{Ver}(\mathsf{vk}, \mathsf{cert})$ that $\mu = \top$.*

Definition 4.11 (Certified deletion security)**.** $\mathsf{CD\text{-}ABE} = (\mathsf{Gen}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Ver})$ *satisfies* certified deletion security *if for any non-uniform QPT adversary $\mathcal{A} = \{\mathcal{A}_\lambda, |\psi\rangle_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(0), \mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(1)\right) = \mathrm{negl}(\lambda),$$

and

$$\left|\Pr\left[\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(0) = 1\right] - \Pr\left[\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(1) = 1\right]\right| = \mathrm{negl}(\lambda),$$

where the experiments $\mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(b)$ and $\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(b)$ are defined as follows. Both experiments take an input b, and interact with $\mathcal{A}$ as follows.

- *Sample $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and initialize $\mathcal{A}_\lambda(|\psi_\lambda\rangle)$ with pk.*

- *Set* $i = 1$.
- *If* $\mathcal{A}_\lambda$ *outputs a key query* P_i*, return* $\mathsf{sk}_{P_i} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, P_i)$ *to* $\mathcal{A}_\lambda$ *and set* $i = i + 1$*. This process can be repeated polynomially many times.*
- *If* $\mathcal{A}_\lambda$ *outputs an attribute* X^* *and a pair of messages* (m_0, m_1) *where* $P_i(X^*) = 0$ *for all predicates* P_i *queried so far, then compute* $(\mathsf{vk}, \mathsf{ct}) = \mathsf{Enc}(\mathsf{pk}, X^*, m_b)$ *and return* ct *to* $\mathcal{A}_\lambda$*, else exit and output* $\bot$*.*
- *If* $\mathcal{A}_\lambda$ *outputs a key query* P_i *such that* $P_i(X^*) = 0$*, return* $\mathsf{sk}_{P_i} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, P_i)$ *to* $\mathcal{A}_\lambda$ *(otherwise return* $\bot$*) and set* $i = i+1$*. This process can be repeated polynomially many times.*
- *Parse* $\mathcal{A}_\lambda$*'s output as a deletion certificate* cert *and a left-over quantum state* ρ*.*
- *If* $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert}) = \top$ *then* $\mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(b)$ *outputs* ρ*, and otherwise* $\mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(b)$ *outputs* $\bot$*, and ends.*
- $\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(b)$ *sends the output* $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert})$ *to* $\mathcal{A}_\lambda$*. Again, upto polynomially many times,* $\mathcal{A}_\lambda$ *sends key queries* P_i*. For each* i*, if* $P_i(X^*) = 0$*, return* $\mathsf{sk}_{P_i} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, P_i)$ *to* $\mathcal{A}_\lambda$ *(otherwise return* $\bot$*) and set* $i = i + 1$*. Finally,* $\mathcal{A}_\lambda$ *generates an output bit, which is set to be the output of* $\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(b)$*.*

Corollary 4.12. *Given any post-quantum attribute-based encryption scheme* $\mathsf{ABE} = (\mathsf{Gen}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$*, the scheme* $\mathsf{CD\text{-}ABE} = (\mathsf{Gen}, \mathsf{KeyGen}, \mathsf{Enc}', \mathsf{Dec}', \mathsf{Del}, \mathsf{Ver})$ *defined as follows is an attribute-based encryption scheme with certified deletion.*

- $\mathsf{Enc}'(\mathsf{pk}, X, m)$: *sample* $x, \theta \leftarrow \{0,1\}^\lambda$ *and output*

$$\mathsf{ct} := \left(|x\rangle_\theta, \mathsf{Enc}\left(\mathsf{pk}, X, \left(\theta, b \oplus \bigoplus_{i:\theta_i=0} x_i\right)\right)\right), \quad \mathsf{vk} := (x, \theta).$$

- $\mathsf{Dec}'(\mathsf{sk}_P, \mathsf{ct})$: *parse* $\mathsf{ct} := (|x\rangle_\theta, \mathsf{ct}')$*, compute* $(\theta, b') \leftarrow \mathsf{Dec}(\mathsf{sk}_P, \mathsf{ct}')$*, measure* $|x\rangle_\theta$ *in the* θ*-basis to obtain* x*, and output* $b = b' \oplus \bigoplus_{i:\theta_i=0} x_i$*.*
- $\mathsf{Del}(\mathsf{ct})$: *parse* $\mathsf{ct} := (|x\rangle_\theta, \mathsf{ct}')$ *and measure* $|x\rangle_\theta$ *in the Hadamard basis to obtain a string* x'*, and output* $\mathsf{cert} := x'$*.*
- $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert})$: *parse* vk *as* (x, θ) *and* cert *as* x' *and output* $\top$ *if and only if* $x_i = x'_i$ *for all* i *such that* $\theta_i = 1$*.*

Proof. Correctness of decryption and deletion follow from the description of the scheme. For certified deletion security, we consider the following:

- First, we observe that

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(0), \mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(1)\right) = \mathrm{negl}(\lambda)$$

  follows from Theorem 3.1 and the semantic security of ABE. To see this, we imagine splitting $\mathcal{A}_\lambda$ into two parts: $\mathcal{A}_{\lambda,0}$ which interacts in the ABE security game until it obtains its challenge ciphertext and all the keys sk_{P_i} that it wants, and $\mathcal{A}_{\lambda,1}$ which takes the final state of $\mathcal{A}_{\lambda,0}$ and

produces a deletion certificate cert and final state ρ. We set the distribution $\mathcal{Z}_\lambda(b)$ to run the $\mathsf{EV\text{-}EXP}_\lambda^{\mathcal{A}_\lambda}(b)$ game with $\mathcal{A}_{\lambda,0}(|\psi_\lambda\rangle)$, and output $\mathcal{A}_{\lambda,0}$'s final state. Then, we set the class of adversaries $\mathscr{A}$ to include all $\mathcal{A}_{\lambda,1}$, which is the class of all uniform families of QPT adversaries. By the semantic security of ABE, $\mathcal{A}_{\lambda,1}$ cannot distinguish between $\mathcal{Z}_\lambda(0)$ and $\mathcal{Z}_\lambda(1)$, and thus the guarantees of Theorem 3.1 apply.

- Next, we observe that

$$\left|\Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{A}}(0)=1\right]-\Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{A}}(1)=1\right]\right|=\mathrm{negl}(\lambda)$$

follows from the fact that the encryption scheme remains semantically secure even when the adversary is given the verification key corresponding to the challenge ciphertext.

This completes our proof. □

Remark 4.13. *Similarly to the setting of public-key encryption, single-bit certified deletion security for ABE implies multi-bit certified deletion security.*

Relation with [HMNY21]'s definitions. The definition of certified deletion security for public-key (resp., attribute-based) encryption in [HMNY21] is different than our definition in two primary respects: (1) it only considers *computationally-bounded* adversaries even after the deletion certificate is computed, and (2) explicitly gives the adversary the secret key sk after the deletion certificate is computed.

Our definition allows the adversary to be unbounded after deletion, which gives a strong *everlasting security* property. Indeed, in Appendix A, we show that our definition implies [HMNY21]'s definition for public-key (attribute-based) encryption schemes. To see this, we consider any adversary $\mathcal{A}$ that contradicts [HMNY21]'s notion of security and construct a reduction $\mathcal{R}$ that contradicts our notion of security. $\mathcal{R}$ will run $\mathcal{A}$ on the challenge that it receives from its challenger, and forward the deletion certificate cert received from $\mathcal{A}$. $\mathcal{R}$ will then, in unbounded time, reverse sample a sk such that $(\mathsf{pk},\mathsf{sk})$ is identically distributed to the output of the honest Gen algorithm. Finally, $\mathcal{R}$ runs $\mathcal{A}$ on sk to obtain $\mathcal{A}$'s guess for b. We can show that the view of $\mathcal{A}$ produced by such an $\mathcal{R}$ matches its view in the [HMNY21] challenge, thus the advantage of $\mathcal{R}$ in contradicting our definition will match that of $\mathcal{A}$ in contradicting [HMNY21]'s definition.

Witness encryption for NP with certified deletion. Finally, we observe that if the underlying scheme PKE is a *witness encryption* scheme, then the scheme with certified deletion that results from the above compiler becomes a witness encryption scheme with certified deletion. That is, we compile any (post-quantum) witness encryption into a witness encryption scheme with certified deletion. Similar to the case of PKE and ABE, we can augment the syntax of any witness encryption scheme to include a deletion algorithm Del and a verification algorithm Ver. That is, the scheme consists of algorithms $(\mathsf{Enc},\mathsf{Dec})$ with syntax and properties identical to standard witness encryption schemes [GGSW13], except where the ciphertexts are potentially quantum, and where the encryption algorithm outputs a (potentially quantum) verification key vk along with a ciphertext. $\mathsf{Ver}(\mathsf{vk},\mathsf{cert})\to\{\top,\bot\}$ is a (potentially quantum) algorithm that takes as input a (potentially quantum) verification key vk and a (potentially quantum) deletion certificate cert and outputs either

$\top$ or $\bot$. $\mathsf{Del}(\mathsf{ct}) \to \mathsf{cert}$ is a quantum algorithm that on input a quantum ciphertext ct outputs a (potentially quantum) deletion certificate cert.

Correctness of decryption is the same as that for (regular) witness encryption. We define correctness of deletion, and certified deletion security for $\mathsf{CD\text{-}WE}$ below.

Definition 4.14 (Correctness of deletion)**.** $\mathsf{CD\text{-}WE} = (\mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Ver})$ *satisfies* correctness of deletion *if for every statement X and message m, it holds with $1-\mathrm{negl}(\lambda)$ probability over $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(X, m), \mathsf{cert} \leftarrow \mathsf{Del}(\mathsf{ct}), \mu \leftarrow \mathsf{Ver}(\mathsf{vk}, \mathsf{cert})$ that $\mu = \top$.*

Definition 4.15 (Certified deletion security)**.** $\mathsf{CD\text{-}WE} = (\mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Ver})$ *satisfies* certified deletion security *if for any non-uniform QPT adversary $\mathcal{A} = \{\mathcal{A}_\lambda, |\psi\rangle_\lambda\}_{\lambda\in\mathbb{N}}$, there is a negligible function $\mathrm{negl}(\cdot)$ for which it holds that*

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(0), \mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(1)\right) = \mathrm{negl}(\lambda),$$

and

$$\left|\Pr\left[\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(0) = 1\right] - \Pr\left[\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(1) = 1\right]\right| = \mathrm{negl}(\lambda),$$

where the experiments $\mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(b)$ and $\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(b)$ are defined as follows. Both experiments take an input b, and interact with $\mathcal{A}$ as follows.

- *Obtain statement X, language $\mathcal{L}$ and messages (m_0, m_1) from $\mathcal{A}_\lambda(|\psi\rangle_\lambda)$. If $X \in L$, abort, otherwise continue.*
- *Set $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(X, m_b)$.*
- *Run $\mathcal{A}_\lambda$ on input ct and parse $\mathcal{A}_\lambda$'s output as a deletion certificate cert, and a left-over quantum state ρ.*
- *If $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert}) = \top$ then output ρ, and otherwise output $\bot$.*

and the experiment $\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(b)$ is a strengthening of semantic security, defined as follows.

- *Obtain statement X, language $\mathcal{L}$ and messages (m_0, m_1) from $\mathcal{A}_\lambda(|\psi\rangle_\lambda)$. If $X \in L$, abort, otherwise continue.*
- *Set $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(X, m_b)$.*
- *Run $\mathcal{A}_\lambda$ on input ct and parse $\mathcal{A}_\lambda$'s output as a deletion certificate cert, and a left-over quantum state ρ.*
- *Output $\mathcal{A}_\lambda(\rho, \mathsf{Ver}(\mathsf{vk}, \mathsf{cert}))$.*

Corollary 4.16. *Given any post-quantum semantically-secure witness encryption scheme $\mathsf{WE} = (\mathsf{Enc}, \mathsf{Dec})$, the scheme $\mathsf{CD\text{-}WE} = (\mathsf{Enc}', \mathsf{Dec}', \mathsf{Del}, \mathsf{Ver})$ defined as follows is a witness encryption scheme with certified deletion.*

- *$\mathsf{Enc}'(X, m)$: sample $x, \theta \leftarrow \{0,1\}^\lambda$ and output*

$$\mathsf{ct} \coloneqq \left(|x\rangle_\theta, \mathsf{Enc}\left(X, \left(\theta, b \oplus \bigoplus_{i:\theta_i=0} x_i\right)\right)\right), \quad \mathsf{vk} \coloneqq (x, \theta).$$

- $\mathsf{Dec}'(W, \mathsf{ct})$: *parse* $\mathsf{ct} := (|x\rangle_\theta, \mathsf{ct}')$*, compute* $(\theta, b') \leftarrow \mathsf{Dec}(W, \mathsf{ct}')$*, measure* $|x\rangle_\theta$ *in the* θ*-basis to obtain* x*, and output* $b = b' \oplus \bigoplus_{i:\theta_i=0} x_i$*.*
- $\mathsf{Del}(\mathsf{ct})$: *parse* $\mathsf{ct} := (|x\rangle_\theta, \mathsf{ct}')$ *and measure* $|x\rangle_\theta$ *in the Hadamard basis to obtain a string* x'*, and output* $\mathsf{cert} := x'$*.*
- $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert})$: *parse* vk *as* (x, θ) *and* cert *as* x' *and output* $\top$ *if and only if* $x_i = x'_i$ *for all* i *such that* $\theta_i = 1$*.*

Proof. Correctness of deletion follows immediately from the description of the scheme. For certified deletion security, we consider the following:

- First, we observe that
$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(0), \mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(1)\right) = \mathrm{negl}(\lambda)$$
follows from Theorem 3.1 and the semantic security of WE by setting the distribution $\mathcal{Z}_\lambda(b)$ to sample $\mathsf{ct} \leftarrow \mathsf{Enc}(X, b)$ and output (X, ct), and setting the class of adversaries $\mathscr{A}$ to be all non-uniform families of QPT adversaries $\{\mathcal{A}_\lambda, |\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$.
- Next, we observe that
$$\left|\Pr\left[\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(0) = 1\right] - \Pr\left[\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(1) = 1\right]\right| = \mathrm{negl}(\lambda)$$
follows from the fact that the witness encryption scheme remains (computationally) semantically secure even when the adversary is given the verification key corresponding to the challenge ciphertext.

This completes our proof. □

4.3 Fully-homomorphic encryption

Next, we consider the syntax of a *fully-homomorphic encryption* scheme (for classical circuits) with certified deletion. Such a scheme consists of algorithms $\mathsf{CD\text{-}FHE} = (\mathsf{CD\text{-}FHE.Gen}, \mathsf{CD\text{-}FHE.Enc}, \mathsf{CD\text{-}FHE.Eval}, \mathsf{CD\text{-}FHE.Dec}, \mathsf{CD\text{-}FHE.Del}, \mathsf{CD\text{-}FHE.Ver})$ with the same syntax as $\mathsf{CD\text{-}PKE}$ (Section 4.2), but including the additional algorithm $\mathsf{CD\text{-}FHE.Eval}$.

- $\mathsf{CD\text{-}FHE.Eval}(\mathsf{pk}, C, \mathsf{ct}) \to \widetilde{\mathsf{ct}}$: On input the public key pk, a classical circuit C, and a quantum ciphertext ct, the evaluation algorithm returns a (potentially quantum) evaluated ciphertext $\widetilde{\mathsf{ct}}$.

We say that $\mathsf{CD\text{-}FHE}$ satisfies evaluation correctness if the following holds.

Definition 4.17 ($\mathsf{CD\text{-}FHE}$ evaluation correctness)**.** *A* $\mathsf{CD\text{-}FHE}$ *scheme satisfies evaluation correctness if for any message* x*, and all polynomial-size circuits* C*, it holds with* $1 - \mathrm{negl}(\lambda)$ *probability over* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{CD\text{-}FHE.Gen}(1^\lambda), \mathsf{ct} \leftarrow \mathsf{CD\text{-}FHE.Enc}(\mathsf{pk}, x)$*,* $\widetilde{\mathsf{ct}} \leftarrow \mathsf{CD\text{-}FHE.Eval}(\mathsf{pk}, C, \mathsf{ct}), y \leftarrow \mathsf{CD\text{-}FHE.Dec}(\mathsf{sk}, \widetilde{\mathsf{ct}})$ *that* $y = C(x)$*.*

Now we can formally define the notion of fully-homomorphic encryption with certified deletion.

Definition 4.18 (Fully-homomorphic encryption with certified deltion)**.** $\mathsf{CD\text{-}FHE} = (\mathsf{CD\text{-}FHE.Gen}, \mathsf{CD\text{-}FHE.Enc}, \mathsf{CD\text{-}FHE.Eval}, \mathsf{CD\text{-}FHE.Dec}, \mathsf{CD\text{-}FHE.Del}, \mathsf{CD\text{-}FHE.Ver})$ *is a secure* fully-homomorphic encryption scheme with certified deletion *if it satisfies (i) correctness of deletion (Definition 4.5), (ii) certified deletion security (Definition 4.6), and (iii) evaluation correctness (Definition 4.17).*

4.3.1 Blind delegation with certified deletion

So far, we have described a PKE scheme with certified deletion augmented with a procedure that allows for homomorphic evaluation over ciphertexts. A fascinating application for such a scheme, as discussed by [BI20, Por22], is the following. A computationally weak client wishes to use the resources of a powerful server to peform some intensive computation C on their input data x. However, they would like to keep x private from the server, and, moreover, they would like to be certain that their data is *deleted* by the server after the computation takes place. Here, by deleted, we mean that the original input x becomes *information-theoretically* hidden from the server after the computation has taken place.

While it is not necessarily clear from the syntax described so far that the server can both compute on *and later* delete the client's input data, we demonstrate, via an interaction pattern described by [Por22], a protocol that achieves this functionality. We refer to such a protocol as a "blind delegation with certified deletion" protocol, and describe it in Protocol 1.

Blind delegation with certified deletion

- Parties: client with input x, and server.
- Ingredients: a $\mathsf{CD\text{-}FHE}$ scheme.

Encryption phase

- The client samples $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{CD\text{-}FHE.Gen}(1^\lambda)$, $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{CD\text{-}FHE.Enc}(\mathsf{pk}, x)$ and sends $(\mathsf{pk}, \mathsf{ct})$ to the server.

Computation phase (this may be repeated arbitrarily many times)

- The client sends the description of a circuit C to the server.
- The server runs the algorithm $\mathsf{CD\text{-}FHE.Eval}(\mathsf{pk}, \mathsf{ct}, C)$ *coherently*. Let O be the (unmeasured) register that holds the output ciphertext $\widetilde{\mathsf{ct}}$. Send O to the client.
- The client runs $\mathsf{CD\text{-}FHE.Dec}(\mathsf{sk}, \cdot)$ *coherently* on register O, and then measures the output register of this computation in the standard basis to obtain the output y. Then, it reverses the computation of $\mathsf{CD\text{-}FHE.Dec}(\mathsf{sk}, \cdot)$ and sends the register O back to the server.
- The server reverses the computation of $\mathsf{CD\text{-}FHE.Eval}(\mathsf{pk}, \mathsf{ct}, C)$ to obtain the original input $(\mathsf{pk}, \mathsf{ct}, C)$.

Deletion phase

- The server runs $\mathsf{cert} \leftarrow \mathsf{CD\text{-}FHE.Del}(\mathsf{ct})$ and sends cert to the client.
- The client runs $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert})$ and outputs the result ($\top$ or $\bot$).

Figure 1: A generic construction of blind delegation with certified deletion, from any $\mathsf{CD\text{-}FHE}$ scheme.

A blind delegation with certified deletion protocol should satisfy the following notions of correctness and security. We present each definition for the case of a single circuit C queried by the client (one repetition of the computation phase), but they easily extend to considering multiple repetitions of the computation phase.

Definition 4.19 (Correctness for blind delegation with certified deletion)**.** *A blind delegation with certified deletion protocol is* correct *if the honest client and server algorithms satisfy the following properties. First, for any x, C, the client obtains $y = C(x)$ after the computation with probability $1 - \mathrm{negl}(\lambda)$. Second, for any x, C, the client outputs $\top$ after the deletion phase with probability $1 - \mathrm{negl}(\lambda)$.*

Definition 4.20 (Security for blind delegation with certified deletion)**.** *A blind delegation with certified deletion protocol is* secure *against a class of adversarial servers $\mathscr{S}$ if for any x_0, x_1, circuit C, and $\{\mathcal{S}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathscr{S}$, the following two properties hold.*

- ***Privacy:*** *For any QPT distinguisher $\{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\left| \Pr\left[\mathcal{D}_\lambda(\rho^{\mathsf{S}}_{x_0,C}) = 1\right] - \Pr\left[\mathcal{D}_\lambda(\rho^{\mathsf{S}}_{x_1,C}) = 1\right] \right| = \mathrm{negl}(\lambda),$$

  where $\rho^{\mathsf{S}}_{x,C} \leftarrow \langle \mathcal{C}(x, C), \mathcal{S}_\lambda\rangle$ is the output state of adversary $\mathcal{S}_\lambda$ after interacting (in the Encryption, Computation, and Delete phases) with an honest client $\mathcal{C}$ with input x and circuit C.

