Weakening Assumptions for Publicly-Verifiable Deletion

James Bartusek[*] Dakshita Khurana[†] Giulio Malavolta[‡] Alexander Poremba[§]
Michael Walter[¶]

Abstract

We develop a simple compiler that generically adds publicly-verifiable deletion to a variety of cryptosystems. Our compiler only makes use of one-way functions (or one-way state generators, if we allow the public verification key to be quantum). Previously, similar compilers either relied on the use of indistinguishability obfuscation (Bartusek et. al., ePrint:2023/265) or almost-regular one-way functions (Bartusek, Khurana and Poremba, arXiv:2303.08676).

1 Introduction

Is it possible to *provably* delete information by leveraging the laws of quantum mechanics? An exciting series of recent works [Unr15, BI20, HMNY21, HMNY22a, HMNY22b, Por23, BK22, BGG+23, AKN+23, APV23, BKP23] built a variety of cryptosystems where an adversary holding a quantum ciphertext is able to *certifiably* delete an underlying plaintext.

The notion of certified deletion was formally introduced by Broadbent and Islam [BI20] for the one-time pad, where once the certificate is successfully verified, the plaintext remains hidden even if the secret (one-time pad) key is later revealed. This work has inspired a large body of research, aimed at understanding what kind of cryptographic primitives can be certifiably deleted. Recently, [BK22] built a compiler that generically adds the certified deletion property described above to any computationally secure commitment, encryption, attribute-based encryption, fully-homomorphic encryption, witness encryption or timed-release encryption scheme, *without making any additional assumptions*. Furthermore, it provides a strong *information-theoretic* deletion guarantee: Once an adversary generates a valid (classical) certificate of deletion, they cannot recover the plaintext that was previously computationally determined by their view even given *unbounded time*. However, the compiled schemes satisfy privately verifiable deletion – namely, an encryptor generates a ciphertext together with secret parameters, these parameters are necessary for verification and must be kept hidden from the adversary.

Publicly Verifiable Deletion. The above limitation was recently overcame in [BGG+23], which obtained *publicly-verifiable* deletion (PVD) for all of the above primitives as well as new ones, such as CCA encryption, obfuscation, maliciously-secure blind delegation and functional encryption[1].

[*]UC Berkeley. Email: bartusek.james@gmail.com
[†]UIUC. Email: dakshita@illinois.edu
[‡]Max Planck Institute for Security and Privacy. Email: giulio.malavolta@hotmail.it
[§]Caltech. Email: aporemba@caltech.edu
[¶]Ruhr-Universität Bochum. michael.walter@rub.de

[1]A concurrent updated version of [HMNY22a] also obtained functional encryption with certified deletion, although in the private-verification settings.

However, the compilation process proposed in [BGG+23] required the strong notion of indistinguishability obfuscation, regardless of what primitive one starts from. This was later improved in [BKP23], which built commitments with PVD from injective (or almost-regular) one-way functions, and X encryption with PVD for $X \in$ {attribute-based, quantum fully-homomorphic, witness, time-revocable}, assuming X encryption and trapdoored variants of injective (or almost-regular) one-way functions.

Weakening Assumptions for PVD. Given this state of affairs, it is natural to ask whether one can further relax the assumptions underlying publicly verifiable deletion, essentially matching what is known in the private verification setting. In this work, we show that the injectivity/regularity constraints on the one-way functions from prior work [BKP23] are not necessary to achieve publicly-verifiable deletion; *any* one-way function suffices, or even a quantum weakening called a one-way state generator (OWSG) [MY22] if we allow the verification key to be quantum. Kretschmer [Kre21] showed that, relative to an oracle, pseudorandom state generators (PRSGs) [JLS18, MY22] exist even if $\mathsf{BQP} = \mathsf{QMA}$ (and thus $\mathsf{NP} \subseteq \mathsf{BQP}$). Because PRSGs are known to imply OWSGs [MY22], this allows us to base our generic compiler for PVD on something potentially even weaker than the existence of one-way functions.

In summary, we improve [BKP23] to obtain X with PVD for $X \in$ {statistically-binding commitment, public-key encryption, attribute-based encryption, fully-homomorphic encryption, quantum fully-homomorphic encryption, witness encryption, timed-release encryption}, assuming only X and any one-way function. We also obtain X with PVD for all the X above, assuming only X and any one-way state generator [MY22], but with a *quantum* verification key. Our primary contribution is conceptual: Our construction is inspired by a recent work on quantum-key distribution [MW23], which we combine with a proof strategy that closely mimics [BKP23] (which in turn builds on the proof techniques of [BK22]).

1.1 Technical Outline

We begin be recalling that prior work [BKP23] observed that, given an appropriate *two-to-one* one-way function f, a commitment (with certified deletion) to a bit b can be

$$\mathsf{ComCD}(b) \propto \Big(y, |x_0\rangle + (-1)^b |x_1\rangle\Big)$$

where $(0, x_0), (1, x_1)$ are the two pre-images of (a randomly sampled) image y. Given an image y and a quantum state $|\psi\rangle$, they showed that any pre-image of y constitutes a valid certificate of deletion of the bit b. This certificate can be obtained by measuring the state $|\psi\rangle$ in the computational basis.

Furthermore, it was shown in [BKP23] that in fact two-to-one functions are not needed to instantiate this template, it is possible to use more general types of one-way functions to obtain a commitment of the form

$$\mathsf{ComCD}(b) \propto \left(y, \sum_{x: f(x)=y, M(x)=0} |x\rangle + (-1)^b \sum_{x: f(x)=y, M(x)=1} |x\rangle\right).$$

where M denotes some binary predicate applied to the preimages of y. The work of [BKP23] developed techniques to show that this satisfies certified deletion, as well as binding as long as the sets

$$\sum_{x:f(x)=y,M(x)=0} |x\rangle \quad \text{and} \quad \sum_{x:f(x)=y,M(x)=1} |x\rangle$$

are somewhat "balanced", i.e. for a random image y and the sets $S_0 = \{x : f(x) = y, M(x) = 0\}$ and $S_1 = \{x : f(x) = y, M(x) = 1\}$, it holds that $\frac{|S_0|}{|S_1|}$ is a fixed constant. Such "balanced" functions can be obtained from injective (or almost-regular) one-way functions by a previous result of [HHK⁺09].

Our first observation is that it is not necessary to require x_0, x_1 to be preimages of the same image y. Instead, we can modify the above template to use randomly sampled x_0, x_1, compute $y_0 = F(x_0), y_1 = F(x_1)$, to obtain

$$\mathsf{ComCD}(b) \propto \Big((y_0, y_1), |x_0\rangle + (-1)^b |x_1\rangle\Big)$$

Unfortunately, as described, the phase b may be statistically hidden when F is a general one-way function (in particular, F does not need to be injective), which implies that the commitment above is not binding. To restore binding, we can simply append a commitment to $(x_0 \oplus x_1)$ to the state above, resulting in

$$\mathsf{ComCD}(b) \propto \Big((y_0, y_1), \mathsf{Com}(x_0 \oplus x_1), |x_0\rangle + (-1)^b |x_1\rangle\Big)$$

Assuming that Com is statistically binding, the bit b is (statistically) determined by the commitment state above, and in fact, can even be efficiently determined given $x_0 \oplus x_1$. This is because a measurement of $|x_0\rangle + (-1)^b |x_1\rangle$ in the Hadamard basis yields a string w such that $b = (x_0 \oplus x_1) \cdot w$.

Certified deletion security for this template follows immediately from the technique in [BKP23] (who themselves build on [BK22]), but we discuss it for completeness below. We consider an experiment where the adversary is given an encryption of b and outputs a deletion certificate. If the certificate is valid, the output of the experiment is defined to be the adversary's left-over state, otherwise is it defined to be $\bot$.

Our first step is to defer the dependence of the experiment on the bit b. In more detail, we will instead imagine sampling the distribution by guessing a uniformly random $c \leftarrow \{0, 1\}$, and initializing the adversary with $((y_0, y_1), \mathsf{Com}(x_0 \oplus x_1), |x_0\rangle + (-1)^c |x_1\rangle)$. The challenger later obtains input b and aborts the experiment (outputs $\bot$) if $c \neq b$. Since c was a uniformly random guess, the trace distance between the $b = 0$ and $b = 1$ outputs of this modified experiment is at least half the trace distance between the outputs of the original experiment. Moreover, we can further delay the process of obtaining input b, and then abort or not until *after* the adversary outputs a certificate of deletion. That is, we can consider a *purification* where a register C is initialized in a superposition $|0\rangle + |1\rangle$ of two choices for c, and is later measured to determine bit c.