- ***Certified deletion:*** *It holds that*

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}^{\mathcal{S}}_\lambda(x_0, C), \mathsf{EV\text{-}EXP}^{\mathcal{S}}_\lambda(x_1, C)\right) = \mathrm{negl}(\lambda),$$

  where the experiment $\mathsf{EV\text{-}EXP}^{\mathcal{S}}_\lambda(x, C)$ is defined as follows.

  - *Run the Encryption, Computation, and Deletion phases between client $\mathcal{C}(x, C)$ and server $\mathcal{S}_\lambda$, obtaining the server's final state $\rho^{\mathsf{S}}_{x,C}$ and the client's decision $\top$ or $\bot$. If $\top$ output $\rho^{\mathsf{S}}_{x,C}$, and otherwise output $\bot$.*

Next, we define a class of adversaries $\mathscr{S}$ that we call *evaluation-honest*. The defining feature of an evaluation-honest adversary $\{\mathcal{S}_\lambda\}_{\lambda\in\mathbb{N}}$ is that, for any client input x and circuit C, the state on register O returned by $\mathcal{S}_\lambda$ during the Computation phase is within negligible trace distance of the state on register O returned by the *honest* server. Otherwise, $\mathcal{S}_\lambda$ may be arbitrarily malicious, including during the Deletion phase and after. Morally, evaluation-honest adversaries are those that are *specious* [DNS10] (which is a quantum analogue of semi-honest) during the Computation phase, and *malicious* afterwards.[17]

Then, we have the following theorem.

Theorem 4.21. *When instantiated with any* $\mathsf{CD\text{-}FHE}$ *scheme, Protocol 1 is a blind delegation with certified deletion scheme, secure against any* evaluation-honest *adversarial server $\{\mathcal{S}^*_\lambda\}_{\lambda\in\mathbb{N}}$.*

[17] Roughly, a specious adversary is one who may, at any step of the computation, apply an operation to their private state such that the joint state of the resulting system is negligibly close to the joint state of an honest interaction. Note that for any specious adversary, the registers they send to the honest party during any round must be negligibly close to the register sent by the honest party during this round, since after transmission, this register is no longer part of their private state. Thus, our definition of evaluation-honest adversaries includes all specious adversaries.

Proof. First, we argue that correctness (Definition 4.19) holds. The evaluation correctness of the underlying CD-FHE scheme (Definition 4.17) implies that the register measured by the client during the computation phase is within negligible trace distance of $|y\rangle$ for $y = C(x)$. This implies the first property of correctness for blind delegation with certified deletion, and it also implies that the state on register O returned to the server is negligibly close to the original state on register O. So, after reversing the computation, the server obtains a ct' that is negligibly close to the original ct received from the client. Thus, the second property of correctness for blind delegation with certified deletion follows from the correctness of deletion property of CD-FHE (Definition 4.5).

Next, we argue that security (Definition 4.20) holds against any evaluation-honest server $\{\mathcal{S}^*_\lambda\}_{\lambda\in\mathbb{N}}$. Consider a hybrid experiment in which there is no interaction between client and server during the Computation phase, that is, the register O is not touched by the client and is immediately return the the server. By the evaluation-honesty of the server, and the above arguments, it follows that the server's view of this hybrid experiment is negligibly close to its view of the real interaction. But now observe that this hybrid experiment is equivalent to the experiment described in Definition 4.9 for defining certified deletion security for standard public-key encryption for multi-bit messages. Thus, the Privacy and Certified Deletion properties of blind delegation follow directly from the certified deletion properties (Definition 4.9) of the underlying CD-FHE scheme. □

4.3.2 Construction of CD-FHE

Now, to obtain a blind delegation with certified deletion protocol, it suffices to construct a CD-FHE scheme. In this section, we show that such a scheme follows from a standard fully-homorphic encryption scheme, and our main theorem.

Corollary 4.22. *Given any classical fully-homomorphic encryption scheme* $\mathsf{FHE} = (\mathsf{FHE.Gen}, \mathsf{FHE.Enc}, \mathsf{FHE.Eval}, \mathsf{FHE.Dec})$*, the scheme defined below* $\mathsf{CD\text{-}FHE} = (\mathsf{FHE.Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Ver})$ *is an* FHE *scheme with certified deletion. We define encryption, decryption, deletion, and verification for one-bit plaintexts, and evaluation over ciphertexts encrypting n bits (which is simply a concatenation of n ciphertexts each encrypting one bit).*

- $\mathsf{Enc}(\mathsf{pk}, b)$: *Sample* $x, \theta \leftarrow \{0,1\}^\lambda$ *and output*

$$\mathsf{ct} := \left(|x\rangle_\theta, \mathsf{FHE.Enc}\left(\theta, b \oplus \bigoplus_{i:\theta_i=0} x_i\right)\right), \quad \mathsf{vk} := (x, \theta).$$

- $\mathsf{Eval}(\mathsf{pk}, C, \mathsf{ct})$: *Parse* $\mathsf{ct} := (|x_1\rangle_{\theta_1}, \mathsf{ct}'_1), \ldots, (|x_n\rangle_{\theta_n}, \mathsf{ct}'_n)$*. Consider the circuit* $\widetilde{C}$ *that takes* $(x'_1, \theta_1, b'_1), \ldots, (x'_n, \theta_n, b'_n)$ *as input, for each* $i \in [n]$ *computes* $b_i = b'_i \oplus \bigoplus_{j:\theta_{i,j}=0} x'_j$*, and then computes and outputs* $C(b_1, \ldots, b_n)$*. Then, apply* $\widetilde{C}$ *homomorphically in superposition to* ct *to obtain* $\widetilde{\mathsf{ct}}$*. Optionally, measure* $\widetilde{\mathsf{ct}}$ *in the standard basis to obtain a classical output ciphertext.*

- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: *parse* $\mathsf{ct} := (|x\rangle_\theta, \mathsf{ct}')$*, compute* $(\theta, b') \leftarrow \mathsf{FHE.Dec}(\mathsf{sk}, \mathsf{ct}')$*, measure* $|x\rangle_\theta$ *in the* θ*-basis to obtain* x*, and output* $b = b' \oplus \bigoplus_{i:\theta_i=0} x_i$*.*

- $\mathsf{Del}(\mathsf{ct})$: *parse* $\mathsf{ct} := (|x\rangle_\theta, \mathsf{ct}')$ *and measure* $|x\rangle_\theta$ *in the Hadamard basis to obtain a string* x'*, and output* $\mathsf{cert} := x'$*.*

- $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert})$: *parse* vk *as* (x, θ) *and* cert *as* x' *and output* $\top$ *if and only if* $x_i = x'_i$ *for all* i *such that* $\theta_i = 1$.

Proof. Semantic security follows immediately from the semantic security of FHE. Correctness of deletion follows immediately by definition the scheme. Certified deletion security follows from Theorem 3.1 and the semantic security of FHE by setting the distribution $\mathcal{Z}_\lambda(b)$ to sample $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{FHE.Gen}(1^\lambda)$, $\mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}, b)$ and output $(\mathsf{pk}, \mathsf{ct})$, and setting the class of adversaries $\mathscr{A}$ to be all non-uniform families of QPT adversaries $\{\mathcal{A}_\lambda, |\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$. □

4.4 Commitments and zero-knowledge

A bit commitment scheme is an interactive protocol between two (potentially quantum) interactive machines, a committer $\mathcal{C} = \{\mathcal{C}_{\mathsf{Com},\lambda}, \mathcal{C}_{\mathsf{Rev},\lambda}\}_{\lambda\in\mathbb{N}}$ and a receiver $\mathcal{R} = \{\mathcal{R}_{\mathsf{Com},\lambda}, \mathcal{R}_{\mathsf{Rev},\lambda}\}_{\lambda\in\mathbb{N}}$. It operates in two stages.

- In the Commit phase, the committer $\mathcal{C}_{\mathsf{Com},\lambda}(b)$ with input bit b interacts with the receiver $\mathcal{R}_{\mathsf{Com},\lambda}$. This interaction results in a joint state on a committer and receiver register, which we denote $(\mathsf{C}, \mathsf{R}) \leftarrow \mathsf{Com}\langle\mathcal{C}_{\mathsf{Com},\lambda}(b), \mathcal{R}_{\mathsf{Com},\lambda}\rangle$.
- In the Reveal phase, the parties continue to interact, and the receiver outputs a trit $\mu \in \{0, 1, \bot\}$, which we denote by $\mu \leftarrow \mathsf{Rev}\langle\mathcal{C}_{\mathsf{Rev},\lambda}(\mathsf{C}), \mathcal{R}_{\mathsf{Rev},\lambda}(\mathsf{R})\rangle$.

A commitment scheme that is *statistically binding and computationally hiding* is one that satisfies the following three properties.

Definition 4.23 (Correctness of decommitment)**.** *A commitment scheme satisfies* correctness of decommitment *if for any* $b \in \{0, 1\}$*, it holds with overwhelming probability over* $(\mathsf{C}, \mathsf{R}) \leftarrow \mathsf{Com}\langle\mathcal{C}_{\mathsf{Com},\lambda}(b), \mathcal{R}_{\mathsf{Com},\lambda}\rangle, \mu \leftarrow \mathsf{Rev}\langle\mathcal{C}_{\mathsf{Rev},\lambda}(\mathsf{C}), \mathcal{R}_{\mathsf{Rev},\lambda}(\mathsf{R})\rangle$ *that* $\mu = b$.

Definition 4.24 (Computational hiding)**.** *A commitment scheme satisfies* computational hiding *if for any non-uniform QPT adversary and distinguisher* $\mathcal{R}^* = \{\mathcal{R}^*_{\mathsf{Com},\lambda}, \mathcal{D}^*_\lambda, |\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$*, where* $|\psi_\lambda\rangle$ *is a state on two registers* $(\mathsf{R}^*, \mathsf{D}^*)$*, it holds that*

$$\Big|\Pr\left[\mathcal{D}^*_\lambda(\mathsf{R}^*, \mathsf{D}^*) = 1 : (\mathsf{C}, \mathsf{R}^*) \leftarrow \mathsf{Com}\langle\mathcal{C}_{\mathsf{Com},\lambda}(0), \mathcal{R}^*_{\mathsf{Com},\lambda}(\mathsf{R}^*)\rangle\right]$$
$$- \Pr\left[\mathcal{D}^*_\lambda(\mathsf{R}^*, \mathsf{D}^*) = 1 : (\mathsf{C}, \mathsf{R}^*) \leftarrow \mathsf{Com}\langle\mathcal{C}_{\mathsf{Com},\lambda}(1), \mathcal{R}^*_{\mathsf{Com},\lambda}(\mathsf{R}^*)\rangle\right]\Big| = \mathrm{negl}(\lambda).$$

We follow [AQY22]'s notion of statistical binding, which asks for an unbounded extractor that obtains the committer's bit during the Commit phase.

Definition 4.25 (Statistical binding)**.** *A commitment scheme satisfies* statistical binding *if for any unbounded adversary* $\{\mathcal{C}^*_{\mathsf{Com},\lambda}\}_{\lambda\in\mathbb{N}}$ *in the Commit phase, there exists an unbounded extractor* $\mathcal{E} = \{\mathcal{E}_\lambda\}_{\lambda\in\mathbb{N}}$ *such that for every unbounded adversary* $\{\mathcal{C}^*_{\mathsf{Rev},\lambda}\}_{\lambda\in\mathbb{N}}$ *in the Reveal phase,*

$$\mathsf{TD}\left(\mathsf{REAL}^{\mathcal{C}^*}_\lambda, \mathsf{IDEAL}^{\mathcal{C}^*,\mathcal{E}}_\lambda\right) = \mathrm{negl}(\lambda),$$

where $\mathsf{REAL}^{\mathcal{C}^*}_\lambda$ *and* $\mathsf{IDEAL}^{\mathcal{C}^*,\mathcal{E}}_\lambda$ *are defined as follows.*

- $\mathsf{REAL}_\lambda^{\mathcal{C}^*}$: *Execute the Commit phase* $(\mathsf{C}^*,\mathsf{R})\leftarrow\mathsf{Com}\langle\mathcal{C}^*_{\mathsf{Com},\lambda},\mathcal{R}_{\mathsf{Com},\lambda}\rangle$. *Execute the Reveal phase to obtain a trit* $\mu\leftarrow\mathsf{Rev}\langle\mathcal{C}^*_{\mathsf{Rev},\lambda}(\mathsf{C}^*),\mathcal{R}_{\mathsf{Rev},\lambda}(\mathsf{R})\rangle$ *along with the updated committer's state on register* C^*. *Output* (μ,C^*).

- $\mathsf{IDEAL}_\lambda^{\mathcal{C}^*,\mathcal{E}}$: *Run the extractor* $(\mathsf{C}^*,\mathsf{R},b^*)\leftarrow\mathcal{E}_\lambda$, *which outputs a joint state on registers* C^*,R *along with a bit* b^*. *Next, execute the Reveal phase to obtain a trit* $\mu\leftarrow\mathsf{Rev}\langle\mathcal{C}^*_{\mathsf{Rev},\lambda}(\mathsf{C}^*),\mathcal{R}_{\mathsf{Rev},\lambda}(\mathsf{R})\rangle$ *along with the updated committer's state on register* C^*. *If* $\mu\in\{\bot,b^*\}$ *output* (μ,C^*), *and otherwise output a special symbol* FAIL.

We will also consider commitment schemes with an additional (optional) *Delete* phase. That is, the committer and receiver will be written as three components: $\mathcal{C}_\lambda=\{\mathcal{C}_{\mathsf{Com},\lambda},\mathcal{C}_{\mathsf{Del},\lambda},\mathcal{C}_{\mathsf{Rev},\lambda}\}$, and $\mathcal{R}_\lambda=\{\mathcal{R}_{\mathsf{Com},\lambda},\mathcal{R}_{\mathsf{Del},\lambda},\mathcal{R}_{\mathsf{Rev},\lambda}\}$, and the protocol proceeds as follows.

- In the Commit phase, the committer $\mathcal{C}_{\mathsf{Com},\lambda}(b)$ with input bit b interacts with the receiver $\mathcal{R}_{\mathsf{Com},\lambda}$. This interaction results in a joint state on a committer and receiver register, which we denote $(\mathsf{C},\mathsf{R})\leftarrow\mathsf{Com}\langle\mathcal{C}_{\mathsf{Com},\lambda}(b),\mathcal{R}_{\mathsf{Com},\lambda}\rangle$.

- In the Delete phase, the parties continue to interact. The committer outputs a bit $d_{\mathcal{C}}\in\{\top,\bot\}$ indicating whether they accept or reject. We denote the resulting output and joint state of the committer and receiver by $(d_{\mathcal{C}},\mathsf{C},\mathsf{R})\leftarrow\mathsf{Del}\langle\mathcal{C}_{\mathsf{Del},\lambda}(\mathsf{C}),\mathcal{R}_{\mathsf{Del},\lambda}(\mathsf{R})\rangle$.

- The Reveal phase is only executed if the Delete phase has not been executed, and the receiver outputs a trit $\mu\in\{0,1,\bot\}$, which we denote by $\mu\leftarrow\mathsf{Rev}\langle\mathcal{C}_{\mathsf{Rev},\lambda}(\mathsf{C}),\mathcal{R}_{\mathsf{Rev},\lambda}(\mathsf{R})\rangle$.

For such commitments, we ask for an additional correctness property, and a stronger hiding property.

Definition 4.26 (Correctness of deletion, [HMNY22b])**.** *A bit commitment scheme satisfies* correctness of deletion *if for any* $b\in\{0,1\}$, *it holds with overwhelming probability over* $(\mathsf{C},\mathsf{R})\leftarrow\mathsf{Com}\langle\mathcal{C}_{\mathsf{Com},\lambda}(b),\mathcal{R}_{\mathsf{Com},\lambda}\rangle,(d_{\mathcal{C}},\mathsf{C},\mathsf{R})\leftarrow\mathsf{Del}\langle\mathcal{C}_{\mathsf{Del},\lambda}(\mathsf{C}),\mathcal{R}_{\mathsf{Del},\lambda}(\mathsf{R})\rangle$ *that* $d_{\mathcal{C}}=\top$.

Definition 4.27 (Certified everlasting hiding, [HMNY22b])**.** *A commitment scheme satisfies* certified everlasting hiding *if it satisfies the following two properties. First, for any non-uniform QPT adversary and distinguisher* $\mathcal{R}^*=\{\mathcal{R}^*_{\mathsf{Com},\lambda},\mathcal{R}^*_{\mathsf{Del},\lambda},\mathcal{D}^*_\lambda,|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$, *where* $|\psi_\lambda\rangle$ *is a state on two registers* $(\mathsf{R}^*,\mathsf{D}^*)$, *it holds that*

$$\left|\Pr\left[\mathcal{D}^*_\lambda(d_{\mathcal{C}},\mathsf{R}^*,\mathsf{D}^*)=1:\begin{array}{c}(\mathsf{C},\mathsf{R}^*)\leftarrow\mathsf{Com}\langle\mathcal{C}_{\mathsf{Com},\lambda}(0),\mathcal{R}^*_{\mathsf{Com},\lambda}(\mathsf{R}^*)\rangle\\(d_{\mathcal{C}},\mathsf{C},\mathsf{R}^*)\leftarrow\mathsf{Del}\langle\mathcal{C}_{\mathsf{Del},\lambda}(\mathsf{C}),\mathcal{R}^*_{\mathsf{Del},\lambda}(\mathsf{R}^*)\rangle\end{array}\right]\right.$$
$$\left.-\Pr\left[\mathcal{D}^*_\lambda(d_{\mathcal{C}},\mathsf{R}^*,\mathsf{D}^*)=1:\begin{array}{c}(\mathsf{C},\mathsf{R}^*)\leftarrow\mathsf{Com}\langle\mathcal{C}_{\mathsf{Com},\lambda}(1),\mathcal{R}^*_{\mathsf{Com},\lambda}(\mathsf{R}^*)\rangle\\(d_{\mathcal{C}},\mathsf{C},\mathsf{R}^*)\leftarrow\mathsf{Del}\langle\mathcal{C}_{\mathsf{Del},\lambda}(\mathsf{C}),\mathcal{R}^*_{\mathsf{Del},\lambda}(\mathsf{R}^*)\rangle\end{array}\right]\right|=\mathrm{negl}(\lambda).$$

Second, for any non-uniform QPT adversary $\mathcal{R}^*=\{\mathcal{R}^*_{\mathsf{Com},\lambda},\mathcal{R}^*_{\mathsf{Del},\lambda},|\psi\rangle\}_{\lambda\in\mathbb{N}}$, *where* $|\psi_\lambda\rangle$ *is a state on two registers* $(\mathsf{R}^*,\mathsf{D}^*)$, *it holds that*

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}^{\mathcal{R}^*}_\lambda(0),\mathsf{EV\text{-}EXP}^{\mathcal{R}^*}_\lambda(1)\right)=\mathrm{negl}(\lambda),$$

where the experiment $\mathsf{EV\text{-}EXP}^{\mathcal{R}^*}_\lambda(b)$ *is defined as follows.*

- *Execute the Commit phase* $(\mathsf{C},\mathsf{R}^*)\leftarrow\mathsf{Com}\langle\mathcal{C}_{\mathsf{Com},\lambda}(b),\mathcal{R}^*_{\mathsf{Com},\lambda}(\mathsf{R}^*)\rangle$.

- *Execute the Delete phase* $(d_{\mathcal{C}}, \mathsf{C}, \mathsf{R}^*) \leftarrow \mathsf{Del}\langle \mathcal{C}_{\mathsf{Del},\lambda}(\mathsf{C}), \mathcal{R}^*_{\mathsf{Del},\lambda}(\mathsf{R}^*))\rangle$.
- *If* $d_{\mathcal{C}} = \top$ *then output* $(\mathsf{R}^*, \mathsf{D}^*)$*, and otherwise output* $\bot$.

Then, we have the following corollary of Theorem 3.1.

Corollary 4.28. *Given any statistically binding computationally hiding commitment scheme* Com*, the commitment defined as follows is a statistically binding commitment scheme with certified everlasting hiding.*

- *The committer, on input* $b \in \{0,1\}$*, samples* $x, \theta \leftarrow \{0,1\}^\lambda$*. Then, the committer and receiver engage in the Commit phase of* Com*, where the committer has input* $(\theta, b \oplus \bigoplus_{i:\theta_i=0} x_i)$*. Finally, the committer sends* $|x\rangle_\lambda$ *to the receiver.*
- *For the Delete phase, the receiver measures the state* $|x\rangle_\theta$ *in the Hadamard basis to obtain a string* x'*, and sends* x' *to the committer. The committer outputs* $\top$ *if and only if* $x_i = x'_i$ *for all* i *such that* $\theta_i = 1$.
- *For the Reveal phase, the committer and receiver engage in the Reveal phase of* Com*, where the committer reveals the committed input* (θ, b')*. If this passes, the receiver measures* $|x\rangle_\theta$ *in the* θ *basis to obtain* x *and outputs* $b = b' \oplus \bigoplus_{i:\theta_i=0} x_i$.