Once the dependence of the experiment on b has been deferred, as above, we can consider another experiment where the challenger measures the superposition $|x_0\rangle + (-1)^c |x_1\rangle$ *before* sending it to the adversary $\mathcal{A}$. Intuitively, performing this measurement *removes* information about b from $\mathcal{A}$'s view in a manner that is computationally undetectable by $\mathcal{A}$. Observe that the ancilla register C is *unentangled* with the rest of the experiment. In fact, the ancilla register is exactly $|+\rangle$ when we give the adversary $|x_0\rangle$, and $|-\rangle$ when we give the adversary $|x_1\rangle$ (along with the remaining auxiliary information). By the semantic security of Com and the one-wayness of F, finding x_1 given x_0 is hard (and vice versa), which means that the following event must always hold:

When the adversary outputs a valid certificate γ, a projection of the pre-image register onto $|+\rangle$ succeeds if $\gamma = x_0$, and a projection of the pre-image register onto $|-\rangle$ succeeds if $\gamma = x_1$.

Finally, we can use the fact (used, e.g., in [DS22]) that distinguishing between mixtures and superpositions is as hard as swapping between x_0 and x_1 to show that the same event must also hold (except with negligible probability) when the register containing the superposition $|x_0\rangle + (-1)^c |x_1\rangle$ is *not* measured prior to sending it to $\mathcal{A}_1$. Because this event always holds, it is possible to show that measuring C in the computational basis results in a uniformly random and independent c. By definition of the experiment (abort when $b \neq c$, continue otherwise), this implies that the bit b is set in a way that is uniformly random and independent of the adversary's view after deletion, giving us the guarantee we desire.

We note that encryption with PVD can be obtained similarly by committing to each bit of the plaintext as

$$\mathsf{EncCD}(b) \propto \Big((y_0, y_1), \mathsf{Enc}(x_0 \oplus x_1), |x_0\rangle + (-1)^b |x_1\rangle\Big)$$

We also note that, following prior work [BK22, Por23, BKP23], a variety of encryption schemes (e.g., ABE, FHE, witness encryption) can be plugged into the above template to obtain the respective encryption schemes with PVD.

2 Preliminaries

Let λ denote the security parameter. We write $\mathrm{negl}(\cdot)$ 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 $n > N$, $f(n) < n^{-c}$.

A finite-dimensional complex Hilbert space is denoted by $\mathcal{H}$, and we use subscripts to distinguish between different systems (or registers); for example, we let $\mathcal{H}_\mathsf{A}$ be the Hilbert space corresponding to a system A. The tensor product of two Hilbert spaces $\mathcal{H}_\mathsf{A}$ and $\mathcal{H}_\mathsf{B}$ is another Hilbert space denoted by $\mathcal{H}_{\mathsf{AB}} = \mathcal{H}_\mathsf{A} \otimes \mathcal{H}_\mathsf{B}$. We let $\mathcal{L}(\mathcal{H})$ denote the set of linear operators over $\mathcal{H}$. A quantum system over the 2-dimensional Hilbert space $\mathcal{H} = \mathbb{C}^2$ is called a *qubit*. For $n \in \mathbb{N}$, we refer to quantum registers over the Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ as n-qubit states. We use the word *quantum state* to refer to both pure states (unit vectors $|\psi\rangle \in \mathcal{H}$) and density matrices $\rho \in \mathcal{D}(\mathcal{H})$, where we use the notation $\mathcal{D}(\mathcal{H})$ to refer to the space of positive semidefinite linear operators of unit trace acting on $\mathcal{H}$. The *trace distance* of two density matrices $\rho, \sigma \in \mathcal{D}(\mathcal{H})$ is given by

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

A quantum channel $\Phi : \mathcal{L}(\mathcal{H}_\mathsf{A}) \to \mathcal{L}(\mathcal{H}_\mathsf{B})$ is a linear map between linear operators over the Hilbert spaces $\mathcal{H}_\mathsf{A}$ and $\mathcal{H}_\mathsf{B}$. We say that a channel Φ is *completely positive* if, for a reference system R of arbitrary size, the induced map $I_\mathsf{R} \otimes \Phi$ is positive, and we call it *trace-preserving* if $\mathsf{Tr}[\Phi(X)] = \mathsf{Tr}[X]$, for all $X \in \mathcal{L}(\mathcal{H})$. A quantum channel that is both completely positive and trace-preserving is called a quantum CPTP channel.

A *unitary* $U : \mathcal{L}(\mathcal{H}_\mathsf{A}) \to \mathcal{L}(\mathcal{H}_\mathsf{A})$ is a special case of a quantum chanel that satisfies $U^\dagger U = UU^\dagger = I_\mathsf{A}$. 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 $\sum_i \Pi_i = I$.

A quantum polynomial-time (QPT) machine is a polynomial-time family of quantum circuits given by $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$, where each circuit $\mathcal{A}_\lambda$ is described by a sequence of unitary gates and measurements; moreover, for each $\lambda \in \mathbb{N}$, there exists a deterministic polynomial-time Turing machine that, on input 1^λ, outputs a circuit description of $\mathcal{A}_\lambda$.

Imported Theorem 2.1 (Distinguishing implies Mapping [DS22]). *Let* D *be a projector,* Π_0, Π_1 *be orthogonal projectors, and* $|\psi\rangle \in \mathsf{Im}(\Pi_0 + \Pi_1)$*. Then,*

$$\|\Pi_1 \mathsf{D} \Pi_0 |\psi\rangle\|^2 + \|\Pi_0 \mathsf{D} \Pi_1 |\psi\rangle\|^2 \geq \frac{1}{2}\left(\|\mathsf{D}|\psi\rangle\|^2 - \left(\|\mathsf{D}\Pi_0|\psi\rangle\|^2 + \|\mathsf{D}\Pi_1|\psi\rangle\|^2\right)\right)^2.$$

3 Main Theorem

Theorem 3.1. *Let* $F : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ *be a one-way function secure against QPT adversaries. Let* $\{\mathcal{Z}_\lambda(\cdot)\}_{\lambda\in\mathbb{N}}$ *be a static or interactive*[2] *distribution with an* $n(\lambda)$*-bit argument, such that for all* $z \in \{0,1\}^{n(\lambda)}$ *and QPT adversary* $\{\mathcal{A}_\lambda\}_\lambda$ *it holds that*

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

if $\mathcal{Z}_\lambda(\cdot)$ *is a static distribution, or*

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

if $\mathcal{Z}_\lambda(\cdot)$ *is an interactive distribution.*

Now, for any QPT adversary $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$*, 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* $x_0, x_1 \leftarrow \{0,1\}^{n(\lambda)}$*, define* $y_0 = F(x_0), y_1 = F(x_1)$ *and initialize* $\mathcal{A}_\lambda$ *with*[3]

$$\mathcal{Z}_\lambda(x_0 \oplus x_1), y_0, y_1, \frac{1}{\sqrt{2}}\left(|x_0\rangle + (-1)^b |x_1\rangle\right).$$

- $\mathcal{A}_\lambda$*'s output is parsed as a string* $x' \in \{0,1\}^{n(\lambda)}$ *and a residual state on register* A'*.*
- *If* $F(x') \in \{y_0, y_1\}$*, then output* A'*, and otherwise output* $\bot$*.*

Then,

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

Proof. We define a sequence of hybrids.

- $\mathsf{Hyb}_0(b)$: This is the distribution $\left\{\widetilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(b)\right\}_{\lambda\in\mathbb{N}, b\in\{0,1\}}$ described above.
- $\mathsf{Hyb}_1(b)$: This distribution is sampled as follows.
  - Sample $x_0, x_1, y_0 = F(x_0), y_1 = F(x_1)$, prepare the state

$$\frac{1}{2}\sum_{c\in\{0,1\}} |c\rangle_{\mathsf{C}} \otimes \left(|x_0\rangle + (-1)^c |x_1\rangle\right)_{\mathsf{A}},$$

  and initialize $\mathcal{A}_\lambda$ with

$$\mathcal{Z}_\lambda(x_0 \oplus x_1), y_0, y_1, \mathsf{A}.$$

[2] By interactive, we mean that $\mathcal{Z}_\lambda(\cdot)$ is the description of an interactive machine.

[3] In the case that $\mathcal{Z}_\lambda(\cdot)$ is interactive, $\mathcal{A}_\lambda$ is given access to the interactive machine $\mathcal{Z}_\lambda(x_0 \oplus x_1)$.