Proof. First, we show that statistical binding is preserved. Given a malicious committer $\{\mathcal{C}^*_{\mathsf{Com},\lambda}\}_{\lambda\in\mathbb{N}}$, consider the experiment $\mathsf{IDEAL}^{\mathcal{C}^*,\mathcal{E}}_\lambda$ specified by an extractor $\mathcal{E}$ defined as follows.

- Invoke the extractor for the underlying commitment scheme on the first part of $\mathcal{C}^*_{\mathsf{Com},\lambda}$, which produces a joint state on $(\mathsf{C}^*, \mathsf{R})$ and extracted values (θ^*, b^*).
- Continue running $\mathcal{C}^*_{\mathsf{Com},\lambda}$ until it outputs a λ-qubit state on register X, which in the honest case will hold a state of the form $|x\rangle_\theta$.
- Measure the register X in the θ^* basis to produce x^*, and set the extracted bit $\widehat{b^*} := b^* \oplus \bigoplus_{i:\theta^*_i=0} x^*_i$.

Then, the Reveal phase of the underlying commitment scheme is run to produce a final committer's state on register C^* and receiver's output, which is either $\bot$ or some (θ', b'). Finally, the receiver either outputs $\bot$ or completes the Reveal phase by measuring register X in the θ basis to obtain x, and outputting $b' \oplus \bigoplus_{i:\theta_i=0} x_i$.

Note that, by the statistical binding property of the underlying commitment, the final state on C^* produced by $\mathcal{C}^*_{\mathsf{Rev},\lambda}$ in this experiment will be within negligible trace distance of the state on C^* output in the $\mathsf{REAL}^{\mathcal{C}^*}_\lambda$ experiment, and moreover the probability that $\mathcal{R}_{\mathsf{Rev},\lambda}$ accepts opened values (θ', b') that are not equal to the previously extracted values (θ^*, b^*) is negligible. Thus, conditioned on opening accepting, with all but negligible probability the extractor's and receiver's measurement of X will be identical. Thus, the extracted bit and receiver's output will be the same, and the outcome FAIL will only occur with negligible probability.

Next, we show certified everlasting hiding. The first property follows immediately from the hiding of the underlying commitment scheme, since there are no messages from $\mathcal{C}$ in the delete phase, and the bit $d_{\mathcal{C}}$ is computed independently of b. The second property follows from hiding of the underlying commitment scheme and Theorem 3.1 by setting $\mathcal{Z}_\lambda(\theta)$ and $\mathcal{A}_\lambda$ as follows.

- $\mathcal{Z}_\lambda(\theta)$ initializes registers $(\mathsf{R}^*, \mathsf{D}^*)$ with $|\psi_\lambda\rangle$, runs $(\mathsf{C}_\theta, \mathsf{R}^*) \leftarrow \mathsf{Com}\langle \mathcal{C}_\lambda(\theta), \mathcal{R}^*_{\mathsf{Com},\lambda}(\mathsf{R}^*)\rangle$ with the first part of $\mathcal{R}^*_{\mathsf{Com},\lambda}$ (where $\mathcal{C}_\lambda$ is the commit algorithm of the underlying commitment scheme Com), and outputs the resulting state on register R^*.
- $\mathcal{A}_\lambda$ receives $|x\rangle_\theta, b \oplus \bigoplus_{i:\theta_i=0} x_i$, and the state on register R^*. It first runs $(\mathsf{C}_b, \mathsf{R}^*) \leftarrow \mathsf{Com}\langle \mathcal{C}_\lambda(b \oplus \bigoplus_{i:\theta_i=0} x_i), \mathcal{R}^*_{\mathsf{Com},\lambda}(\mathsf{R}^*)\rangle$ with the remaining part of $\mathcal{R}^*_{\mathsf{Com},\lambda}$, and then runs $\mathcal{R}^*_{\mathsf{Del},\lambda}(\mathsf{R}^*, |x\rangle_\theta)$, which outputs a classical certificate and a left-over quantum state.

□

Remark 4.29. *We note that the above corollary explicitly considers underlying statistically binding commitment schemes that may include quantum communication, and thus one implication is that statistically binding commitments with certified everlasting hiding can be built just from the assumption that pseudo-random quantum states exist [MY22, AQY22].*

Remark 4.30. *Similarly to the setting of public-key encryption, single-bit certified everlasting hiding for statistically binding commitments implies multi-bit certified everlasting hiding.*

4.4.1 Certified everlasting zero-knowledge proofs for QMA

We begin by defining proofs for QMA with certified everlasting zero-knowledge, introduced in [HMNY22b]. Our definition is identical to theirs, except that we also guarantee computational zero-knowledge in the case that the verifier outputs invalid deletion certificates. In what follows, we will assume familiarity with the notion of a (statistically sound) proof for a QMA promise problem.

Definition 4.31. *A certified everlasting zero-knowledge proof for a QMA promise problem $A = (A_{\mathsf{yes}}, A_{\mathsf{no}})$ is a proof for A that additionally satisfies the following properties.*

- **(Perfect) Correctness of certified deletion.** *For every instance $x \in A_{\mathsf{yes}}$ and every state $|\psi\rangle \in R_A(x)$, the prover outputs $\top$ as its output in the interaction $\langle \mathcal{P}(x, |\psi\rangle^{\otimes k(|x|)}), \mathcal{V}(x)\rangle$.*
- **Certified everlasting zero-knowledge.** *Let $\mathsf{REAL}_\lambda\langle \mathcal{P}(x, |\psi\rangle^{\otimes k(|x|)}), \mathcal{V}^*(x)\rangle$ denote the joint distribution of the output of an honest prover and the state of an arbitrary QPT verifier $\mathcal{V}^*$ after they execute the proof on instance $x \in A_{\mathsf{yes}}$, where the prover has as quantum input a polynomial number $k(|x|)$ copies of a state $|\psi\rangle \in R_A(x)$. Then there exists a QPT algorithm Sim that on input any $x \in A_{\mathsf{yes}}$ and with oracle access to any non-uniform QPT $\mathcal{V}^* = \{\mathcal{V}^*_\lambda\}_{\lambda\in\mathbb{N}}$, outputs distribution $\mathsf{Sim}^{\mathcal{V}^*}_\lambda(x)$ such that:*
  - *First, we have everlasting zero-knowledge against adversaries that produce a valid deletion certificate, i.e.,*
  $$\mathsf{TD}\left(\mathsf{EV}\left(\mathsf{REAL}_\lambda\langle \mathcal{P}\left(x, |\psi\rangle^{\otimes k(|x|)}\right), \mathcal{V}^*(x)\rangle\right), \mathsf{EV}\left(\mathsf{Sim}^{\mathcal{V}^*}_\lambda(x)\right)\right) = \mathrm{negl}(\lambda),$$
  where $\mathsf{EV}(\cdot)$ is a quantum circuit that on input a classical string $o \in \{\top, \bot\}$ and a quantum state ρ outputs $(\top, \rho)$ when $o = \top$, and otherwise outputs $(\bot, \bot)$.
  - *Second, we have computational zero-knowledge against all adversaries, even when they do not necessarily output valid deletion certificates, i.e., for every QPT distinguisher $\mathcal{D}^* = \{\mathcal{D}^*_\lambda\}_{\lambda\in\mathbb{N}}$,*
  $$\left|\Pr\left[\mathcal{D}^*_\lambda\left(\mathsf{REAL}_\lambda\langle \mathcal{P}\left(x, |\psi\rangle^{\otimes k(|x|)}\right), \mathcal{V}^*(x)\rangle\right) = 1\right] - \Pr\left[\mathcal{D}^*_\lambda\left(\mathsf{Sim}^{\mathcal{V}^*}_\lambda(x)\right) = 1\right]\right| = \mathrm{negl}(\lambda)$$

Next, we define a notion of classical extractor-based binding for commitments. This definition was introduced in [HMNY22b], and while their definition requires perfect extraction, we observe that their theorem holds even if the underlying commitment satisfies only statistical extraction. As such we allow for $\mathrm{negl}(n)$ statistical error in our definition.

Definition 4.32 (Classical extractor-based binding [HMNY22b])**.** *A quantum commitment with classical non-interactive decommitment satisfies classical extractor-based binding if there exists an unbounded-time deterministic algorithm* Ext *that on input the classical transcript of a (possibly quantum) commitment* com*, outputs the only unique classical decommitment string* d *that will cause the verifier to accept the reveal phase, except with negligible probability.*

Finally, we will rely on the following theorem from [HMNY22b], which we describe below, paraphrased according to our definitions.

Theorem 4.33 ([HMNY22b])**.** *Assuming the existence of commitments satisfying statistical* classical extractor-based *binding and certified everlasting hiding (according to Definition 4.27), there exists a zero-knowledge proof for QMA satisfying certified everlasting zero-knowledge (according to Definition 4.31).*

We obtain the following corollary of Theorem 4.33 and our Corollary 4.28.

Corollary 4.34. *Assuming the existence of post-quantum one-way functions, there exists a zero-knowledge proof for QMA satisfying certified everlasting zero-knowledge (according to Definition 4.31).*

This corollary follows from the observation that our construction of commitments with certified everlasting hiding, when instantiated with any classical statistically binding commitment (and in particular Naor's commitment from one-way functions) satisfies classical extractor-based binding. The extractor simply outputs the decommitment of the classical part of our commitment. The resulting commitment with certified everlasting hiding and classical extractor-based binding can be plugged into Theorem 4.33 to obtain the corollary above.

4.5 Timed-release encryption

A timed-release encryption scheme [RSW96, Unr14] $\mathsf{TRE} = (\mathsf{TRE.Enc}, \mathsf{TRE.Dec})$ has the following syntax.

- $\mathsf{TRE.Enc}(1^\lambda, m) \to \mathsf{ct}$ is a polynomial-time algorithm that takes as input the security parameter 1^λ and a message m and outputs a ciphertext ct.
- $\mathsf{TRE.Dec}(\mathsf{ct}) \to m$ is a polynomial-time algorithm that takes as input a ciphertext ct and outputs a message m.

A timed-released encryption scheme is (post-quantum) $T(\lambda)$-hiding if the following holds.

Definition 4.35 (Hiding time-released encryption)**.** *A timed-released encryption scheme* $\mathsf{TRE} = (\mathsf{TRE.Enc}, \mathsf{TRE.Dec})$ *is $T(\lambda)$-hiding if for any non-uniform quantum polynomial-time*[18] *adversary $\mathcal{A} = \{\mathcal{A}_\lambda, |\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$ with at most $T(\lambda)$ parallel time,*

$$\left|\Pr\left[\mathcal{A}_\lambda(\mathsf{TRE.Enc}(1^\lambda, 0)) = 1\right] - \Pr\left[\mathcal{A}_\lambda(\mathsf{TRE.Enc}(1^\lambda, 1)) = 1\right]\right| = \mathrm{negl}(\lambda).$$

[18] As discussed in [Unr14], it is important to have a polynomial-time bound on the overall complexity of the adversary, in addition to the $T(\lambda)$ parallel time bound.

Now, we augment the syntax of a TRE scheme with algorithms $\mathsf{RTRE.Del}, \mathsf{RTRE.Ver}$ to arrive at the notion of a *revocable* timed-release encryption scheme RTRE.

- $\mathsf{RTRE.Enc}(1^\lambda, m) \to (\mathsf{ct}, \mathsf{vk})$ is a polynomial-time algorithm that takes as input the security parameter 1^λ and a message m and outputs a quantum ciphertext ct and a (potentially quantum) verification key vk.
- $\mathsf{RTRE.Dec}(\mathsf{ct}) \to m$ is a polynomial-time algorithm that takes as input a quantum ciphertext ct and outputs a message m.
- $\mathsf{RTRE.Del}(\mathsf{ct}) \to \mathsf{cert}$ is a quantum algorithm that takes as input a quantum ciphertext ct and outputs a (potentially quantum) deletion certificate cert.
- $\mathsf{RTRE.Ver}(\mathsf{vk}, \mathsf{cert}) \to \{\top, \bot\}$ is a (potentially quantum) algorithm that takes as input a (potentially quantum) verification key vk and a (potentially quantum) deletion certificate cert and outputs either $\top$ or $\bot$.

We say that RTRE satisfies revocable hiding if the following holds.

Definition 4.36 (Revocably hiding time-released encryption)**.** *A timed-released encryption scheme* $\mathsf{RTRE} = (\mathsf{RTRE.Enc}, \mathsf{RTRE.Dec}, \mathsf{RTRE.Del}, \mathsf{RTRE.Ver})$ *is $T(\lambda)$-revocably hiding if for any non-uniform quantum polynomial-time adversary $\mathcal{A} = \{\mathcal{A}_\lambda, |\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$ with at most $T(\lambda)$ parallel time, it holds that*

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}_\lambda^{\mathcal{A}}(0), \mathsf{EV\text{-}EXP}\lambda^{\mathcal{A}}(1)\right) = \mathrm{negl}(\lambda),$$

and

$$\left|\Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{A}}(0) = 1\right] - \Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{A}}(1) = 1\right]\right| = \mathrm{negl}(\lambda),$$

where the experiment $\mathsf{EV\text{-}EXP}\lambda^{\mathcal{A}}(b)$ is defined as follows.

- *Sample* $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{RTRE.Enc}(1^\lambda, b)$.
- *Initialize* $\mathcal{A}_\lambda(|\psi_\lambda\rangle)$ *with* ct.
- *Parse* $\mathcal{A}_\lambda$*'s output as a deletion certificate* cert *and a left-over quantum state* ρ.
- *If* $\mathsf{RTRE.Ver}(\mathsf{vk}, \mathsf{cert}) = \top$ *then output* ρ*, and otherwise output* $\bot$.

and the experiment $\mathsf{C\text{-}EXP}\lambda^{\mathcal{A}}(b)$ is defined as follows.

- *Sample* $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{RTRE.Enc}(1^\lambda, b)$.
- *Initialize* $\mathcal{A}_\lambda(|\psi_\lambda\rangle)$ *with* ct.
- *Parse* $\mathcal{A}_\lambda$*'s output as a deletion certificate* cert *and a left-over quantum state* ρ.
- *Output* $\mathcal{A}_\lambda(\rho, \mathsf{Ver}(\mathsf{vk}, \mathsf{cert}))$.

We say that RTRE is a revocable time-released encryption scheme against $T(\lambda)$-parallel time adversaries if it satisfies (i) hiding (Definition 4.35) against $T(\lambda)$-parallel time adversaries, (ii) correctness of deletion (Definition 4.5), and (iii) recovable hiding (Definition 4.36) against $T(\lambda)$-parallel time adversaries.

Then, we have the following corollary of Theorem 3.1.

Corollary 4.37. *Given any post-quantum secure time-released encryption* $\mathsf{TRE} = (\mathsf{TRE.Enc}, \mathsf{TRE.Dec})$ *against* $T(\lambda)$*-parallel time adversaries, the scheme* $\mathsf{RTRE} = (\mathsf{Enc}', \mathsf{Dec}', \mathsf{Del}, \mathsf{Ver})$ *defined as follows is a secure* revocable *time-released encryption scheme against* $T(\lambda)$*-parallel time adversaries.*

- $\mathsf{Enc}'(\mathsf{pk}, m)$: *sample* $x, \theta \leftarrow \{0,1\}^\lambda$ *and output*

$$\mathsf{ct} := \left(|x\rangle_\theta, \mathsf{TRE.Enc}\left(\theta, b \oplus \bigoplus_{i:\theta_i=0} x_i\right)\right), \quad \mathsf{vk} := (x, \theta).$$

- $\mathsf{Dec}'(\mathsf{sk}, \mathsf{ct})$: *parse* $\mathsf{ct} := (|x\rangle_\theta, \mathsf{ct}')$*, compute* $(\theta, b') \leftarrow \mathsf{TRE.Dec}(\mathsf{sk}, \mathsf{ct}')$*, measure* $|x\rangle_\theta$ *in the* θ*-basis to obtain* x*, and output* $b = b' \oplus \bigoplus_{i:\theta_i=0} x_i$*.*

- $\mathsf{Del}(\mathsf{ct})$: *parse* $\mathsf{ct} := (|x\rangle_\theta, \mathsf{ct}')$ *and measure* $|x\rangle_\theta$ *in the Hadamard basis to obtain a string* x'*, and output* $\mathsf{cert} := x'$*.*

- $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert})$: *parse* vk *as* (x, θ) *and* cert *as* x' *and output* $\top$ *if and only if* $x_i = x'_i$ *for all* i *such that* $\theta_i = 1$*.*

Proof. Hiding follows immediately from the hiding of TRE. Correctness of deletion follows immediately from the description of the scheme. Revocable hiding follows because

- First,
$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}^{\mathcal{A}}_\lambda(0), \mathsf{EV\text{-}EXP}\lambda^{\mathcal{A}}(1)\right) = \mathrm{negl}(\lambda),$$
follows from Theorem 3.1 and the hiding of TRE, and by setting the distribution $\mathcal{Z}(b)$ to sample $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{TRE.Enc}(1^\lambda, b)$ and output ct, and setting the class of adversaries $\mathscr{A}$ to be all non-uniform QPT adversaries $\{\mathcal{A}_\lambda, |\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$ with at most $T(\lambda)$ parallel time.

- Second,
$$\left|\Pr\left[\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(0) = 1\right] - \Pr\left[\mathsf{C\text{-}EXP}^{\mathcal{A}}_\lambda(1) = 1\right]\right| = \mathrm{negl}(\lambda),$$
follows from the fact that the timed-release encryption remains (computationally) semantically secure even when the adversary is given the verification key corresponding to the challenge ciphertext.

This completes our proof. □

Remark 4.38. *Similarly to the setting of public-key encryption, single-bit certified everlasting hiding for timed-release encryption implies certified everlasting hiding for multi-bit messages.*

5 Cryptography with Everlasting Security Transfer

In this section, we construct bit commitment and secure computation schemes that satisfy our notion of Everlasting Security Transfer (EST). In Section 5.1, we formalize the notion of deletion and EST in the context of simulation security. We also derive a quantum sequential composition theorem for *reactive functionalities*, extending the framework of [HSS11]. Extending to reactive functionalities is crucial for us, since the bit commitment functionality we compose has multiple

phases. Finally, we formalize composition of protocols with EST, defining the notion of a "deletion-composable" protocol. Next, we show how to construct ideal commitments with EST in Section 5.2 and Section 5.3, via a two-step process outlined in Section 1.2. Finally, in Section 5.4, we make use of our ideal commitment with EST, our composition theorems, and known compilers, to obtain the notions of two-party and multi-party computation with EST.

5.1 Definitions

Ideal functionalities. An ideal functionality $\mathcal{F}$ is a classical interactive machine specifying some (potentially reactive) distributed classical computation. Reactive means that the distributed computation is broken into multiple "phases" with distinct inputs and outputs, and the outputs of previous phases may be used as inputs in later phases. For now, we will specifically consider *two-party* functionalities. Each invocation of an ideal functionality is associated with some session id sid. We will be interested in designing protocols that *securely realize* ideal functionalities (defined later), but first we specify the main ideal functionality that we consider in this work: bit commitment $\mathcal{F}_{\mathsf{com}}$.

Ideal functionality $\mathcal{F}_{\mathsf{com}}$

Parties: committer C and receiver R

- Commit phase: $\mathcal{F}_{\mathsf{com}}$ receives a query $(\mathsf{Commit}, \mathsf{sid}, b)$ from C, records this query, and sends $(\mathsf{Commit}, \mathsf{sid})$ to R.
- Reveal phase: $\mathcal{F}_{\mathsf{com}}$ receives a query $(\mathsf{Reveal}, \mathsf{sid})$ from C, and if a message $(\mathsf{Commit}, \mathsf{sid}, b)$ has been recorded, sends $(\mathsf{Reveal}, \mathsf{sid}, b)$ to R.

Figure 2: Specification of the bit commitment ideal functionality.

In this work, we will consider augmenting ideal functionalities with a "deletion phase", which can be used by parties to transfer everlasting security. When parties are labeled A and B, we maintain the precedent that deletion is from B to A, that is, A can request that B deletes A's information.

Deletion phase

Parties: A and B

- Receive a query $(\mathsf{DelRequest}, \mathsf{sid})$ from A, and send $(\mathsf{DelRequest}, \mathsf{sid})$ to B.
- Receive a query $(\mathsf{DelResponse}, \mathsf{sid})$ from B. If a message $(\mathsf{DelRequest}, \mathsf{sid})$ has been recorded, send $(\mathsf{DelResponse}, \mathsf{sid})$ to A, and otherwise ignore the message.