– $\mathcal{A}_\lambda$'s output is parsed as a string $x' \in \{0,1\}^{n(\lambda)}$ and a residual state on register A'.

– If $F(x') \notin \{y_0, y_1\}$, then output $\bot$. Next, measure register C in the computational basis and output $\bot$ if the result is $1-b$. Otherwise, output A'.

- $\mathsf{Hyb}_2(b)$: This distribution is sampled as follows.

  – Sample $x_0, x_1, y_0 = F(x_0), y_1 = F(x_1)$, prepare the state

  $$\frac{1}{2}\sum_{c\in\{0,1\}} |c\rangle_{\mathsf{C}} \otimes (|x_0\rangle + (-1)^c |x_1\rangle)_{\mathsf{A}},$$

  and initialize $\mathcal{A}_\lambda$ with

  $$\mathcal{Z}_\lambda(x_0 \oplus x_1), y_0, y_1, \mathsf{A}.$$

  – $\mathcal{A}_\lambda$'s output is parsed as a string $x' \in \{0,1\}^{n(\lambda)}$ and a residual state on register A'.

  – If $F(x') \notin \{y_0, y_1\}$, then output $\bot$. Next, let $c' \in \{0,1\}$ be such that $F(x') = y_{c'}$, measure register C in the Hadamard basis, and output $\bot$ if the result is $1-c'$. Next, measure register C in the computational basis and output $\bot$ if the result is $1-b$. Otherwise, output A'.

Next, we define a hybrid Hyb_2' that will be convenient for the sake of the proof.

- $\mathsf{Hyb}_2'(b)$: This distribution is sampled as follows.

  – Sample $x_0, x_1, y_0 = F(x_0), y_1 = F(x_1)$, prepare the state

  $$\frac{1}{2}\sum_{c\in\{0,1\}} |c\rangle_{\mathsf{C}} \otimes (|x_0\rangle + (-1)^c |x_1\rangle)_{\mathsf{A}},$$

  measure register A in the computational basis, and initialize $\mathcal{A}_\lambda$ with

  $$\mathcal{Z}_\lambda(x_0 \oplus x_1), y_0, y_1, \mathsf{A}.$$

  – $\mathcal{A}_\lambda$'s output is parsed as a string $x' \in \{0,1\}^{n(\lambda)}$ and a residual state on register A'.

  – If $F(x') \notin \{y_0, y_1\}$, then output $\bot$. Next, let $c' \in \{0,1\}$ be such that $F(x') = y_{c'}$, measure register C in the Hadamard basis, and output $\bot$ if the result is $1-c'$. Next, measure register C in the computational basis and output $\bot$ if the result is $1-b$. Otherwise, output A'.

We define $\mathsf{Advt}(\mathsf{Hyb}_i) \coloneqq \mathsf{TD}\left(\mathsf{Hyb}_i(0), \mathsf{Hyb}_i(1)\right)$. Let $E_2'(b)$ be the event that $\mathsf{Hyb}_2'(b)$ outputs $\bot$ after measuring register C in the Hadamard basis and obtaining outcome $1-c'$, and let $E_2(b)$ be the corresponding event in $\mathsf{Hyb}_2(b)$. To complete the proof, we show the following sequence of claims.

Claim 3.2. *For any $b \in \{0,1\}$, $\Pr[E_2'(b)] = \mathrm{negl}(\lambda)$.*

Proof. First, note that the state of the system after the measurement of register A is the mixture

$$\frac{1}{2}|+\rangle\langle+|_{\mathsf{C}} \otimes |x_0\rangle\langle x_0|_{\mathsf{A}} + \frac{1}{2}|-\rangle\langle-|_{\mathsf{C}} \otimes |x_1\rangle\langle x_1|_{\mathsf{A}}.$$

Thus, the event $E_2'(b)$ only occurs if the adversary outputs an x' such that $F(x') = y_{1-c'}$, given input $\mathcal{Z}_\lambda(x_0 \oplus x_1), y_0, y_1, x_{c'}$. But this can only occur with negligible probability due to the semantic security of $\mathcal{Z}_\lambda(\cdot)$ and the one-wayness of F. $\square$

Claim 3.3. *For any* $b \in \{0,1\}$, $\Pr[E_2(b)] = \text{negl}(\lambda)$.