Figure 3: Specification of a generic deletion phase that can be added to any ideal functionality $\mathcal{F}$.

Importantly, if a deletion phase is added to a *reactive* functionality $\mathcal{F}$, we allow party A to

request the Deletion phase (send $\mathsf{DelRequest}$) *between any two phases of* $\mathcal{F}$, or *at the end of* $\mathcal{F}$. But, once the Deletion phase has been executed, this marks the end of the reactive functionality, so no other phases will be executed.

Security with abort. In what follows, we will by default consider the notion of *security with abort*, where the ideal functionality $\mathcal{F}$ is always modified to (1) know the identities of corrupted parties and (2) be slightly reactive: after all parties have provided input, the functionality computes outputs and sends these outputs to the corrupt parties only. Then the functionality awaits either a "deliver" or "abort" command from the corrupted parties. Upon receiving "deliver", the functionality delivers any honest party outputs. Upon receiving "abort", the functionality instead delivers abort to all the honest parties.

The real-ideal paradigm. A two-party protocol $\Pi_{\mathcal{F}}$ for computing the (potentially reactive) functionality $\mathcal{F}$ consists of two families of quantum interactive machines A and B. An adversary intending to attack the protocol by corrupting a party $M \in \{A, B\}$ can be described by a family of sequences of quantum interactive machines $\{\mathcal{A}_\lambda := (\mathcal{A}_{\lambda,1}, \ldots, \mathcal{A}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$, where ℓ is the number of phases of $\mathcal{F}$. This adversarial interaction happens in the presence of an *environment*, which is a family of sequences of quantum operations $\{\mathcal{Z}_\lambda := (\mathcal{Z}_{\lambda,1}, \ldots, \mathcal{Z}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$, and a family of initial advice states $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$. It proceeds as follows.

- $\mathcal{Z}_{\lambda,1}$ receives as input $|\psi_\lambda\rangle$. It outputs what (if any) inputs the honest party $H \in \{A, B\}$ is initialized with for the first phase of $\Pi_{\mathcal{F}}$. It also outputs a quantum state on registers (A, Z), where A holds the state of the adversary and Z holds the state of the environment,
- $\mathcal{A}_{\lambda,1}$ receives as input a state on register A, and interacts with the honest party in the first phase of $\Pi_{\mathcal{F}}$. It outputs a state on register A.
- $\mathcal{Z}_{\lambda,2}$ receives as input registers (A, Z) along with the honest party outputs from the first phase. It computes honest party inputs for the second phase, and updates registers (A, Z).
- $\mathcal{A}_{\lambda,2}, \mathcal{Z}_{\lambda,3}, \ldots, \mathcal{A}_{\lambda,\ell}$ are defined analogously.

Given an adversary, environment, and advice, we define the random variable $\Pi_{\mathcal{F}}[\mathcal{A}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$ as the output of the above procedure, which includes registers (A, Z) and the final honest party outputs.

An *ideal-world* protocol $\widetilde{\Pi}_{\mathcal{F}}$ for functionality $\mathcal{F}$ consists of "dummy" parties $\widetilde{A}$ and $\widetilde{B}$ that have access to an additional "trusted" party that implements $\mathcal{F}$. That is, $\widetilde{A}$ and $\widetilde{B}$ only interact directly with $\mathcal{F}$, providing inputs and receiving outputs, and do not interact with each other. We consider the execution of ideal-world protocols in the presence of a simulator, described by a family of sequences of quantum interactive machines $\{\mathcal{S}_\lambda := (\mathcal{S}_{\lambda,1}, \ldots, \mathcal{S}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$, analogous to the definition of an adversary above. This interaction also happens in the presence of an environment $\{\mathcal{Z}_\lambda := (\mathcal{Z}_{\lambda,1}, \ldots, \mathcal{Z}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$, and a family of initial advice states $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$, as described above, and we define the analogous random variable $\widetilde{\Pi}_{\mathcal{F}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$.

Secure realization and composition. Now, we formally define what it means for a protocol $\Pi_{\mathcal{F}}$ to securely realize a (potentially reactive) functionality $\mathcal{F}$. We give definitions for both computational and statistical security.

Definition 5.1 (Computational secure realization)**.** *A protocol $\Pi_{\mathcal{F}}$* computationally securely realizes *the ℓ-phase functionality $\mathcal{F}$ if for every QPT adversary $\{\mathcal{A}_\lambda := (\mathcal{A}_{\lambda,1}, \ldots, \mathcal{A}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$ corrupting either party A or B, there exists a QPT simulator $\{\mathcal{S}_\lambda := (\mathcal{S}_{\lambda,1}, \ldots, \mathcal{S}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$ such that for any QPT environment $\{\mathcal{Z}_\lambda := (\mathcal{Z}_{\lambda,1}, \ldots, \mathcal{Z}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$, polynomial-size family of advice $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$, and QPT distinguisher $\{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\left|\Pr\left[\mathcal{D}_\lambda\left(\Pi_{\mathcal{F}}[\mathcal{A}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]\right) = 1\right] - \Pr\left[\mathcal{D}_\lambda\left(\widetilde{\Pi}_{\mathcal{F}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]\right) = 1\right]\right| = \mathrm{negl}(\lambda).$$

For the notion of statistical secure realization, we allow the adversary and environment to be unbounded, but we require that the simulator is at most polynomially larger than the adversary.

Definition 5.2 (Statistical secure realization)**.** *A protocol $\Pi_{\mathcal{F}}$* statistically securely realizes *the ℓ-phase functionality $\mathcal{F}$ if there exists a polynomial $p(\cdot)$ such that for every (potentially unbounded) adversary $\{\mathcal{A}_\lambda := (\mathcal{A}_{\lambda,1}, \ldots, \mathcal{A}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$ corrupting either party A or B, there exists a simulator $\{\mathcal{S}_\lambda := (\mathcal{S}_{\lambda,1}, \ldots, \mathcal{S}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$ with size at most $p(\lambda)$ times the size of $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, such that for any (potentially unbounded) environment $\{\mathcal{Z}_\lambda := (\mathcal{Z}_{\lambda,1}, \ldots, \mathcal{Z}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$ and polynomial-size family of advice $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\mathsf{TD}\left(\Pi_{\mathcal{F}}[\mathcal{A}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle], \widetilde{\Pi}_{\mathcal{F}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]\right) = \mathrm{negl}(\lambda).$$

Remark 5.3. *Recall that trace distance between two distributions is an upper bound on the advantage that any unbounded machine has in distinguishing the distributions, so the above definition is equivalent to saying that no unbounded distinguisher has better than negligible advantage in distinguishing the real and ideal world outputs.*

Next, we consider the *hybrid* model, where parties can make calls to an ideal-world protocol implementing some ideal functionality $\mathcal{G}$. We call such a protocol a $\mathcal{G}$-hybrid protocol, and denote it $\Pi^{\mathcal{G}}$. Supposing that we also have a real-world protocol Γ implementing $\mathcal{G}$, we can consider the *composed* protocol $\Pi^{\mathcal{G}/\Gamma}$, where each invocation of $\mathcal{G}$ is replaced with an invocation of the protocol Γ for computing $\mathcal{G}$. In this work, while we allow Π to utilize many invocations of Γ, **we require that each phase of each invocation of Γ is *atomic*, meaning that no other protocol messages are interleaved during each phase of Γ.** That is, if $\mathcal{G}$ is a reactive functionality, we allow different phases of different invocations to be interleaved, but we require that at any point in time, only a single phase is being executed, and no other protocol messages are interleaved during the computation of this phase. In this case, we can show the following sequential composition theorem, which is a straightforward extension of the composition theorem given in [HSS11] to handle reactive functionalities.

Theorem 5.4 (Extension of [HSS11])**.** *Let $\mathcal{F}$ and $\mathcal{G}$ be (potentially reactive) functionalities, let $\Pi^{\mathcal{G}}$ be a $\mathcal{G}$-hybrid protocol that computationally (resp. statistically) securely realizes $\mathcal{F}$, and let Γ be a protocol that computationally (resp. statistically) securely realizes $\mathcal{G}$. Then, $\Pi^{\mathcal{G}/\Gamma}$ computationally (resp. statistically) securely realizes $\mathcal{F}$.*

Proof. Let $\mathcal{G}$ be a reactive ℓ-phase functionality. Throughout this proof, we drop the dependence on λ for convenience. Let $(\mathcal{A}, \mathcal{Z})$ be any adversary and environment attacking the protocol $\Pi^{\mathcal{G}/\Gamma}$. Consider the first time in $\Pi^{\mathcal{G}/\Gamma}$ that Γ is invoked, which means the first time that the first phase of some subroutine Γ is invoked (we note that other Γ subroutines could occur between the phases of this first invocation). Write $(\mathcal{Z}_1, \mathcal{A}_1, \ldots, \mathcal{Z}_\ell, \mathcal{A}_\ell)$ as an adversary and environment attacking the protocol Γ, according to the following.

- $\mathcal{Z}_1$ runs $\mathcal{Z}$ and then runs the interaction between $\mathcal{A}$ and the honest party until right before the first time Γ is invoked in $\Pi^{\mathcal{G}/\Gamma}$. It outputs the adversary's state on register A, the honest party's input to Γ, and any other state kept by $\mathcal{Z}$ along with the honest party's state on register Z.
- $\mathcal{A}_1$ consists of the part of $\mathcal{A}$ that interacts in the first phase of Γ. It takes as input a state on A and outputs a state on A.
- $\mathcal{Z}_2$ takes as input registers (A, Z) and the honest party's output from Γ. It runs the interaction between $\mathcal{A}$ and the honest party in $\Pi^{\mathcal{G}/\Gamma}$ until right before the second phase of Γ is invoked.
- $\mathcal{A}_2, \mathcal{Z}_3, \ldots, \mathcal{A}_\ell$ are defined analogously.

Now, since Γ computationally (resp. statistically) securely realizes $\mathcal{G}$, there exists a simulator $(\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ defined based on $(\mathcal{A}_1, \ldots, \mathcal{A}_\ell)$ such that, if we replace each $\mathcal{A}_i$ interacting with the honest party with $\mathcal{S}_i$ interacting with the ideal functionality $\mathcal{G}$, then the output remains computationally (resp. statistically) indistinguishable. Note that the resulting interaction, defined by $(\mathcal{Z}_1, \mathcal{S}_1, \ldots, \mathcal{Z}_\ell, \mathcal{S}_\ell)$, can be described by an adversary and environment $(\mathcal{A}', \mathcal{Z})$ attacking the protocol $\Pi^{\mathcal{G}/\Gamma}$ where the first invocation of Γ is replaced with the parties querying the ideal functionality $\mathcal{G}$. This follows because only the parts of $\mathcal{A}$ that interacted in the first invocation of Γ were changed, since each phase of Γ was atomic. Now, continuing this argument for each invocation of Γ, we eventually arrive at an adversary and environment $(\mathcal{A}'', \mathcal{Z})$ attacking the $\mathcal{G}$-hybrid protocol $\Pi^{\mathcal{G}}$. Note that $\mathcal{A}''$ was defined based on $\mathcal{A}$, and $\mathcal{Z}$ remained unchanged. Thus, the fact that $\Pi^{\mathcal{G}}$ computationally (resp. statistically) securely realizes $\mathcal{F}$ completes the proof of the theorem, since we can define a simulator $\mathcal{S}''$ based on $\mathcal{A}''$, where indistinguishability will hold for any environment $\mathcal{Z}$. □

Secure realization with everlasting security transfer. Next, we define the notion of secure realization with everlasting security transfer (EST). Here, parties are interested in securely computing an ideal functionality $\mathcal{F}$ with a *deletion phase* added to the end, which we denote by $\mathcal{F}^{\mathsf{Del}}$.

The deletion phase adds one bits to each honest party output, which we denote by DelReq (which is party B's output, and is set to 1 if party A initiates the Delete phase by issuing a request, and 0 otherwise) and DelRes (which is party A's output, and is set to 1 if party B sends a Delete response and 0 otherwise). Then, we have the following definition.

Definition 5.5 (Secure realization with Everlasting Security Transfer)**.** *A protocol $\Pi_{\mathcal{F}}$ securely realizes the ℓ-phase functionality $\mathcal{F}$ between parties A and B with EST if $\Pi_{\mathcal{F}}$ computationally securely realizes $\mathcal{F}^{\mathsf{Del}}$ (Definition 5.1) and the following additional properties hold.*

- **Statistical security against A when no security transfer occurs.** *There exists a polynomial $p(\cdot)$ such that for every (potentially unbounded) adversary $\{\mathcal{A}_\lambda := (\mathcal{A}_{\lambda,1}, \ldots, \mathcal{A}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$ corrupting party A, there exists a simulator $\{\mathcal{S}_\lambda := (\mathcal{S}_{\lambda,1}, \ldots, \mathcal{S}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$ with size at most $p(\lambda)$ times the of size of $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, such that for any (potentially unbounded) environment $\{\mathcal{Z}_\lambda := (\mathcal{Z}_{\lambda,1}, \ldots, \mathcal{Z}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$ and polynomial-size family of advice $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$,*

$$\mathsf{TD}\left(\Pi^{\mathsf{DelReq}=0}_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{A}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle], \widetilde{\Pi}^{\mathsf{DelReq}=0}_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]\right) = \mathrm{negl}(\lambda),$$

where $\Pi^{\mathsf{DelReq}=0}_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{A}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$ is defined to be equal to $\Pi_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{A}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$ if party B's output DelReq is set to 0, and defined to be $\perp$ otherwise, and likewise for $\widetilde{\Pi}^{\mathsf{DelReq}=0}_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$.

- **Certified everlasting security against B.** *For every QPT adversary $\{\mathcal{A}_\lambda := (\mathcal{A}_{\lambda,1},\ldots,\mathcal{A}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$ corrupting party B, there exists a QPT simulator $\{\mathcal{S}_\lambda := (\mathcal{S}_{\lambda,1},\ldots,\mathcal{S}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$ such that for any QPT environment $\{\mathcal{Z}_\lambda := (\mathcal{Z}_{\lambda,1},\ldots,\mathcal{Z}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$, and polynomial-size family of advice $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$,*

$$\mathsf{TD}\left(\Pi^{\mathsf{DelRes}=1}_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{A}_\lambda,\mathcal{Z}_\lambda,|\psi_\lambda\rangle],\widetilde{\Pi}^{\mathsf{DelRes}=1}_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{S}_\lambda,\mathcal{Z}_\lambda,|\psi_\lambda\rangle]\right)=\mathrm{negl}(\lambda),$$

where $\Pi^{\mathsf{DelRes}=1}_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{A}_\lambda,\mathcal{Z}_\lambda,|\psi_\lambda\rangle]$ is defined to be equal to $\Pi_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{A}_\lambda,\mathcal{Z}_\lambda,|\psi_\lambda\rangle]$ if party A's output DelRes is set to 1, and defined to be $\bot$ otherwise, and likewise for $\widetilde{\Pi}^{\mathsf{DelRes}=1}_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{S}_\lambda,\mathcal{Z}_\lambda,|\psi_\lambda\rangle]$.

Deletion-composable protocols. Finally, we consider the composition of protocols that securely realize functionalities with EST. Suppose we have a $\mathcal{G}^{\mathsf{Del}}$-hybrid protocol $\Pi^{\mathcal{G}^{\mathsf{Del}}}$ for implementing a functionality $\mathcal{F}^{\mathsf{Del}}$. We say that $\Pi^{\mathcal{G}^{\mathsf{Del}}}$ is *deletion-composable* if the following two properties hold.

1. If the deletion phase of $\mathcal{F}^{\mathsf{Del}}$ is never requested, then none of the deletion phases of $\mathcal{G}^{\mathsf{Del}}$ are requested.
2. If the deletion phase of $\mathcal{F}^{\mathsf{Del}}$ is accepted by party A, meaning that $\mathsf{DelRes}=1$, then it must be the case that the deletion phases of all the $\mathcal{G}^{\mathsf{Del}}$ sub-routines are requested and accepted by A.

Then, we can show the following composition theorem, which essentially follows from Theorem 5.4.

Theorem 5.6. *Let $\Pi^{\mathcal{G}^{\mathsf{Del}}}$ be a deletion-composable protocol that statistically securely realizes[19] a functionality $\mathcal{F}^{\mathsf{Del}}$, and let Γ be a protocol that securely implements $\mathcal{G}^{\mathsf{Del}}$ with EST. Then $\Pi^{\mathcal{G}^{\mathsf{Del}}/\Gamma}$ securely implements $\mathcal{F}^{\mathsf{Del}}$ with EST.*

Proof. First, the fact that $\Pi^{\mathcal{G}^{\mathsf{Del}}/\Gamma}$ computationally securely realizes $\mathcal{F}^{\mathsf{Del}}$ follows from directly from Theorem 5.4 and the fact that both statistical secure realization and secure realization with EST imply computational secure realization.

Next, statistical security against A in $\Pi^{\mathcal{G}^{\mathsf{Del}}/\Gamma}$ when $\mathsf{DelReq}=0$ also follows directly from Theorem 5.4 (applied to statistical secure realization), since by the first property of deletion-composability, all of the underlying Γ protocols are statistically secure against A.

Finally, we argue certified everlasting security against B in $\Pi^{\mathcal{G}^{\mathsf{Del}}/\Gamma}$ when $\mathsf{DelRes}=1$. This does not follow generically from the statement of Theorem 5.4. However, it can be shown via essentially the same proof as the proof of Theorem 5.4. Starting with $\Pi^{\mathcal{G}^{\mathsf{Del}}/\Gamma}$, we replace each invocation of Γ with an invocation of $\mathcal{G}^{\mathsf{Del}}$ one by one. Conditioned on the deletion phase of $\mathcal{G}^{\mathsf{Del}}$ passing, we know that this switch is statistically indistinguishable by the environment, due to the fact that Γ securely implements $\mathcal{G}^{\mathsf{Del}}$ with EST. Thus, conditioned on the deletion phase of each $\mathcal{G}^{\mathsf{Del}}$ being accepted, we know that protocols $\Pi^{\mathcal{G}^{\mathsf{Del}}/\Gamma}$ and $\Pi^{\mathcal{G}^{\mathsf{Del}}}$ are statistically indistinguishable by the environment. We also know that $\Pi^{\mathcal{G}^{\mathsf{Del}}}$ and $\mathcal{F}^{\mathsf{Del}}$ are statistically indistinguishable by the environment, by assumption. Thus, by the second property of deletion composability, it follows that $\Pi^{\mathcal{G}^{\mathsf{Del}}/\Gamma}$ and $\mathcal{F}^{\mathsf{Del}}$ are statistically indistinguishable by the environment conditioned on $\mathsf{DelRes}=1$, completing the proof. □

[19] One could strengthen this theorem to only requiring that $\Pi^{\mathcal{G}^{\mathsf{Del}}}$ securely realizes $\mathcal{F}^{\mathsf{Del}}$ with EST, but we state the theorem with statistical security for simplicity.

5.2 One-sided ideal commitments

In this section, we construct what we call a *one-sided ideal commitment with EST*. In the following subsection, we define this primitive as well as some underlying building blocks.

5.2.1 Definitions and building blocks

A one-sided ideal commitment with EST satisfies full-fledged security with EST against a malicious committer, but not against a malicious receiver. This commitment satisfies the weaker property of certified everlasting hiding against a malicious receiver. These properties are formalized below, where we denote by $\mathcal{F}_{\mathsf{Com}}^{\mathsf{Del}}$ the commitment ideal functionality from Protocol 2 augmented with the delete phase from Protocol 3.

Definition 5.7 (One-sided ideal commitment with EST)**.** *A three-phase (Commit, Reveal, Delete) commitment scheme is a* one-sided ideal commitment with EST *if*

1. *It computationally securely realizes $\mathcal{F}_{\mathsf{Com}}^{\mathsf{Del}}$ (Definition 5.1) against a corrupt committer C.*
2. *It satisfies statistical security against a corrupt committer C that does not initiate deletion (first part of Definition 5.5).*
3. *It satisfies correctness of deletion (Definition 4.26) and it satisfies certified everlasting hiding (Definition 4.27) against adversaries that corrupt the receiver R.*

To construct this object, our building block will be a *computationally-hiding statistically-efficiently-extractable* (CHSEE) commitment, which is a two-phase (Commit, Reveal) commitment that satisfies correctness (Definition 4.23), standard computational hiding (Definition 4.24), and the following notion of binding. Note that this is similar to Definition 4.25, except that the extractor must be *efficient*.