Proof. By Claim 3.2, it suffices to show that the difference between these events in Hyb_2 and Hyb_2' is negligible. The only difference between these hybrids is the measurement of register A before the adversary is initialized. At a high level, Imported Theorem 2.1 and the semantic security of $\mathcal{Z}_\lambda$ imply that any adversary that can distinguish whether or not register A is measured can be used to invert the one-way function with noticeable probability. Details follow.

First note that by the semantic security of $\mathcal{Z}_\lambda$, no QPT adversary can distinguish $\mathcal{Z}_\lambda(x_0 \oplus x_1), y_0, y_1, \mathsf{A}$ from $\mathcal{Z}_\lambda(0^{n(\lambda)}), y_0, y_1, \mathsf{A}$. Furthermore, note that by Imported Theorem 2.1 and the one-wayness of F, it is impossible for a QPT adversary given (y_0, y_1) and register A initialized with either the superposition state

$$\frac{1}{\sqrt{2}}\left(|x_0\rangle + |x_1\rangle\right)$$

or the mixture

$$\frac{1}{2}|x_0\rangle\langle x_0| + \frac{1}{2}|x_1\rangle\langle x_1|,$$

to predict with better than $\frac{1}{2} + \text{negl}(\lambda)$ advantage whether A contains a mixture or a superposition. That is, we can define

$$\Pi_0 \coloneqq \sum_{x:F(x)=y_0} |x\rangle\langle x|, \quad \Pi_1 \coloneqq \sum_{x:F(x)=y_1} |x\rangle\langle x|$$

and see that any adversary with noticeable advantage in distinguishing the measurement (RHS of Imported Theorem 2.1) implies an adversary that is just given (y_0, y_1) and x_b for some $b \in \{0,1\}$, and outputs an x_{1-b} such that $F(x_{1-b}) = y_{1-b}$ (LHS of Imported Theorem 2.1).

Next, consider a reduction $\mathcal{R}$ that obtains (y_0, y_1) and register A from the challenger, where A contains either the superposition or has been measured to obtain the mixture as above. $\mathcal{R}$ first initializes register C to the $|+\rangle$ state. Then, it defines the function $G_{y_1} : \{0,1\}^{1+n(\lambda)} \to \{0,1\}$ to map (b, x) to 1 iff $b = 1$ and $F(x) = y_1$, and applies G_{y_1} as a phase oracle on registers (C, A).

Depending on whether A initially contained the superposition or the mixture, $\mathcal{R}$ ends up with either the superposition state

$$\frac{1}{2}\sum_{c\in\{0,1\}} |c\rangle_{\mathsf{C}} \otimes \left(|x_0\rangle + (-1)^c |x_1\rangle\right)_{\mathsf{A}}$$

or the mixture

$$\frac{1}{2}|+\rangle\langle+|_{\mathsf{C}} \otimes |x_0\rangle\langle x_0|_{\mathsf{A}} + \frac{1}{2}|-\rangle\langle-|_{\mathsf{C}} \otimes |x_1\rangle\langle x_1|_{\mathsf{A}}.$$

It then initializes $\mathcal{A}_\lambda$ with $\mathcal{Z}_\lambda(0), y_0, y_1, \mathsf{A}$, and then completes the rest of the game according to $\mathsf{Hyb}_2(b)$, checking whether or not the event $E_2(b)$ occurred. Any noticeable difference between $E_2(b)$ and $E_2'(b)$ would imply that $\mathcal{R}$ distiguishes whether or not register A was measured with noticeable probability, a contradiction. □

Claim 3.4. $\mathsf{Advt}(\mathsf{Hyb}_1) = \text{negl}(\lambda)$.

Proof. This follows directly from Claim 3.3 and the fact that $\mathsf{Advt}(\mathsf{Hyb}_2) = 0$. □

Claim 3.5. $\mathsf{Advt}(\mathsf{Hyb}_0) = \text{negl}(\lambda)$.

Proof. This follows because $\mathsf{Hyb}_1(b)$ is identically distributed to the distribution that outputs $\bot$ with probability 1/2 and otherwise outputs $\mathsf{Hyb}_0(b)$, so the advantage of Hyb_0 is at most double the advantage of Hyb_1. □

□

4 Cryptography with Publicly-Verifiable Deletion

Let us now introduce some formal definitions. A public-key encryption (PKE) scheme with publicly-verifiable deletion (PVD) has the following syntax.