Definition 5.8 (Statistical efficient extractability)**.** *A commitment scheme satisfies* statistical efficient extractability *if for any QPT adversary $\{\mathcal{C}^*_{\mathsf{Com},\lambda}\}_{\lambda\in\mathbb{N}}$ in the Commit phase, there exists a QPT extractor $\mathcal{E} = \{\mathcal{E}_\lambda\}_{\lambda\in\mathbb{N}}$, such that for any initial advice $\{|\psi_\lambda\rangle^{\mathsf{Aux},\mathsf{C}^*}\}_{\lambda\in\mathbb{N}}$ and any QPT adversary $\{\mathcal{C}^*_{\mathsf{Rev},\lambda}\}_{\lambda\in\mathbb{N}}$ in the Reveal phase,*

$$\mathsf{TD}\left(\mathsf{REAL}_\lambda^{\mathcal{C}^*}, \mathsf{IDEAL}_\lambda^{\mathcal{C}^*,\mathcal{E}}\right) = \mathrm{negl}(\lambda),$$

where $\mathsf{REAL}_\lambda^{\mathcal{C}^*}$ and $\mathsf{IDEAL}_\lambda^{\mathcal{C}^*,\mathcal{E}}$ are defined as follows.

- $\mathsf{REAL}_\lambda^{\mathcal{C}^*}$*: Execute the Commit phase $(\mathsf{C}^*,\mathsf{R}) \leftarrow \mathsf{Com}\langle\mathcal{C}^*_{\mathsf{Com},\lambda}(\mathsf{C}^*), \mathcal{R}_{\mathsf{Com},\lambda}\rangle$, where $\mathcal{C}^*_{\mathsf{Com},\lambda}$ has as input the C^* register of $|\psi_\lambda\rangle^{\mathsf{Aux},\mathsf{C}^*}$. Execute the Reveal phase to obtain a trit $\mu \leftarrow \mathsf{Rev}\langle\mathcal{C}^*_{\mathsf{Rev},\lambda}(\mathsf{C}^*), \mathcal{R}_{\mathsf{Rev},\lambda}(\mathsf{R})\rangle$ along with the committer's final state on register C^*. Output $(\mu,\mathsf{C}^*,\mathsf{Aux})$, which includes the Aux register of the original advice state.*
- $\mathsf{IDEAL}_\lambda^{\mathcal{C}^*,\mathcal{E}}$*: Run the extractor $(b^*,\mathsf{C}^*,\mathsf{R}) \leftarrow \mathcal{E}_\lambda(\mathsf{C}^*)$, where the extractor takes as input the C^* register of $|\psi_\lambda\rangle^{\mathsf{Aux},\mathsf{C}^*}$, and outputs a bit b^* and a state on registers C^*,R. Next, execute the Reveal phase to obtain a trit $\mu \leftarrow \mathsf{Rev}\langle\mathcal{C}^*_{\mathsf{Rev},\lambda}(\mathsf{C}^*), \mathcal{R}_{\mathsf{Rev},\lambda}(\mathsf{R})\rangle$ along with the committer's final state on register C^*. If $\mu \in \{\bot, b^*\}$ output $(\mu,\mathsf{C}^*,\mathsf{Aux})$, and otherwise output a special symbol FAIL.*

Imported Theorem 5.9 ([BCKM21]). *There exists a construction of CHSEE commitments that makes black-box use of any computationally-hiding statistically-binding commitment (Definition 4.24 and Definition 4.25).*[20]

These are implied by OT with statistical security against one party, which was constructed in [BCKM21], based on the black-box use of computationally-hiding statistically-binding commitments. Alternatively, CHSEE commitments can be obtained more directly by plugging in the statistically-equivocal computationally-extractable commitments from [BCKM21] into the extractability compiler [BCKM21, Section 5]. Furthermore, this implies that CHSEE commitments can be based on the black-box use of one-way functions [BCKM21] or pseudo-random quantum states [MY22, AQY22].

5.2.2 Construction

We construct one-sided ideal commitments with EST from CHSEE commitments in Protocol 4. We note that constructing one-sided ideal commitments with EST does not just follow immediately from applying our certified deletion compiler, as in our construction of commitments with certified everlasting hiding from statistically-binding commitments in Section 4.4. The reason is that we need indistinguishability between the real and ideal worlds to hold against a malicious committer *even if the delete phase is run.* To satisfy this property, we actually use additional invocations of the CHSEE commitment going in the "opposite" direction during the Delete phase of Protocol 4.

Next, we prove the following theorem.

Theorem 5.10. *Protocol 4 is a one-sided ideal commitment with EST (according to Definition 5.7).*

The theorem follows by combining Lemmas 5.11, 5.12, 5.13 and 5.14 proved below.

5.2.3 Security against a corrupt committer

Lemma 5.11. *Protocol 4 computationally securely realizes $\mathcal{F}_{\mathsf{Com}}^{\mathsf{Del}}$ (Definition 5.1) against a corrupt committer C. That is, for every QPT committer $\{\mathcal{C}^*_\lambda := (\mathcal{C}^*_{\lambda,\mathsf{Com}}, \mathcal{C}^*_{\lambda,\mathsf{Rev}}, \mathcal{C}^*_{\lambda,\mathsf{Del}})\}_{\lambda\in\mathbb{N}}$, there exists a QPT simulator $\{\mathcal{S}_\lambda := (\mathcal{S}_{\lambda,\mathsf{Com}}, \mathcal{S}_{\lambda,\mathsf{Rev}}, \mathcal{S}_{\lambda,\mathsf{Del}})\}_{\lambda\in\mathbb{N}}$, such that for any QPT environment $\{\mathcal{Z}_\lambda := (\mathcal{Z}_{\lambda,1}, \mathcal{Z}_{\lambda,2}, \mathcal{Z}_{\lambda,3})\}_{\lambda\in\mathbb{N}}$, polynomial-size family of advice $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$, and QPT distinguisher $\{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\left|\Pr\left[\mathcal{D}_\lambda\left(\Pi_{\mathcal{F}_{\mathsf{Com}}^{\mathsf{Del}}}[\mathcal{C}^*_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]\right) = 1\right] - \Pr\left[\mathcal{D}_\lambda\left(\widetilde{\Pi}_{\mathcal{F}_{\mathsf{Com}}^{\mathsf{Del}}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]\right) = 1\right]\right| = \mathrm{negl}(\lambda).$$

Lemma 5.12. *Protocol 4 satisfies statistical security against a corrupt committer C that does not initiate deletion. That is, there exists a polynomial $p(\cdot)$ such that for every (potentially unbounded) committer $\{\mathcal{C}^*_\lambda := (\mathcal{C}^*_{\mathsf{Com},\lambda}, \mathcal{C}^*_{\mathsf{Rev},\lambda}, \mathcal{C}^*_{\mathsf{Del},\lambda})\}_{\lambda\in\mathbb{N}}$, there exists a simulator $\{\mathcal{S}_\lambda := (\mathcal{S}_{\lambda,\mathsf{Com}}, \mathcal{S}_{\lambda,\mathsf{Rev}}, \mathcal{S}_{\lambda,\mathsf{Del}})\}_{\lambda\in\mathbb{N}}$ with size at most $p(\lambda)$ times the of size of $\mathcal{C}^*$, such that for any (potentially unbounded) environment $\{\mathcal{Z}_\lambda := (\mathcal{Z}_{\lambda,1}, \mathcal{Z}_{\lambda,2}, \mathcal{Z}_{\lambda,3})\}_{\lambda\in\mathbb{N}}$ and polynomial-size family of advice $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$,*

$$\mathsf{TD}\left(\Pi^{\mathsf{DelReq}=0}_{\mathcal{F}_{\mathsf{Com}}^{\mathsf{Del}}}[\mathcal{C}^*_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle], \widetilde{\Pi}^{\mathsf{DelReq}=0}_{\mathcal{F}_{\mathsf{Com}}^{\mathsf{Del}}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]\right) = \mathrm{negl}(\lambda),$$

[20] In [BCKM21], a different notion of statistical binding for the underlying commitment was used, but it was noted by [AQY22] that the extractor-based definition of statistical binding suffices.

Protocol 4: One-sided ideal commitment with EST

Ingredients: a CHSEE commitment $(\mathsf{Com}, \mathsf{Rev})$
Parties: committer C with input $b \in \{0,1\}$ and receiver R.

Commit phase

- C samples $x, \theta \leftarrow \{0,1\}^\lambda$.
- C and R execute $\mathsf{C}_\theta, \mathsf{R}_\theta \leftarrow \mathsf{Com}\langle C(\theta), R\rangle$.
- C and $\mathcal{R}$ execute $\mathsf{C}_b, \mathsf{R}_b \leftarrow \mathsf{Com}\langle C(b \oplus \bigoplus_{i:\theta_i=0} x_i), R\rangle$.
- C sends $|x\rangle_\theta$ to R on register X.

Reveal phase

- C and R execute $\theta \leftarrow \mathsf{Rev}\langle C(\mathsf{C}_\theta), R(\mathsf{R}_\theta)\rangle$.
- C and R execute $b' \leftarrow \mathsf{Rev}\langle C(\mathsf{C}_b), R(\mathsf{R}_b)\rangle$.
- R measures the qubits i of register X such that $\theta_i = 0$ to obtain x_i, and then outputs $b' \oplus \bigoplus_{i:\theta_i=0} x_i$.

Delete phase[a]

- R measures all qubits of register X in the Hadamard basis to obtain a string $x' \in \{0,1\}^\lambda$.
- R and C execute λ Commit phases of Com, with R as the committer, committing bit-by-bit to x': $\mathsf{R}_{x',i}, \mathsf{C}_{x',i} \leftarrow \mathsf{Com}\langle R(x'_i), C\rangle$.
- C and R execute $\theta \leftarrow \mathsf{Rev}\langle C(\mathsf{C}_\theta), R(\mathsf{R}_\theta)\rangle$.
- R and C execute the Reveal phase of Com for each i such that $\theta_i = 1$: $x'_i \leftarrow \mathsf{Rev}\langle R(\mathsf{R}_{x',i}), C(\mathsf{C}_{x',i})\rangle$.
- C accepts (outputs 1) if $x'_i = x_i$ for all i such that $\theta_i = 1$.

[a] Note that both the Reveal phase and the Delete phase require C and R to run $\mathsf{Rev}\langle C(\mathsf{C}_\theta), R(\mathsf{R}_\theta)\rangle$. So if the Reveal phase has already been run, we can instruct R to abort if a deletion is requested, since we don't require any correctness of deletion or everlasting security after Reveal.

Figure 4: Construction of one-sided ideal commitment with EST, from a CHSEE commitment.

where $\Pi^{\mathsf{DelReq}=0}_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{C}^*_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$ *is defined to equal* $\Pi_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{C}^*_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$ *if the receiver's output* DelReq *is set to 0, and defined to be* $\perp$ *otherwise, and likewise for* $\widetilde{\Pi}^{\mathsf{DelReq}=0}_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$.

Proof. (of Lemmas 5.11 and 5.12) We define a simulator $\mathcal{S} = (\mathcal{S}_{\mathsf{Com}}, \mathcal{S}_{\mathsf{Rev}}, \mathcal{S}_{\mathsf{Del}})$ based on any adversary $\mathcal{C}^* = (\mathcal{C}^*_{\mathsf{Com}}, \mathcal{C}^*_{\mathsf{Rev}}, \mathcal{C}^*_{\mathsf{Del}})$ that will suffice to prove both lemmas. We have dropped the

dependence on λ for notational convenience.

1. **Commit Phase.** $\mathcal{S}_{\mathsf{Com}}$ does the following.
   - Run the CHSEE extractor $(\theta^*, \mathsf{C}^*, \mathsf{R}_\theta) \leftarrow \mathcal{E}_\lambda[\mathcal{C}^*_{\mathsf{Com},\theta}](\mathsf{C}^*)$, where the extractor is defined based on the part of $\mathcal{C}^*_{\mathsf{Com}}$ that interacts in the commitment to θ. It takes as input $\mathcal{C}^*_{\mathsf{Com}}$'s private state register C^*, and outputs a sequence of committed bits θ^* and a state on $\mathsf{C}^*, \mathsf{R}_\theta$.
   - Next, run the CHSEE extractor $(d^*, \mathsf{C}^*, \mathsf{R}_b) \leftarrow \mathcal{E}_\lambda[\mathcal{C}^*_{\mathsf{Com},b}](\mathsf{C}^*)$, where the extractor is defined based on the part of $\mathcal{C}^*_{\mathsf{Com}}$ that interacts in the commitment to $b \oplus \bigoplus_{i:\theta_i=0} x_i$. It takes as input $\mathcal{C}^*$'s private state register C^*, and outputs a committed bit d^* and a state on registers $\mathsf{C}^*, \mathsf{R}_b$.
   - Keep running $\mathcal{C}^*_{\mathsf{Com}}$ until it outputs a state on register X.
   - Measure the qubits i of register X such that $\theta^*_i = 0$ to obtain x^*_i, and then send $(\mathsf{Commit}, \mathsf{sid}, b^*)$ where $b^* = d^* \oplus \bigoplus_{i:\theta^*_i=0} x^*_i$ to the ideal functionality.
2. **Reveal Phase.** $\mathcal{S}_{\mathsf{Rev}}$ does the following.
   - Execute the Reveal phase of CHSEE to obtain $\theta' \leftarrow \mathsf{Rev}\langle\mathcal{C}^*_{\mathsf{Rev},\theta}(\mathsf{C}^*), R(\mathsf{R}_\theta)\rangle$(and update the register C^*).
   - Execute the Reveal phase of CHSEE to obtain $d' \leftarrow \mathsf{Rev}\langle\mathcal{C}^*_{\mathsf{Rev},b}(\mathsf{C}^*), R(\mathsf{R}_b)\rangle$ (and update the register C^*).
   - If $\theta^* = \theta'$ and $d^* = d'$ then send $(\mathsf{Reveal}, \mathsf{sid})$ to the ideal functionality.
3. **Delete Phase.** If $\mathcal{C}^*_{\mathsf{Del}}$ initializes the Delete phase, $\mathcal{S}_{\mathsf{Del}}$ sends $(\mathsf{DelRequest}, \mathsf{sid})$ to the ideal functionality, and upon obtaining $(\mathsf{DelResponse}, \mathsf{sid})$, it does the following.
   - For every i such that $\theta^*_i = 1$, measure the i^{th} qubit of register X in the Hadamard basis to obtain x'_i. For all i such that $\theta^*_i \in \{0, \perp\}$, set $x'_i = 0$.
   - Execute λ commit phases of Com, with the simulator as the committer, committing bit-by-bit to x'.
   - Execute the Reveal phase of $\mathcal{C}^*$'s commitments to θ to obtain $\theta' \leftarrow \mathsf{Rev}\langle\mathcal{C}^*_{\mathsf{Del},\theta}(\mathsf{C}^*), R(\mathsf{R}_\theta)\rangle$ (and update the register C^*), where $\mathcal{C}^*_{\mathsf{Del},\theta}$ is the part of $\mathcal{C}^*_{\mathsf{Del}}$ that interacts in the reveal phase of the commitment to θ.
   - If $\theta' \neq \theta^*$, abort. Otherwise, execute with $\mathcal{C}^*$ the Reveal phase of commitments to x' restricted to indices $i \in [\lambda]$ such that $\theta'_i = 1$.

It is straightforward to see that the simulator runs in quantum polynomial time as long as $\mathcal{C}^*$ runs in quantum polynomial time.

Statistical indistinguishability between the real and ideal distributions at the end of the Commit Phase or the Reveal Phase follows directly from Definition 5.8, thereby proving Lemma 5.12.

Furthermore, in the Delete phase, simulator and receiver strategies are identical on indices where $\theta^*_i = 1$. The only difference between these strategies is that the simulator commits to 0 when $\theta^*_i = 0$ whereas the receiver commits to outcomes of measurements of the i^{th} qubit of register X in the Hadamard basis. Now, the Reveal phase for these commitments are only run when $\theta' = \theta^*$, and only restricted to indices $i \in [\lambda]$ such that $\theta'_i = 1$. Thus, computational indistinguishability during the Delete phase follows from a reduction to the computational hiding of Com. This proves Lemma 5.11. □

5.2.4 Security against a corrupt receiver

Lemma 5.13. *Protocol 4 satisfies correctness of deletion (Definition 4.26).*

Proof. This follows immediately from the description of the scheme. □

Lemma 5.14. *Protocol 4 satisfies certified everlasting hiding (Definition 4.27) against adversaries that corrupt the receiver R.*

Proof. The first property of certified everlasting hiding follows immediately from the computational hiding of CHSEE and the fact that the delete phase is completely independent of the committed bit b.

The second property follows from the computational hiding of CHSEE and Theorem 3.1 by setting $\mathcal{Z}_\lambda(\theta)$ and $\mathcal{A}_\lambda$ as follows, based on any non-uniform corrupt receiver $\mathcal{R}^* = \{\mathcal{R}^*_{\lambda,\mathsf{Com}}, \mathcal{R}^*_{\lambda,\mathsf{Del}}, |\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$.

- $\mathcal{Z}_\lambda(\theta)$ initializes registers $(\mathsf{R}^*, \mathsf{D}^*)$ with $|\psi_\lambda\rangle$, runs $(\mathsf{C}_\theta, \mathsf{R}^*) \leftarrow \mathsf{Com}\langle C(\theta), \mathcal{R}^*_{\lambda,\mathsf{Com}}(\mathsf{R}^*)\rangle$ with the first part of $\mathcal{R}^*_{\mathsf{Com},\lambda}$, and outputs the resulting state on register R^*.
- $\mathcal{A}_\lambda$ receives $|x\rangle_\theta$, $b\oplus\bigoplus_{i:\theta_i=0} x_i$, and the state on register R^*. It first runs $(\mathsf{C}_b, \mathsf{R}^*) \leftarrow \mathsf{Com}\langle C(b\oplus \bigoplus_{i:\theta_i=0} x_i), \mathcal{R}^*_{\lambda,\mathsf{Com}}(\mathsf{R}^*)\rangle$ with the remaining part of $\mathcal{R}^*_{\lambda,\mathsf{Com}}$. Then, it sends $|x\rangle_\theta$ to $\mathcal{R}^*_{\lambda,\mathsf{Com}}$. Next, it runs $\mathcal{R}^*_{\lambda,\mathsf{Del}}$ until the beginning of the part where $\mathcal{R}^*_{\lambda,\mathsf{Del}}$ is supposed to commit to x'. At this point, it runs the extractor $\mathcal{E}_\lambda(\mathsf{R}^*)$ for Com, which outputs a certificate x' and a left-over quantum state on register R^*.

Note that the delete phase succeeds in the experiment $\mathsf{EV\text{-}EXP}^{\mathcal{R}^*}_\lambda(b)$ iff for every i where $\theta_i = 1$, $x'_i = x_i$, where the x'_i are opened by $\mathcal{R}^*_{\lambda,\mathsf{Del}}$. Also, by statistical efficient extractability of Com, the $\{x'_i\}_{i:\theta_i=1}$ output by $\mathcal{E}_\lambda$ are equal to the $\{x'_i\}_{i:\theta_i=1}$ opened by $\mathcal{R}^*_{\lambda,\mathsf{Del}}$, except with negligible probability. Thus, Theorem 3.1 implies that

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}^{\mathcal{R}^*}_\lambda(0), \mathsf{EV\text{-}EXP}^{\mathcal{R}^*}_\lambda(1)\right) = \mathrm{negl}(\lambda).$$

This completes the proof of the lemma. □

5.3 Ideal commitments

In this section, we show how to generically upgrade a one-sided ideal commitment with EST to a full-fledged ideal commitment with EST. Our construction, which is given in Protocol 5, is essentially the "equivocality compiler" from [BCKM21] with an added Delete phase.

Theorem 5.15. *Protocol 5 securely realizes the commitment ideal functionality with EST (according to Definition 5.5).*

The theorem follows by combining Lemmas 5.16, 5.17, 5.19 and 5.20 proved below.

5.3.1 Security against a corrupt committer

Lemma 5.16. *Protocol 5 computationally securely realizes $\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}$ (Definition 5.1) against a corrupt committer C.*

Protocol 5: Ideal commitment with EST

Ingredients: a one-sided ideal commitment with EST $(\mathsf{Com}, \mathsf{Rev}, \mathsf{Del})$.
Parties: committer C with input $b \in \{0,1\}$ and receiver R.

Commit phase

1. C samples uniformly random bits $a_{i,j}$ for $i \in [\lambda]$ and $j \in \{0,1\}$.
2. For every $i \in [\lambda]$, C and R sequentially perform the following steps.
   (a) C and R execute four Commit phases sequentially, namely:
   - $\mathsf{C}_{i,0,0}, \mathsf{R}_{i,0,0} \leftarrow \mathsf{Com}\langle C(a_{i,0}), R\rangle$,
   - $\mathsf{C}_{i,0,1}, \mathsf{R}_{i,0,1} \leftarrow \mathsf{Com}\langle C(a_{i,0}), R\rangle$,
   - $\mathsf{C}_{i,1,0}, \mathsf{R}_{i,1,0} \leftarrow \mathsf{Com}\langle C(a_{i,1}), R\rangle$,
   - $\mathsf{C}_{i,1,1}, \mathsf{R}_{i,1,1} \leftarrow \mathsf{Com}\langle C(a_{i,1}), R\rangle$.