- $\mathsf{PVGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$: the key generation algorithm takes as input the security parameter λ and outputs a public key pk and secret key sk.
- $\mathsf{PVEnc}(\mathsf{pk}, b) \to (\mathsf{vk}, |\mathsf{ct}\rangle)$: the encryption algorithm takes as input the public key pk and a plaintext b, and outputs a (public) verification key vk and a ciphertext $|\mathsf{ct}\rangle$.
- $\mathsf{PVDec}(\mathsf{sk}, |\mathsf{ct}\rangle) \to b$: the decryption algorithm takes as input the secret key sk and a ciphertext $|\mathsf{ct}\rangle$ and outputs a plaintext b.
- $\mathsf{PVDel}(|\mathsf{ct}\rangle) \to \pi$: the deletion algorithm takes as input a ciphertext $|\mathsf{ct}\rangle$ and outputs a deletion certificate π.
- $\mathsf{PVVrfy}(\mathsf{vk}, \pi) \to \{\top, \bot\}$: the verify algorithm takes as input a (public) verification key vk and a proof π, and outputs $\top$ or $\bot$.

Definition 4.1 (Correctness of deletion)**.** *A PKE scheme with PVD satisfies* correctness of deletion *if for any b, it holds with $1-\text{negl}(\lambda)$ probability over* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PVGen}(1^\lambda)$, $(\mathsf{vk}, |\mathsf{ct}\rangle) \leftarrow \mathsf{PVEnc}(\mathsf{pk}, b)$, $\pi \leftarrow \mathsf{PVDel}(|\mathsf{ct}\rangle)$, $\mu \leftarrow \mathsf{PVVrfy}(\mathsf{vk}, \pi)$ *that* $\mu = \top$.

Definition 4.2 (Certified deletion security)**.** *A PKE scheme with PVD satisfies* certified deletion security *if it satisfies standard semantic security, and moreover, for any QPT adversary $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\mathsf{TD}\left(\mathsf{EvPKE}_{\mathcal{A},\lambda}(0), \mathsf{EvPKE}_{\mathcal{A},\lambda}(1)\right) = \text{negl}(\lambda),$$

where the experiment $\mathsf{EvPKE}_{\mathcal{A},\lambda}(b)$ *is defined as follows.*

- *Sample* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PVGen}(1^\lambda)$ *and* $(\mathsf{vk}, |\mathsf{ct}\rangle) \leftarrow \mathsf{PVEnc}(\mathsf{pk}, b)$.
- *Run* $\mathcal{A}_\lambda(\mathsf{pk}, \mathsf{vk}, |\mathsf{ct}\rangle)$*, and parse their output as a deletion certificate π and a state on register* A'.
- *If* $\mathsf{PVVrfy}(\mathsf{vk}, \pi) = \top$*, output* A'*, and otherwise output* $\bot$.

Construction via OWF. We now present our generic compiler that augments any (post-quantum secure) PKE scheme with the PVD property, assuming the existence of one-way functions.

Construction 4.3 (PKE with PVD from OWF). Let $\lambda \in \mathbb{N}$ and let $F : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ be a one-way function, and let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a standard (post-quantum) public-key encryption scheme. Consider the PKE scheme with PVD consisting of the following efficient algorithms:

- $\mathsf{PVGen}(1^\lambda)$: Same as $\mathsf{Gen}(1^\lambda)$.
- $\mathsf{PVEnc}(\mathsf{pk}, b)$: Sample $x_0, x_1 \leftarrow \{0,1\}^{n(\lambda)}$, define $y_0 = F(x_0), y_1 = F(x_1)$, and output

$$\mathsf{vk} := (y_0, y_1), \quad |\mathsf{ct}\rangle := \left(\mathsf{Enc}(\mathsf{pk}, x_0 \oplus x_1), \frac{1}{\sqrt{2}}\left(|x_0\rangle + (-1)^b |x_1\rangle\right)\right).$$

- $\mathsf{PVDec}(\mathsf{sk}, |\mathsf{ct}\rangle)$: Parse $|\mathsf{ct}\rangle$ as a classical ciphertext ct' and a quantum state $|\psi\rangle$. Compute $z \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}')$, measure $|\psi\rangle$ in the Hadamard basis to obtain $w \in \{0,1\}^{n(\lambda)}$, and output the bit $b = z \cdot w$.
- $\mathsf{PVDel}(|\mathsf{ct}\rangle)$: Parse $|\mathsf{ct}\rangle$ as a classical ciphertext ct' and a quantum state $|\psi\rangle$. Measure $|\psi\rangle$ in the computational basis to obtain $x' \in \{0,1\}^{n(\lambda)}$, and output $\pi := x'$.
- $\mathsf{PVVrfy}(\mathsf{vk}, \pi)$: Parse vk as (y_0, y_1) and output $\top$ if and only if $F(\pi) \in \{y_0, y_1\}$.