   (b) R sends a choice bit $c_i \leftarrow \{0,1\}$.
   (c) C and R execute two Reveal phases, obtaining the opened bits:
   - $u \leftarrow \mathsf{Rev}\langle C(\mathsf{C}_{i,c_i,0}), R(\mathsf{R}_{i,c_i,0})\rangle$,
   - $v \leftarrow \mathsf{Rev}\langle C(\mathsf{C}_{i,c_i,1}), R(\mathsf{R}_{i,c_i,1})\rangle$.

   If $u \neq v$, R aborts. Otherwise, C and R continue.
3. For $i \in [\lambda]$, C sets $b_i = b \oplus a_{i,1-c_i}$ and sends $\{b_i\}_{i\in[\lambda]}$ to R.

Reveal phase

1. C sends b to R. In addition,
   (a) For $i \in [\lambda]$, C picks $\alpha_i \leftarrow \{0,1\}$ and sends it to $\mathcal{R}$.
   (b) C and R execute $a'_i \leftarrow \mathsf{Rev}\langle C(\mathsf{C}_{i,1-c_i,\alpha_i}), R(\mathsf{R}_{i,1-c_i,\alpha_i})\rangle$.
2. R accepts and outputs b if for every $i \in [\lambda]$, $a'_i = b \oplus b_i$.

Delete phase

1. For every $i \in [\lambda]$, C and R sequentially perform the following steps.
   (a) If Reveal was performed, execute $D_i \leftarrow \mathsf{Del}\langle C(\mathsf{C}_{i,1-c_i,1-\alpha_i}, R(\mathsf{R}_{i,1-c_i,1-\alpha_i})\rangle$.
   (b) Otherwise, set $D_i = D_{i,0} \wedge D_{i,1}$ where $D_{i,0} \leftarrow \mathsf{Del}\langle C(\mathsf{C}_{i,1-c_i,0}, R(\mathsf{R}_{i,1-c_i,0})\rangle$ and $D_{i,1} \leftarrow \mathsf{Del}\langle C(\mathsf{C}_{i,1-c_i,1}, R(\mathsf{R}_{i,1-c_i,1})\rangle$.
2. If $D_i = 1$ for all $i \in [\lambda]$, then C outputs 1.

Figure 5: Ideal commitment with EST, from a one-sided ideal commitment with EST.

Lemma 5.17. *Protocol 5 satisfies statistical security against a corrupt committer C that does not initiate deletion. That is, there exists a polynomial $p(\cdot)$ such that for every (potentially unbounded) adversary $\{\mathcal{C}^*_\lambda := (\mathcal{C}^*_{\lambda,\mathsf{Com}}, \mathcal{C}^*_{\lambda,\mathsf{Rev}}, \mathcal{C}^*_{\lambda,\mathsf{Del}})\}_{\lambda\in\mathbb{N}}$ corrupting C, there exists a simulator $\{\mathcal{S}_\lambda := (\mathcal{S}_{\lambda,\mathsf{Com}}, \mathcal{S}_{\lambda,\mathsf{Rev}}, \mathcal{S}_{\lambda,\mathsf{Del}})\}_{\lambda\in\mathbb{N}}$ with size at most $p(\lambda)$ times the of size of $\{\mathcal{C}^*_\lambda\}_{\lambda\in\mathbb{N}}$, such that for any*

(potentially unbounded) environment $\{\mathcal{Z}_\lambda := (\mathcal{Z}_{\lambda,1}, \mathcal{Z}_{\lambda,2}, \mathcal{Z}_{\lambda,3})\}_{\lambda\in\mathbb{N}}$ *and polynomial-size family of advice* $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$,

$$\mathsf{TD}\left(\Pi^{\mathsf{DelReq}=0}_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{C}^*_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle], \widetilde{\Pi}^{\mathsf{DelReq}=0}_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]\right) = \mathrm{negl}(\lambda),$$

where $\Pi^{\mathsf{DelReq}=0}_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{C}^*_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$ *is defined to equal* $\Pi_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{C}^*_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$ *if the receiver's output* DelReq *is set to 0, and defined to be* $\perp$ *otherwise, and likewise for* $\widetilde{\Pi}^{\mathsf{DelReq}=0}_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$.

Proof. (of Lemmas 5.16 and 5.17)

The Simulator. The simulator $(\mathcal{S}_{\mathsf{Com}}, \mathcal{S}_{\mathsf{Rev}}, \mathcal{S}_{\mathsf{Del}})$ is defined as follows.

1. **Commit Phase.** $\mathcal{S}_{\mathsf{Com}}$ does the following.
   - For all $i \in [\lambda]$,
     - Execute four sequential simulated Commit phases where the simulator for the commitment Com is run on the part of the committer $\mathcal{C}^*_{\mathsf{Com}}$ participating in each of the four sequential sessions. Denote the bit output by the simulator in each session by $(d_{i,0,0}, d_{i,0,1}, d_{i,1,0}, d_{i,1,1})$.
     - Sample and send choice bit $c_i \leftarrow \{0,1\}$ to $\mathcal{C}^*_{\mathsf{Com}}$.
     - Execute two simulated Reveal phases where the simulator is run on the part of the committer $\mathcal{C}^*_{\mathsf{Com}}$ corresponding to sessions $(i, c_i, 0)$ and $(i, c_i, 1)$. If the simulator outputs $(\mathsf{Reveal}, \mathsf{sid})$ for both sessions and $d_{i,c_i,0} = d_{i,c_i,1}$, continue, and otherwise abort.
   - Obtain $\{b_i\}_{i\in[\lambda]}$ from $\mathcal{C}^*_{\mathsf{Com}}$. Fix b^* to be the most frequently occuring bit in $\{b_i \oplus d_{i,1-c_i,0}\}_{i\in[\lambda]}$. Send $(\mathsf{Commit}, \mathsf{sid}, b^*)$ to the commitment ideal functionality.
2. **Reveal Phase.** $\mathcal{S}_{\mathsf{Rev}}$ does the following.
   (a) Obtain b from $\mathcal{C}^*_{\mathsf{Rev}}$. Additionally, for $i \in [\lambda]$,
   - Obtain α_i from $\mathcal{C}^*_{\mathsf{Rev}}$.
   - Execute the simulated Reveal phase where simulator is run on the part of the committer $\mathcal{C}^*_{\mathsf{Rev}}$ corresponding to session $(i, 1-c_i, \alpha_i)$. If $\mathcal{S}_{\mathsf{Rev}}$ outputs $(\mathsf{Reveal}, \mathsf{sid})$ and $d_{i,1-c_i,\alpha_i} = b \oplus b_i$, continue. Otherwise, abort.

   (b) Send $(\mathsf{Reveal}, \mathsf{sid})$ to the ideal functionality.
3. **Delete Phase.** If $\mathcal{C}^*$ makes a delete request, $\mathcal{S}_{\mathsf{Del}}$ sends $(\mathsf{DelRequest}, \mathsf{sid})$ to the ideal functionality, and upon obtaining $(\mathsf{DelResponse}, \mathsf{sid})$, it does the following.
   - If the Reveal phase was executed, then for every $i \in [\lambda]$, run the simulator on the part of $\mathcal{C}^*_{\mathsf{Del}}$ that interacts in the delete phase of session $(i, 1-c_i, 1-\alpha_i)$.
   - If the Reveal phase was not executed, then for every $i \in [\lambda]$, run the simulator on the part of $\mathcal{C}^*_{\mathsf{Del}}$ that interacts (sequentially) in the delete phases of sessions $(i, 1-c_i, 0)$ and $(i, 1-c_i, 1)$.

Analysis.

Note that there are a total of 4λ commitment sessions. Denote the real experiment by $\mathsf{Hybrid}_{0,1,1}$. For each $i \in [\lambda], j \in [0,1], k \in [0,1]$, define $\mathsf{Hybrid}_{i,j,k}$ to be the distribution obtained as follows.

Commit Phase. Set $\gamma = 1$, $\mathsf{DelReq} = 0$ and do the following:

1. If $\gamma = \lambda + 1$, obtain $\{b_i\}_i$ from $\mathcal{C}^*_{\mathsf{Com}}$ and end.
2. If $\gamma < i$,
   (a) Execute four sequential simulated Commit phases where the simulator for the commitment Com is run on the part of the committer $\mathcal{C}^*_{\mathsf{Com}}$ participating in each of the four sequential sub-sessions. Denote the bit output by the the simulator in each sub-session respectively by $(d_{\gamma,0,0}, d_{\gamma,0,1}, d_{\gamma,1,0}, d_{\gamma,1,1})$.
   (b) Sample and send choice bit $c_\gamma \leftarrow \{0,1\}$ to $\mathcal{C}^*_{\mathsf{Com}}$.
   (c) Execute two simulated Reveal phases where the simulator is run on the part of the committer $\mathcal{C}^*_{\mathsf{Com}}$ corresponding to sub-sessions $(\gamma, c_\gamma, 0)$ and $(\gamma, c_\gamma, 1)$. If the simulator outputs (Reveal, sid) for both sub-sessions and $d_{\gamma,c_\gamma,0} = d_{\gamma,c_\gamma,1}$, continue, and otherwise abort.
3. If $\gamma = i$,
   (a) Do the same as above (i.e., for the case $\gamma < i$) except execute simulated Commit (and if needed, Reveal) phases where the simulator for the commitment Com is run on the part of the committer $\mathcal{C}^*_{\mathsf{Com}}$ participating in sequential sub-sessions (γ, j', k') whenever $(j', k') \leq (j, k)$ where we have $(0,0) \leq (0,1) \leq (1,0) \leq (1,1)$ for transitive relation $\leq$. But for $(j', k') \not\leq (j, k)$, follow honest receiver strategy in sub-session (i, j', k').
4. If $\gamma > i$,
   (a) Execute honest receiver strategy for all Commit (and Reveal) phases for all sessions (i, j', k') for every $j', k' \in \{0,1\}^2$.
5. Set $\gamma = \gamma + 1$.

Reveal Phase. Do the following.

- Obtain b from $\mathcal{C}^*_{\mathsf{Rev}}$. Additionally, for $\gamma \in [\lambda]$,
  - Obtain α_γ from $\mathcal{C}^*_{\mathsf{Rev}}$.
  - If $\gamma < i$, execute the simulated Reveal phase where the simulator is run on the part of the committer $\mathcal{C}^*_{\mathsf{Rev}}$ corresponding to session $(\gamma, 1 - c_\gamma, \alpha_\gamma)$. If the simulator outputs (Reveal, sid) and if $d_{\gamma,1-c_\gamma,\alpha_\gamma} = b \oplus b_i$, continue. Otherwise, abort.
  - If $\gamma = i$, do the same as above when $(1 - c_\gamma, \alpha_\gamma) \leq (j, k)$ otherwise follow honest receiver strategy. If $\gamma > i$, follow honest receiver strategy.
- Set $b^* = b$.

Delete Phase. If $\mathcal{C}^*$ makes a delete request, send $(\mathsf{DelRequest}, \mathsf{sid})$ to the ideal functionality, and upon obtaining $(\mathsf{DelResponse}, \mathsf{sid})$, do the following.

- If the Reveal phase was executed, then

- For every $\gamma \in [1, i-1]$, run the simulator on the part of $\mathcal{C}^*_{\mathsf{Del}}$ that interacts in the delete phase of session $(\gamma, 1-c_\gamma, 1-\alpha_\gamma)$.
- For $\gamma = i$, if $(1-c_\gamma, 1-\alpha_\gamma) \leq (j,k)$, run the simulator on the part of $\mathcal{C}^*_{\mathsf{Del}}$ that interacts in the delete phase of session $(\gamma, 1-c_\gamma, 1-\alpha_\gamma)$. Otherwise run honest receiver strategy on session $(\gamma, 1-c_\gamma, 1-\alpha_\gamma)$.
- For $\gamma \in [i+1, \lambda]$, follow honest receiver strategy.

- If the Reveal phase was not executed, then
  - For every $\gamma \in [1, i-1]$, run the simulator on the part of $\mathcal{C}^*_{\mathsf{Del}}$ that interacts in the delete phases (sequentially) of $(\gamma, 1-c_\gamma, 0)$ and $(\gamma, 1-c_\gamma, 1)$.
  - For $\gamma = i$, run the simulator on the part of $\mathcal{C}^*_{\mathsf{Del}}$ that interacts in the delete phases (sequentially) of $(\gamma, 1-c_\gamma, b)$ for all $b \in \{0,1\}$ for which $(1-c_\gamma, b) \leq (j,k)$, and use honest receiver strategy on other sessions.
  - For $\gamma \in [i+1, \lambda]$, follow honest receiver strategy.

- Set $\mathsf{DelReq} = 1$.

The output of $\mathsf{Hybrid}_{i,j,k}$ is the final state of $\mathcal{C}^*$ together with the bit b^* (which is set to $\perp$ if the game aborted before b^* was set), and the bit DelReq.

We consider the interaction of $\mathcal{C}^*$ with an honest receiver, and denote the state output by $\mathcal{C}^*$ jointly with the bit output by the honest receiver in this interaction by $\mathsf{Hybrid}_{0,1,1}$. We now prove the following claim about consecutive hybrids.

Claim 5.18. *There exists a negligible function $\mu(\cdot)$ such that for every $i \in [\lambda]$, every $(\iota, j, k, \iota', j', k') \in \{(i-1,1,1,i,0,0), (i,0,0,i,0,1), (i,0,1,i,1,0), (i,1,0,i,1,1)\}$,*

- *for every QPT distinguisher $\mathcal{D}$,*

$$|\Pr[\mathcal{D}(\mathsf{Hybrid}_{\iota,j,k}) = 1] - \Pr[\mathcal{D}(\mathsf{Hybrid}_{\iota',j',k'}) = 1]| = \mu(\lambda)$$

- *and furthermore, for every unbounded distinguisher $\mathcal{D}$,*

$$|\Pr[\mathcal{D}(\mathsf{Hybrid}^{\mathsf{DelReq}=0}_{\iota,j,k}) = 1] - \Pr[\mathcal{D}(\mathsf{Hybrid}^{\mathsf{DelReq}=0}_{\iota',j',k'}) = 1]| = \mu(\lambda)$$

  where $\mathsf{Hybrid}^{\mathsf{DelReq}=0}_{\iota,j,k}$ is defined to be equal to $\mathsf{Hybrid}_{\iota,j,k}$ when DelReq is set to 0, and defined to be $\perp$ otherwise, and likewise for $\mathsf{Hybrid}^{\mathsf{DelReq}=0}_{\iota',j',k'}$.

Proof. Suppose this is not the case. Then there exists an adversarial QPT committer $\mathcal{C}^*$, a polynomial $p(\cdot)$, and an initial committer state $|\psi\rangle$ that corresponds to a state just before the beginning of commitment (ι', j', k') where for some QPT distinguisher $\mathcal{D}$,

$$\Pr[\mathcal{D}(\mathsf{Hybrid}_{\iota,j,k}) = 1] - \Pr[\mathcal{D}(\mathsf{Hybrid}_{\iota',j',k'}) = 1]| \geq \frac{1}{p(\lambda)}. \tag{1}$$

or for unbounded $\mathcal{C}^*$ and some unbounded distinguisher $\mathcal{D}'$,

$$|\Pr[\mathcal{D}'(\mathsf{Hybrid}^{\mathsf{DelReq}=0}_{\iota,j,k}) = 1] - \Pr[\mathcal{D}'(\mathsf{Hybrid}^{\mathsf{DelReq}=0}_{\iota',j',k'}) = 1]| \geq \frac{1}{p(\lambda)} \tag{2}$$

Consider a reduction/adversarial committer $\widetilde{\mathcal{C}}$ that obtains initial state $|\psi\rangle$, then internally runs $\mathcal{C}^*$, forwarding all messages between an external receiver and $\mathcal{C}^*$ for the $(\iota', j', k')^{th}$ commitment session, while running all other sessions according to the strategy in $\mathsf{Hybrid}_{\iota,j,k}$. The commit phase then ends, and $\widetilde{C}$ initiates the opening phase with the external receiver. Internally, it continues to run the remaining commit sessions with $\mathcal{C}^*$ – generating for it the messages on behalf of the receiver according to the strategy in $\mathsf{Hybrid}_{\iota,j,k}$. The only modification is that it forwards $\mathcal{C}^*$'s opening of the $(\iota', j', k')^{th}$ commitment (if and when it is executed) to the external challenger. Finally, $\widetilde{\mathcal{C}}$ behaves similarly if there is a delete phase, i.e., it forwards $\mathcal{C}^*$'s deletion request and any messages generated in the delete phase of the $(\iota', j', k')^{th}$ commitment between $\mathcal{C}^*$ and the external challenger.

Then, equation (1) and equation (2) respectively contradict the security of one-sided ideal commitments with EST against the committer $\mathcal{C}^*$ (Definition 5.7). More specifically, equation (1) contradicts the computationally secure realization of $\mathcal{F}_{\mathsf{Com}}^{\mathsf{Del}}$ whereas equation (2) contradicts the statistical security of Com against a corrupt committer that does not initiate deletion. This completes the proof of the claim. □

To complete the proof of the two lemmas, we observe that the only difference between $\mathsf{Hybrid}_{\lambda,1,1}$ and Ideal is the way the bit b^* (output by the honest receiver) is computed. In more detail, in $\mathsf{Hybrid}_{\lambda,1,1}$, the bit b^* is computed as the majority of $\{b_i \oplus d_{i,1-c_i,0}\}_{i\in[\lambda]}$. Now for every commitment strategy and every $i \in [\lambda]$, by correctness of extraction (which follows from the indistinguishability between real and ideal distributions for every commitment), the probability that $d_{i,1-c_i,0} \neq d_{i,1-c_i,1}$ and yet the receiver does not abort in Step 2(c) in the i^{th} sequential repetition, is $\leq \frac{1}{2} + \mathrm{negl}(\lambda)$. Thus, this implies that the probability that $\mathsf{Hybrid}_{\lambda,1,1}$ and Ideal output different bits b^* is at most $2^{-\lambda/2} + \mathrm{negl}(\lambda) = \mathrm{negl}(\lambda)$, which implies that the two are statistically close.

This, combined with the claim above, completes the proof. □

5.3.2 Security against a corrupt receiver

Lemma 5.19. *Protocol 5 computationally securely realizes $\mathcal{F}_{\mathsf{Com}}^{\mathsf{Del}}$ (Definition 5.1) against a corrupt R.*

Lemma 5.20. *Protocol 5 satisfies certified everlasting security against R. That is, for every QPT adversary $\{\mathcal{R}^*_\lambda := (\mathcal{R}^*_{\lambda,\mathsf{Com}}, \mathcal{R}^*_{\lambda,\mathsf{Rev}}, \mathcal{R}^*_{\lambda,\mathsf{Del}})\}_{\lambda\in\mathbb{N}}$ corrupting party R, there exists a QPT simulator $\{\mathcal{S}_\lambda := (\mathcal{S}_{\lambda,\mathsf{Com}}, \mathcal{S}_{\lambda,\mathsf{Rev}}, \mathcal{S}_{\lambda,\mathsf{Del}})\}_{\lambda\in\mathbb{N}}$ such that for any QPT environment $\{\mathcal{Z}_\lambda := (\mathcal{Z}_{\lambda,1}, \mathcal{Z}_{\lambda,2}, \mathcal{Z}_{\lambda,3})\}_{\lambda\in\mathbb{N}}$, and polynomial-size family of advice $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$,*

$$\mathsf{TD}\left(\Pi^{\mathsf{DelRes}=1}_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{R}^*_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle], \widetilde{\Pi}^{\mathsf{DelRes}=1}_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]\right) = \mathrm{negl}(\lambda),$$

where $\Pi^{\mathsf{DelRes}=1}_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{R}^*_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$ is defined to be equal to $\Pi_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{R}^*_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$ if the committer's output DelRes is set to 1, and defined to be $\perp$ otherwise, and likewise for $\widetilde{\Pi}^{\mathsf{DelRes}=1}_{\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$.

Proof. (of Lemmas 5.19 and 5.20)

The simulator. The first stage of the simulator $\mathcal{S}_{\mathsf{Com}}$, defined based on $\mathcal{R}^*_{\mathsf{Com}}$, will be obtained via the use of the Watrous rewinding lemma (Lemma 2.8) [Wat06]. For the purposes of defining the simulation strategy, it will be sufficient (w.l.o.g.) to consider a restricted receiver $\mathcal{R}^*_{\mathsf{Com}}$ that operates as follows in the i^{th} sequential step of the commitment phase of the protocol. In the simulation, the

state of this step of $\mathcal{R}^*_{\mathsf{Com}}$ will be initialized to the final state at the end of simulating the $(i-1)^{th}$ step.

1. $\mathcal{R}^*_{\mathsf{Com}}$ takes a quantum register W, representing its auxiliary quantum input. $\mathcal{R}^*_{\mathsf{Com}}$ will use two additional quantum registers that function as work space: V, which is an arbitrary (polynomial-size) register, and A, which is a single qubit register. The registers V and A are initialized to the all-zero state before the protocol begins.

2. Let M denote the polynomial-size register used by the committer C to send messages to $\mathcal{R}^*_{\mathsf{Com}}$. After carrying out step 2(a) by running on registers $(\mathsf{W}, \mathsf{V}, \mathsf{A}, \mathsf{M})$, $\mathcal{R}^*_{\mathsf{Com}}$ measures the register A to obtain a bit c_i for Step 2(b), which it sends back to C.

3. Next, $\mathcal{R}^*_{\mathsf{Com}}$ computes the reveal phases (with messages from C placed in register M) according to Step 2(c). $\mathcal{R}^*_{\mathsf{Com}}$ outputs registers $(\mathsf{W}, \mathsf{V}, \mathsf{A}, \mathsf{M})$.

Any QPT receiver can be modeled as a receiver of this restricted form followed by some polynomial-time post-processing of the restricted receiver's output. The same post-processing can be applied to the output of the simulator that will be constructed for the given restricted receiver.

Following [Wat06], we define a simulator that uses two additional registers, C and Z, which are both initialized to the all-zero state. C is a one qubit register, while Z is an auxiliary register used to implement the computation that will be described next. Consider a quantum procedure $\mathcal{S}_{\mathsf{partial}}$ that implements the strategy described in Protocol 6 using these registers.

Protocol 6

Circuit $\mathcal{S}_{\mathsf{partial}}$.

1. Sample a uniformly random classical bit $\widehat{c}$, and store it in register C.
2. Sample uniformly random bits (z, d).
3. If $\widehat{c} = 0$, initialize committer input as follows, corresponding to four sequential sessions:
   - For the first two sessions, set committer input to z.
   - For the third and fourth sessions, set committer input to d and $1-d$ respectively.
4. If $\widehat{c} = 1$, initialize committer input as follows, corresponding to four sequential sessions:
   - For the first and second sessions, set committer input to d and $1-d$ respectively.
   - For the last two sessions, set committer input to z.
5. Run the commitment phase interaction between the honest committer and $\mathcal{R}^*_{\mathsf{Com}}$'s sequence of unitaries on registers $(\mathsf{W}, \mathsf{V}, \mathsf{A}, \mathsf{M})$ initialized as above.
6. Measure the qubit register A to obtain a bit c. If $c = \widehat{c}$, output 0, otherwise output 1.

Figure 6: Partial Equivocal Simulator.

Next, we apply the Watrous rewinding lemma to the $\mathcal{S}_{\mathsf{partial}}$ circuit to obtain a circuit $\widehat{\mathcal{S}}_{\mathsf{partial}}$. To satisfy the premise of Lemma 2.8, we argue that the probability $p(|\psi\rangle)$ that $\mathcal{S}_{\mathsf{partial}}$ outputs 0 is such that $|p(|\psi\rangle) - \frac{1}{2}| = \mathsf{negl}(\lambda)$, regardless of the auxiliary input $|\psi\rangle$ to the i'th sequential stage of $\mathcal{R}^*_{\mathsf{Com}}$. This follows from the fact that the commitments are computationally hiding. In more detail, by definition, Step 5 produces a distribution on $\mathcal{R}^*_{\mathsf{Com}}$'s side that is identical to the distribution generated by $\mathcal{R}^*_{\mathsf{Com}}$ in its interaction with the committer, who either has input $(z, z, d, 1-d)$ (if $\widehat{c} = 0$) or input $(d, 1-d, z, z)$ (if $\widehat{c} = 1$). If $|p(|\psi\rangle) - \frac{1}{2}|$ were non-negligible, then the sequence of unitaries applied by $\mathcal{R}^*_{\mathsf{Com}}$ could be used to distinguish commitments generated according to the case $\widehat{c} = 0$ from commitments generated according to the case $\widehat{c} = 1$, which would contradict the hiding of the commitment.

Now consider the residual state on registers $(\mathsf{W}, \mathsf{V}, \mathsf{A}, \mathsf{M}, \mathsf{C}, \mathsf{Z})$ of $\mathcal{S}_{\mathsf{partial}}$ conditioned on a measurement of its output register A being 0. The output state of $\widehat{\mathcal{S}}_{\mathsf{partial}}$ will have negligible trace distance from the state on these registers. Now, the simulator $\mathcal{S}_{\mathsf{Com}}$ must further process this state as follows.

- Measure the register C, obtaining challenge c. Place the classical bits (c, d) in the register Z, which also contains the current state of the honest committer algorithm.
- Use information in register Z to execute Step 2(c) of Protocol 5.
- Discard register C, re-define register $\mathsf{Z}_i := \mathsf{Z}$ to be used later in the Reveal / Delete phases, and output registers $(\mathsf{W}, \mathsf{V}, \mathsf{A}, \mathsf{M})$ to be used in the next sequential step of the Commit phase.

The simulator $\mathcal{S}_{\mathsf{Com}}$ for the commit phase executes all λ sequential interactions in this manner, and then samples $b_1, \ldots, b_\lambda \leftarrow \{0,1\}^\lambda$, as the committer messages for Step 3 of Protocol 5. It then outputs the final state of $\mathcal{R}^*_{\mathsf{Com}}$ on registers $(\mathsf{W}, \mathsf{V}, \mathsf{A}, \mathsf{M})$, and additionally outputs a private state on registers $(\mathsf{Z}_1, \ldots, \mathsf{Z}_\lambda)$, which consist of the honest committer's state after each of the i sequential steps, as well as bits $(b_1, c_1, d_1, \ldots, b_\lambda, c_\lambda, d_\lambda)$.

The reveal stage of the simulator $\mathcal{S}_{\mathsf{Rev}}$ takes as input a bit b, and a state on registers $(\mathsf{Z}_1, \ldots, \mathsf{Z}_\lambda, \mathsf{W}, \mathsf{V}, \mathsf{A}, \mathsf{M})$, and does the following for each $i \in [\lambda]$.

- Let $\widehat{d_i} = b \oplus b_i$.
- If $c_i = 0$, it executes the decommitment phase for the $((\widehat{d_i} \oplus d_i) + 2)^{th}$ session with $\mathcal{R}^*_{\mathsf{Rev}}$.
- If $c_i = 1$, it executes the decommitment phase for the $(\widehat{d_i} \oplus d_i)^{th}$ session with $\mathcal{R}^*_{\mathsf{Rev}}$.
- Output $\mathcal{R}^*_{\mathsf{Rev}}$'s resulting state. Note that each decommitment will be to the bit $\widehat{d_i} = b \oplus b_i$.

Finally, the simulator $\mathcal{S}_{\mathsf{Del}}$ for the delete phase executes the honest committer's algorithm on the commitments that were not revealed above.

Analysis. Lemma 5.19 follows from the computational hiding of the underlying commitment scheme Com, via an identical proof to [BCKM21]. We have already argued above that the distribution produced by $\mathcal{S}_{\mathsf{Com}}$ is statistically close to the distribution that would result from conditioning on the output of $\mathcal{S}_{\mathsf{partial}}$ being 0 in each sequential step. Thus, it remains to argue that this is computationally indistinguishable from the real distribution. If not, then there exists a session $i \in [\lambda]$ such that the distribution in the real experiments up to the $i-1^{th}$ session is indistinguishable,

but up to the i^{th} session is distinguishable. However, this directly contradicts the computational hiding of the underlying commitment scheme.

In what follows, we prove Lemma 5.20. This only considers executions where $\mathsf{DelRes} = 1$, i.e., executions where $\mathcal{R}^*$ successfully completes the delete phase. We again consider a sequence of λ intermediate hybrids between the real and ideal executions. We will let $\mathsf{Hybrid}_0^{\mathsf{DelRes}=1}$ denote the final state of $\mathcal{R}^*$ in the real experiment when the honest party output $\mathsf{DelRes} = 1$ and $\perp$ otherwise. Let Hybrid_i denote the final state of $\mathcal{R}^*$ when the first i (out of λ) sequential commit sessions are simulated using the $\widehat{\mathcal{S}}_{\mathsf{partial}}$ circuit, defined based on $\mathcal{S}_{\mathsf{partial}}$ from Protocol 6. Let $\mathsf{Hybrid}_i^{\mathsf{DelRes}=1}$ denote the output of Hybrid_i when the honest party output $\mathsf{DelRes} = 1$ and $\perp$ otherwise.

For every $i \in [\lambda]$, statistical indistinguishability between $\mathsf{Hybrid}_{i-1}^{\mathsf{DelRes}=1}$ and $\mathsf{Hybrid}_i^{\mathsf{DelRes}=1}$ follows by a reduction to the certified everlasting security of Com (according to Definition 5.7), as follows. The reduction Red is different depending on whether or not the Reveal phase is executed.

- Case 1: The Reveal Phase is not executed. Red acts as receiver in one session of Com, interacting with an external challenger. Red samples a uniformly random bit d and sends it to the challenger. The challenger samples a uniformly random bit b'. If $b' = 0$, the challenger participates as a committer in a commit session to d and otherwise to $(1 - d)$.

  Red internally follows the strategy in Hybrid_{i-1} in the Commit phase for sessions $1, \ldots, i-1$ and $i+1, \ldots, \lambda$, based on the adversary $\mathcal{R}^*_{\mathsf{Com}}$. During the i^{th} session, Red interacts with the challenger and the adversary. In particular, it runs the strategy $\mathcal{S}_{\mathsf{partial}}$ from Protocol 6, with the following exception. For $\widehat{c}$ sampled uniformly at random, if $\widehat{c} = 0$, it forwards messages between $\mathcal{R}^*_{\mathsf{Com}}$ and the challenger for either the third or fourth commitment (sampled randomly) and commits to d in the other session and otherwise forwards messages between $\mathcal{R}^*_{\mathsf{Com}}$ and the challenger for either the first or second commitment (sampled randomly) and commits to d in the other session. If $\mathcal{R}^*_{\mathsf{Com}}$'s challenge $c_i = \widehat{c}$, Red continues the experiment, otherwise it aborts. Red continues to follow the strategy in Hybrid_{i-1}, except setting $b_i = b \oplus d$. Note that the challenge commitment is never opened.

  In the Delete phase, Red again follows the strategy in Hybrid_{i-1} except that it executes the Delete phase for the (two) unopened commitments in the i^{th} session, one that it generated on its own, and the other by forwarding messages between $\mathcal{R}^*_{\mathsf{Del}}$ and the external challenger.

  By computational hiding of the challenger's commitment, the probability that the reduction aborts is at most $\frac{1}{2} + \mathsf{negl}(n)$. Furthermore, conditioned on not aborting, the distribution output by Red is identical to $\mathsf{Hybrid}_{i-1}^{\mathsf{DelRes}=1}$ when $b' = 0$ and is statistically close to $\mathsf{Hybrid}_i^{\mathsf{DelRes}=1}$ when $b' = 1$ (the latter follows because the output of $\widehat{\mathcal{S}}_{\mathsf{partial}}$ and $\mathcal{S}_{\mathsf{partial}}$ conditioned on $c_i = \widehat{c}$ are statistically close, due to Watrous rewinding). Thus if $\mathsf{Hybrid}_{i-1}^{\mathsf{DelRes}=1}$ and $\mathsf{Hybrid}_i^{\mathsf{DelRes}=1}$ are not negligibly close in trace distance, Red breaks certified everlasting hiding of Com, as desired. Finally, we observe that Hybrid_λ is identical to the ideal experiment.

- Case 2: The Reveal Phase is executed. Red acts as receiver in one session of Com, interacting with an external challenger. Red samples a uniformly random bit d and sends it to the challenger. The challenger samples a uniformly random bit b'. If $b' = 0$, the challenger generates a commitment to d, and otherwise to $1 - d$.

  Red internally follows the strategy in Hybrid_{i-1} in the Commit phase for sessions $1, \ldots, i-1$ and $i+1, \ldots, \lambda$. For the i^{th} session Red runs the strategy $\mathcal{S}_{\mathsf{partial}}$ from Protocol 6, with the following exception. For bits $\widehat{c}, \widehat{b}$ sampled uniformly at random, it sets the commitment in

sub-session $(2\widehat{c}+1+\widehat{b})$ as the external commitment, and generates the commitment in sub-session $(2\widehat{c}+1+(1-\widehat{b}))$ as a commitment to d. It sets commitments in the remaining two sessions according to the strategy in Hybrid_{i-1}. If $\mathcal{R}^*_{\mathsf{Com}}$'s challenge $c_i = \widehat{c}$, Red continues the experiment, otherwise it aborts. Red continues to follow the strategy in Hybrid_{i-1}, except setting $b_i = b \oplus d$. Note that the challenge commitment is not opened in the Commit phase.

In the Reveal phase, Red behaves identically to Hybrid_{i-1} in sessions $(1,\ldots,i-1,i+1,\ldots,\lambda)$, and for session i it runs the Reveal phase of the commitment in sub-session $(2\widehat{c}+1+(1-\widehat{b}))$.

In the Delete phase, Red again follows the strategy of Hybrid_{i-1} except that it executes the Delete phase for the unopened commitments in the i^{th} session by forwarding messages between $\mathcal{R}^*_{\mathsf{Del}}$ and the external challenger. Thus, if $\mathsf{Hybrid}_{i-1}^{\mathsf{DelRes}=1}$ and $\mathsf{Hybrid}_{i}^{\mathsf{DelRes}=1}$ are not negligibly close in trace distance, Red breaks certified everlasting hiding of Com, as desired. Finally, we observe that $\mathsf{Hybrid}_{\lambda}$ is identical to the ideal experiment.

This completes the proof. □

5.4 Secure computation

In this section, we show that, following compilers in previous work, ideal commitments with EST imply oblivious transfer with EST and thus two-party computation of arbitrary functionalities with EST. Since prior compilers in the commitment hybrid model actually make use of "commitments with selective opening", we will first discuss this primitive, then describe a simple (deletion-composable) protocol that securely realizes commitments with selective opening. Next, we will invoke prior results [GLSV21] that together with our composition theorem imply secure two-party computation with EST.

Finally, we define the notion of multi-party computation with EST, and again show that it follows from ideal commitments with EST.

5.4.1 Two-party computation

We define the "commitment with selective opening" ideal functionality $\mathcal{F}_{\mathsf{so\text{-}com}}$ in Protocol 7, and we describe a simple (deletion-composable) protocol $\Pi^{\mathcal{F}^{\mathsf{Del}}_{\mathsf{com}}}$ that statistically securely realizes $\mathcal{F}^{\mathsf{Del}}_{\mathsf{so\text{-}com}}$.

- The committer, with input $(b_1,\ldots,b_{r(\lambda)})$, sequentially sends $(\mathsf{Commit}, i, b_i)$ to $\mathcal{F}^{\mathsf{Del}}_{\mathsf{Com}}$ for $i \in [r(\lambda)]$.
- The receiver, with input I, sends I to the committer.
- The committer sequentially sends (Reveal, i) for $i \in I$ to $\mathcal{F}^{\mathsf{Del}}_{\mathsf{com}}$.
- The receiver obtains output $\{(\mathsf{Reveal}, i, b_i)\}_{i\in I}$ from $\mathcal{F}^{\mathsf{Del}}_{\mathsf{com}}$.
- The parties perform the delete phase as follows.
  - If the committer is instructed to request a deletion, it sends $\{(\mathsf{DelRequest}, i)\}_{i\in[r(\lambda)]}$ to $\mathcal{F}^{\mathsf{Del}}_{\mathsf{com}}$, which are forwarded to the receiver.
  - If the receiver obtains *any* $(\mathsf{DelRequest}, i)$ for $i \in [r(\lambda)]$, it sets its output $\mathsf{DelReq} = 1$.
  - For each $(\mathsf{DelRequest}, i)$ obtained by the receiver, it sends $(\mathsf{DelResponse}, i)$ to $\mathcal{F}^{\mathsf{Del}}_{\mathsf{com}}$, which are forwarded to the committer.

Ideal functionality $\mathcal{F}_{\mathsf{so\text{-}com}}$

Parties: committer C and receiver R
Parameters: security parameter λ and function $r(\cdot)$

- Commit phase: $\mathcal{F}_{\mathsf{so\text{-}com}}$ receives a query $(\text{Commit}, \mathsf{sid}, b_1, \ldots, b_{r(\lambda)})$ from C, where each $b_i \in \{0,1\}$, records this query, and sends $(\text{Commit}, \mathsf{sid})$ to R.
- Reveal phase: $\mathcal{F}_{\mathsf{so\text{-}com}}$ receives a query $(\text{Reveal}, \mathsf{sid}, I)$ from R, where I is an index set of size $|I| \leq r(\lambda)$. $\mathcal{F}_{\mathsf{so\text{-}com}}$ ignores this message if no $(\text{Commit}, \mathsf{sid}, b_1, \ldots, b_{r(\lambda)})$ is recorded. Otherwise, $\mathcal{F}_{\mathsf{so\text{-}com}}$ records I and sends a message $(\text{Reveal}, \mathsf{sid}, \{b_i\}_{i \in I})$ to R, and a message $(\text{Choice}, \mathsf{sid}, I)$ to C.

Figure 7: Specification of the bit commitment with selective opening ideal functionality.

– If the committer obtains *all* $\{(\mathsf{DelResponse}, i)\}_{i \in [r(\lambda)]}$, it sets its output $\mathsf{DelRes} = 1$.

It is clear by definition that the above protocol statistically securely realizes $\mathcal{F}_{\mathsf{so\text{-}com}}^{\mathsf{Del}}$. Thus, by combining Imported Theorem 5.9, Theorem 5.10, and Theorem 5.15, which together show that there exists a protocol that realizes $\mathcal{F}_{\mathsf{com}}^{\mathsf{Del}}$ with EST assuming computationally-hiding statistically-binding commitments, and Theorem 5.4, which is our composition theorem, we obtain the following theorem.

Theorem 5.21. *There exists a protocol that securely realizes $\mathcal{F}_{\mathsf{so\text{-}com}}^{\mathsf{Del}}$ with EST that makes black-box use of a computationally-hiding statistically-binding commitment (Definition 4.24 and Definition 4.25).*

Note that it was necessary that our composition theorem handled reactive functionalities in order to establish this claim. Moreover, it is crucial in the definition of a reactive functionality with a deletion phase that we allow the deletion phase to be run after any phase of the reactive functionality. Indeed, in the above construction, some underlying commitments are not revealed but they still must be deleted.

Finally, it was shown in [GLSV21] (building on the work of [CK88, DFL+09], among others) that using quantum communication, it is possible to statistically realize the primitive of *oblivious transfer* in the $\mathcal{F}_{\mathsf{so\text{-}com}}$-hybrid model. Moreover, the work of [Kil88] showed how to statistically realize *arbitrary two-party computation* in the oblivious transfer hybrid model. Since it is straightfoward to make these protocols deletion-composable with a delete phase at the end, we have the following corollary.

Corollary 5.22. *Secure two-party computation of any polynomial-time functionality with Everlasting Security Transfer (Definition 5.5) exists, assuming only black-box use of a computationally-hiding statistically-binding commitment (Definition 4.24 and Definition 4.25).*

5.4.2 Multi-party computation

In order to define and construct multi-party computation with EST, we first have to specify a multi-party version of the Delete phase, which is described in Protocol 8.

Muti-party Deletion phase

Parties: $\{P_i\}_{i\in[n]}$

- Receive a sequence of queries $(\mathsf{DelRequest}, \mathsf{sid}, i, j)$ which indicate that party i is requesting party j to delete their data. For each such query received, record it and send $(\mathsf{DelRequest}, \mathsf{sid}, i, j)$ to party j.
- Receive a sequence of queries $(\mathsf{DelResponse}, \mathsf{sid}, i, j)$ which indicate that party i has deleted party j's data. For each such query received, if there does not exist a recorded $(\mathsf{DelRequest}, \mathsf{sid}, j, i)$, then ignore the query. Otherwise, send $(\mathsf{DelResponse}, \mathsf{sid}, i, j)$ to party j.

Figure 8: A specification of a generic multi-party deletion phase that can be added to any multi-party ideal functionality $\mathcal{F}$.

To define security, we first note that it is straightforward to extend the discussion on the "real-ideal paradigm" from Section 5.1 to handle multi-party protocols where the adversary may corrupt any subset $M \subset [n]$ of n parties. One can similarly generalize the definitions of computational and statistical secure realization (Definition 5.1 and Definition 5.2) to apply to multi-party protocols. Finally, we note that the multi-party Deletion phase introduced above adds $2(n-1)$ bits to each honest party i's output, which we denote by $\{\mathsf{DelReq}_{j\to i}, \mathsf{DelRes}_{j\to i}\}_{j\in[n]\setminus\{i\}}$, where each $\mathsf{DelReq}_{j\to i}$ indicates whether party j requested that party i delete its data, and each $\mathsf{DelRes}_{j\to i}$ indicates whether party j deleted party i's data. Now, we can generalize the notion of secure realization with EST (Definition 5.5) to the multi-party setting.

Definition 5.23 (Secure realization with Everlasting Security Transfer: Multi-party protocols)**.** *A protocol $\Pi_{\mathcal{F}}$ securely realizes the ℓ-phase n-party functionality $\mathcal{F}$ with EST if $\Pi_{\mathcal{F}}$ computationally securely realizes $\mathcal{F}^{\mathsf{Del}}$ (Definition 5.1) and the following holds. For every QPT adversary $\{\mathcal{A}_\lambda := (\mathcal{A}_{\lambda,1}, \ldots, \mathcal{A}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$ corrupting a subset of parties $M \subset [n]$, there exists a QPT simulator $\{\mathcal{S}_\lambda := (\mathcal{S}_{\lambda,1}, \ldots, \mathcal{S}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$ such that for any QPT environment $\{\mathcal{Z}_\lambda := (\mathcal{Z}_{\lambda,1}, \ldots, \mathcal{Z}_{\lambda,\ell})\}_{\lambda\in\mathbb{N}}$, and polynomial-size family of advice $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$,*

$$\mathsf{TD}\left(\Pi^{\mathsf{Del}_M=1}_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{A}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle], \widetilde{\Pi}^{\mathsf{Del}_M=1}_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]\right) = \mathrm{negl}(\lambda),$$

where $\Pi^{\mathsf{Del}_M=1}_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{A}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$ is defined to be equal to $\Pi_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{A}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$ if the bit $\mathsf{Del}_M = 1$ and defined to be $\perp$ otherwise, and likewise for $\widetilde{\Pi}^{\mathsf{Del}_M=1}_{\mathcal{F}^{\mathsf{Del}}}[\mathcal{S}_\lambda, \mathcal{Z}_\lambda, |\psi_\lambda\rangle]$. The bit Del_M is computed based on the honest party outputs, and is set to 1 if and only if for all $i \in [n]\setminus M$ and $j \in M$, $\mathsf{DelReq}_{j\to i} = 0$ and $\mathsf{DelRes}_{j\to i} = 1$.

Note that we here we did not include a designated party (or parties) against whom statistical security should hold by default (as in Definition 5.5), but in principle one could define security in

this manner. The definition as written in the multi-party case captures a type of *dynamic* statistical security property, where after the completion of the protocol, any arbitrary subset of parties can comply with a deletion request and certifiably remove information about the other party inputs from their view.

Finally, we can prove the following corollary of Theorem 5.21.

Corollary 5.24. *Secure multi-party computation of any polynomial-time functionality with Everlasting Security Transfer (Definition 5.23) exists, assuming only black-box use of a computationally-hiding statistically-binding commitment (Definition 4.24 and Definition 4.25).*

Proof. It was shown by [CvT95] that multi-party computation of any polynomial-time functionality can be statistically realized in the oblivious transfer hybrid model, where each pair of parties has access to an ideal oblivious transfer functionality. The stand-alone composition theorem (Theorem 5.4) shows that this is also true in the quantum setting, so it remains to argue that the resulting multi-party protocol can be made to satisfy security with EST, assuming that the underlying oblivious transfers do. This only requires extending the notion of deletion-composability to this setting, which can be achieved with the following deletion phase.

- If party i is instructed to issue a deletion request to party j, they issue deletion requests for *all* oblivious transfers that occurred between party i and j that were not already statistically secure against j.
- If party i obtains a deletion request from *any* one of the oblivious transfers between party i and party j, they output $\mathsf{DelReq}_{j\to i} = 1$.
- For each deletion request obtained by party i from party j, party i is instructed to send a deletion response to party j.
- If party i obtains a deletion response from party j for *all* oblivious transfers between party i and party j that were not already statistically secure against j, they output $\mathsf{DelRes}_{j\to i} = 1$.

This completes the proof. □

Acknowledgments

D.K. was supported in part by DARPA and NSF QIS award 2112890. This material is based on work supported by DARPA under Contract No. HR001120C0024. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA.

References

[ABKK22] Amit Agarwal, James Bartusek, Dakshita Khurana, and Nishant Kumar. A new framework for quantum oblivious transfer. *CoRR*, abs/2209.04520, 2022.

[AQY22] Prabhanjan Ananth, Luowen Qian, and Henry Yuen. Cryptography from pseudorandom quantum states. To appear in CRYPTO, 2022. https://ia.cr/2021/1663.

[BB84] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179, 1984.

[BCJL93] Gilles Brassard, Claude Crépeau, Richard Jozsa, and Denis Langlois. A quantum bit commitment scheme provably unbreakable by both parties. In *34th FOCS*, pages 362–371. IEEE Computer Society Press, November 1993.

[BCKM21] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. One-way functions imply secure computation in a quantum world. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 467–496, Cham, 2021. Springer International Publishing.

[BF10] Niek J. Bouman and Serge Fehr. Sampling in a quantum population, and applications. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 724–741. Springer, Heidelberg, August 2010.

[BGKR22] James Bartusek, Sanjam Garg, Dakshita Khurana, and Bhaskar Roberts. Functional cryptography with certified deletion. Personal Communication, 2022.

[BI20] Anne Broadbent and Rabib Islam. Quantum encryption with certified deletion. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography*, pages 92–122, Cham, 2020. Springer International Publishing.

[BMW98] Ingrid Biehl, Bernd Meyer, and Susanne Wetzel. Ensuring the integrity of agent-based computations by short proofs. In Kurt Rothermel and Fritz Hohl, editors, *Mobile Agents, Second International Workshop, MA'98, Stuttgart, Germany, September 1998, Proceedings*, volume 1477 of *Lecture Notes in Computer Science*, pages 183–194. Springer, 1998.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 97–106, 2011.

[Cal18] California Consumer Privacy Act (CCPA), 2018.

[CJJ21] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. Snargs for $\mathcal{P}$ from LWE. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 68–79. IEEE, 2021.

[CK88] Claude Crépeau and Joe Kilian. Achieving oblivious transfer using weakened security assumptions (extended abstract). In *29th FOCS*, pages 42–52. IEEE Computer Society Press, October 1988.

[CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 556–584. Springer, 2021.

[CvT95] Claude Crépeau, Jeroen van de Graaf, and Alain Tapp. Committed oblivious transfer and private multi-party computation. In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 110–123. Springer, Heidelberg, August 1995.

[CW19] Xavier Coiteux-Roy and Stefan Wolf. Proving erasure. In *IEEE International Symposium on Information Theory, ISIT 2019, Paris, France, July 7-12, 2019*, pages 832–836, 2019.

[DFL+09] Ivan Damgård, Serge Fehr, Carolin Lunemann, Louis Salvail, and Christian Schaffner. Improving the security of quantum protocols via commit-and-open. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 408–427. Springer, Heidelberg, August 2009.

[DGJ+20] Yfke Dulek, Alex B. Grilo, Stacey Jeffery, Christian Majenz, and Christian Schaffner. Secure multi-party quantum computation with a dishonest majority. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 729–758. Springer, Heidelberg, May 2020.

[DNS10] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Secure two-party quantum evaluation of unitaries against specious adversaries. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 685–706. Springer, Heidelberg, August 2010.

[Eur16] European Commission. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance), 2016.

[FM18] Honghao Fu and Carl A. Miller. Local randomness: Examples and application. *Phys. Rev. A*, 97:032324, Mar 2018.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 169–178, New York, NY, USA, 2009. Association for Computing Machinery.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013.

[GGV20] Sanjam Garg, Shafi Goldwasser, and Prashant Nalini Vasudevan. Formalizing data deletion in the context of the right to be forgotten. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 373–402. Springer, Heidelberg, May 2020.

[GLSV21] Alex B. Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in miniqcrypt. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the*

Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II, volume 12697 of *Lecture Notes in Computer Science*, pages 531–561. Springer, 2021.

[Got03] Daniel Gottesman. Uncloneable encryption. *Quantum Inf. Comput.*, 3:581–602, 2003.

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

[Hei27] W. Heisenberg. Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. *Zeitschrift fur Physik*, 43(3-4):172–198, March 1927.

[HMNY21] Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Quantum encryption with certified deletion, revisited: Public key, attribute-based, and classical communication. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021*, pages 606–636, Cham, 2021. Springer International Publishing.

[HMNY22a] Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Certified everlasting functional encryption. Cryptology ePrint Archive, Paper 2022/969, 2022. https://eprint.iacr.org/2022/969.

[HMNY22b] Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Certified everlasting zero-knowledge proof for QMA. CRYPTO, 2022. https://ia.cr/2021/1315.

[HSS11] Sean Hallgren, Adam Smith, and Fang Song. Classical cryptographic protocols in a quantum world. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 411–428. Springer, Heidelberg, August 2011.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *20th ACM STOC*, pages 20–31. ACM Press, May 1988.

[KM20] Dakshita Khurana and Muhammad Haris Mughees. On statistical security in two-party computation. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part II*, volume 12551 of *Lecture Notes in Computer Science*, pages 532–561. Springer, 2020.

[KR09] Yael Tauman Kalai and Ran Raz. Probabilistically checkable arguments. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 143–159. Springer, Heidelberg, August 2009.

[KT20] Srijita Kundu and Ernest Y. Z. Tan. Composably secure device-independent encryption with certified deletion, 2020.

[KTZ13] Jonathan Katz, Aishwarya Thiruvengadam, and Hong-Sheng Zhou. Feasibility and infeasibility of adaptively secure fully homomorphic encryption. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 14–31. Springer, Heidelberg, February / March 2013.

[LC97] Hoi-Kwong Lo and Hoi Fung Chau. Is quantum bit commitment really possible? *Physical Review Letters*, 78(17):3410, 1997.

[Lo97] Hoi-Kwong Lo. Insecurity of quantum secure computations. *Phys. Rev. A*, 56:1154–1162, Aug 1997.

[May97] Dominic Mayers. Unconditionally secure quantum bit commitment is impossible. *Physical review letters*, 78(17):3414, 1997.

[MS94] Dominic Mayers and Louis Salvail. Quantum oblivious transfer is secure against all individual measurements. In *Proceedings Workshop on Physics and Computation. PhysComp'94*, pages 69–77. IEEE, 1994.

[MY22] Tomoyuki Morimae and Takashi Yamakawa. Quantum commitments and signatures without one-way functions. To appear in CRYPTO, 2022. https://ia.cr/2021/1691.

[Nao90] Moni Naor. Bit commitment using pseudo-randomness. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 128–136. Springer, Heidelberg, August 1990.

[Por22] Alexander Poremba. Quantum proofs of deletion for learning with errors. Cryptology ePrint Archive, Report 2022/295, 2022. https://ia.cr/2022/295.

[RSW96] Ronald L. Rivest, Adi Shamir, and David Wagner. Time-lock puzzles and timed-release crypto. 1996.

[Unr13] Dominique Unruh. Everlasting multi-party computation. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 380–397. Springer, Heidelberg, August 2013.

[Unr14] Dominique Unruh. Revocable quantum timed-release encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 129–146. Springer, Heidelberg, May 2014.

[Wat06] John Watrous. Zero-knowledge against quantum attacks. In Jon M. Kleinberg, editor, *38th ACM STOC*, pages 296–305. ACM Press, May 2006.

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15:78–88, 1983.

[Win99] Andreas J. Winter. Coding theorem and strong converse for quantum channels. *IEEE Trans. Inf. Theory*, 45(7):2481–2485, 1999.

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd FOCS*, pages 160–164. IEEE Computer Society Press, November 1982.

[Yao95] Andrew Chi-Chih Yao. Security of quantum protocols against coherent measurements. In *27th ACM STOC*, pages 67–75. ACM Press, May / June 1995.

A Relation with [HMNY21]'s definitions

In this section, we prove that our definitions of certified deletion security for PKE and ABE imply prior definitions [HMNY21]. First, we reproduce the definitions in [HMNY21], albeit following our notational conventions, for the settings of public-key encryption and attribute-based encryption below.

Definition A.1 (Certified deletion security for PKE in [HMNY21])**.** $\mathsf{CD\text{-}PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Ver})$ *satisfies* certified deletion security *if for any non-uniform QPT adversary* $\mathcal{A} = \{\mathcal{A}_\lambda, |\psi\rangle_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\left|\Pr\left[\mathsf{C'\text{-}EXP}^{\mathcal{A}}_\lambda(0) = 1\right] - \Pr\left[\mathsf{C'\text{-}EXP}^{\mathcal{A}}_\lambda(1) = 1\right]\right| = \mathrm{negl}(\lambda),$$

where the experiment $\mathsf{C'\text{-}EXP}^{\mathcal{A}}_\lambda(b)$ *is defined as follows.*

- *Sample* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ *and* $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(\mathsf{pk}, b)$*.*
- *Initialize* $\mathcal{A}_\lambda(|\psi_\lambda\rangle)$ *with* pk *and* ct*.*
- *Parse* $\mathcal{A}_\lambda$*'s output as a deletion certificate* cert *and a left-over quantum state* ρ*.*
- *If* $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert}) = \top$*, set* $\mathsf{ret} = \mathsf{sk}$*, otherwise set* $\mathsf{ret} = \bot$*.*
- *Output* $\mathcal{A}_\lambda\ (\rho, \mathsf{ret})$*.*

Definition A.2 (Certified deletion security for ABE in [HMNY21])**.** $\mathsf{CD\text{-}ABE} = (\mathsf{Gen}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Ver})$ *satisfies* certified deletion security *if for any non-uniform QPT adversary* $\mathcal{A} = \{\mathcal{A}_\lambda, |\psi\rangle_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\left|\Pr\left[\mathsf{C'\text{-}EXP}^{\mathcal{A}}_\lambda(0) = 1\right] - \Pr\left[\mathsf{C'\text{-}EXP}^{\mathcal{A}}_\lambda(1) = 1\right]\right| = \mathrm{negl}(\lambda),$$

where the experiment $\mathsf{C'\text{-}EXP}^{\mathcal{A}}_\lambda(b)$ *is defined as follows.*

- *Sample* $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Gen}(1^\lambda)$ *and initialize* $\mathcal{A}_\lambda(|\psi_\lambda\rangle)$ *with* pk*.*
- *Set* $i = 1$*.*
- *If* $\mathcal{A}_\lambda$ *outputs a key query* P_i*, return* $\mathsf{sk}_{P_i} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, P_i)$ *to* $\mathcal{A}_\lambda$ *and set* $i = i + 1$*. This process can be repeated polynomially many times.*
- *If* $\mathcal{A}_\lambda$ *outputs an attribute* X^* *and a pair of messages* (m_0, m_1) *where* $P_i(X^*) = 0$ *for all predicates* P_i *queried so far, then compute* $(\mathsf{vk}, \mathsf{ct}) = \mathsf{Enc}(\mathsf{pk}, X^*, m_b)$ *and return* ct *to* $\mathcal{A}_\lambda$*, else exit and output* $\bot$*.*
- *If* $\mathcal{A}_\lambda$ *outputs a key query* P_i *such that* $P_i(X^*) = 0$*, return* $\mathsf{sk}_{P_i} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, P_i)$ *to* $\mathcal{A}_\lambda$ *(otherwise return* $\bot$*) and set* $i = i+1$*. This process can be repeated polynomially many times.*
- *Parse* $\mathcal{A}_\lambda$*'s output as a deletion certificate* cert *and a left-over quantum state* ρ*.*
- *If* $\mathsf{Ver}(\mathsf{vk}, \mathsf{cert}) = \top$ *set* $\mathsf{ret} = \mathsf{msk}$*, and otherwise set* $\mathsf{ret} = \bot$*. Send* ret *to* $\mathcal{A}_\lambda$*.*
- *Again, upto polynomially many times,* $\mathcal{A}_\lambda$ *sends key queries* P_i*. For each* i*, if* $P_i(X^*) = 0$*, return* $\mathsf{sk}_{P_i} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, P_i)$ *to* $\mathcal{A}_\lambda$ *(otherwise return* $\bot$*) and set* $i = i + 1$*. Finally,* $\mathcal{A}_\lambda$ *generates an output bit, which is set to be the output of* $\mathsf{C'\text{-}EXP}^{\mathcal{A}}_\lambda(b)$*.*

Claim A.3. *Any PKE scheme satisfying Definition 4.6 also satisfies Definition A.1.*

Proof. Suppose the claim is not true. Then there exists an adversary $\mathcal{A}$ and polynomial $p(\cdot)$ such that with respect to the notation in Definition A.1,

$$\left|\Pr\left[\mathsf{C'\text{-}EXP}_\lambda^{\mathcal{A}}(0)=1\right]-\Pr\left[\mathsf{C'\text{-}EXP}_\lambda^{\mathcal{A}}(1)=1\right]\right|=\frac{1}{p(\lambda)},$$

and yet for every adversary $\mathcal{B}$, with respect to the notation in Definition 4.6,

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}_\lambda^{\mathcal{B}}(0),\mathsf{EV\text{-}EXP}_\lambda^{\mathcal{B}}(1)\right)=\mathrm{negl}(\lambda), \tag{3}$$

and

$$\left|\Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{B}}(0)=1\right]-\Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{B}}(1)=1\right]\right|=\mathrm{negl}(\lambda),$$

Now we consider the following (efficient) reduction $\mathcal{R}=\{\mathcal{R}_\lambda\}_{\lambda\in\mathbb{N}}$ that acts as an adversary in the experiment $\mathsf{C\text{-}EXP}$. $\mathcal{R}_\lambda$ passes pk and ct to $\mathcal{A}_\lambda$, then passes the deletion certificate output by $\mathcal{A}_\lambda$ to its challenger and saves its residual state ρ. $\mathcal{R}_\lambda$ then obtains a verification outcome in $\{\bot,\top\}$ from the challenger. If the outcome is $\top$, $\mathcal{R}$ aborts, and otherwise $\mathcal{R}_\lambda$ outputs $\mathcal{A}_\lambda(\rho,\bot)$.

By equation (3), when the outcome is $\top$, the resulting state ρ in $\mathsf{C'\text{-}EXP}_\lambda^{\mathcal{A}}(b)$ is statistically independent of b (and in particular, since the distribution over sk is fixed by pk and is otherwise independent of ct, the state is also statistically independent given sk). Thus, except for a negligible loss, any advantage of $\mathcal{A}_\lambda$ can only manifest in the case when the output is $\bot$, which implies that

$$\left|\Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{R}}(0)=1\right]-\Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{R}}(1)=1\right]\right|=\frac{1}{p(\lambda)}-\mathrm{negl}(\lambda)>\frac{1}{2p(\lambda)},$$

a contradiction. □

Claim A.4. *Any ABE scheme satisfying Definition 4.15 also satisfies Definition A.2.*

Proof. Suppose the claim is not true. Then there exists an adversary $\mathcal{A}$ and polynomial $p(\cdot)$ such that with respect to the notation in Definition A.2,

$$\left|\Pr\left[\mathsf{C'\text{-}EXP}_\lambda^{\mathcal{A}}(0)=1\right]-\Pr\left[\mathsf{C'\text{-}EXP}_\lambda^{\mathcal{A}}(1)=1\right]\right|=\frac{1}{p(\lambda)},$$

and yet for every adversary $\mathcal{B}$,with respect to the notation in Definition 4.15,

$$\mathsf{TD}\left(\mathsf{EV\text{-}EXP}_\lambda^{\mathcal{B}}(0),\mathsf{EV\text{-}EXP}_\lambda^{\mathcal{B}}(1)\right)=\mathrm{negl}(\lambda), \tag{4}$$

and

$$\left|\Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{B}}(0)=1\right]-\Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{B}}(1)=1\right]\right|=\mathrm{negl}(\lambda),$$

Now we consider the following (efficient) reduction $\mathcal{R}=\{\mathcal{R}_\lambda\}_{\lambda\in\mathbb{N}}$ that acts as an adversary in the experiment $\mathsf{C\text{-}EXP}$. $\mathcal{R}_\lambda$ passes pk to $\mathcal{A}_\lambda$, then forwards all key queries of $\mathcal{A}_\lambda$ to its challenger, and forwards challenger responses back to $\mathcal{A}_\lambda$. Furthermore, it forwards any attribute X^* and pair of messages (m_0,m_1) output by $\mathcal{A}_\lambda$ to its challenger. Finally, it passes the deletion certificate output by $\mathcal{A}_\lambda$ to its challenger and saves its residual state ρ. $\mathcal{R}_\lambda$ then obtains a verification outcome in $\{\bot,\top\}$ from the challenger. If the outcome is $\top$, $\mathcal{R}$ aborts, and otherwise $\mathcal{R}_\lambda$ outputs $\mathcal{A}_\lambda(\rho,\bot)$.

By equation (4), when the outcome is $\top$, the resulting state ρ in $\mathsf{C'\text{-}EXP}_\lambda^{\mathcal{A}}(b)$ is statistically independent of b (and in particular, since the distribution over msk is fixed by pk and is otherwise independent of ct, the state is also statistically independent given msk). Thus, except for a negligible loss, any advantage of $\mathcal{A}_\lambda$ can only manifest in the case when the output is $\perp$, which implies that

$$\left| \Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{R}}(0) = 1\right] - \Pr\left[\mathsf{C\text{-}EXP}_\lambda^{\mathcal{R}}(1) = 1\right] \right| = \frac{1}{p(\lambda)} - \mathrm{negl}(\lambda) > \frac{1}{2p(\lambda)},$$

a contradiction. □