Theorem 4.4. *If one-way functions exist, then Construction 4.3 instantiated with any (post-quantum) public-key encryption scheme satisfies correctness of deletion (according to Definition 4.1) as well as (everlasting) certified deletion security according to Definition 4.2.*

Proof. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a standard (post-quantum) public-key encryption scheme. Then, correctness of deletion follows from the fact that measuring $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle)$ in the Hadamard basis produces a vector orthogonal to $x_0 \oplus x_1$, whereas measuring the state $\frac{1}{\sqrt{2}}(|x_0\rangle - |x_1\rangle)$ in the Hadamard basis produces a vector that is not orthogonal to $x_0 \oplus x_1$. Certified deletion security follows from Theorem 3.1, by setting $\mathcal{Z}_\lambda(x_0 \oplus x_1) = \mathsf{Enc}(\mathsf{pk}, x_0 \oplus x_1)$ and invoking the semantic security of the public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. □

Remark 4.5. *Following [BK22], we can plug various primitives into the above compiler to obtain X with* PVD *for $X \in \{$commitment, attribute-based encryption, fully-homomormphic encryption, witness encryption, timed-release encryption$\}$. To obtain the result for attribute-based encryption, we need to rely on Theorem 3.1 instantiated with an interactive distribution $\mathcal{Z}_\lambda$, following [BKP23].*

5 Publicly-Verifiable Deletion from One-Way State Generators

In this section, we show how to base the assumptions behind our generic compiler for PVD to something potentially even weaker than one-way functions, namely the existence of so-called one-way state generators (if we allow for quantum verification keys). Morimae and Yamakawa [MY22] introduced one-way state generator (OWSG) as a quantum analogue of a one-way function.

Definition 5.1 (One-Way State Generator)**.** *Let $n \in \mathbb{N}$ be the security parameter. A one-way state generator* (OWSG) *is a tuple* $(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Ver})$ *consisting of QPT algorithms:*

$\mathsf{KeyGen}(1^n) \to k$: *given as input 1^n, it outputs a uniformly random key $k \leftarrow \{0,1\}^n$.*

$\mathsf{StateGen}(k) \to \phi_k$: *given as input a key $k \in \{0,1\}^n$, it outputs an m-qubit quantum state ϕ_k.*

$\mathsf{Ver}(k', \phi_k) \to \top/\bot$: *given as input a supposed key k' and state ϕ_k, it outputs $\top$ or $\bot$.*

We require that the following property holds:

Correctness: *For any $n \in \mathbb{N}$, the scheme* (KeyGen, StateGen, Ver) *satisfies*

$$\Pr[\top \leftarrow \mathsf{Ver}(k, \phi_k) \,:\, k \leftarrow \mathsf{KeyGen}(1^n),\, \phi_k \leftarrow \mathsf{StateGen}(k)] \geq 1 - \mathrm{negl}(n).$$

Security: For any computationally bounded quantum algorithm $\mathcal{A}$ and any $t = \mathrm{poly}(\lambda)$:

$$\Pr[\top \leftarrow \mathsf{Ver}(k', \phi_k) \,:\, k \leftarrow \mathsf{KeyGen}(1^n),\, \phi_k \leftarrow \mathsf{StateGen}(k),\, k' \leftarrow \mathcal{A}(\phi_k^{\otimes t})] \leq \mathrm{negl}(n).$$

Morimae and Yamakawa [MY22] showed that if pseudorandom quantum state generators with $m \geq c \cdot n$ for some constant $c > 1$ exist, then so do one-way state generators. Informally, a pseudorandom state generator [JLS18, MY22] is a QPT algorithm that, on input $k \in \{0,1\}^n$, outputs an m-qubit state $|\phi_k\rangle$ such that $|\phi_k\rangle^{\otimes t}$ over uniformly random k is computationally indistinguishable from a Haar random states of the same number of copies, for any polynomial $t(n)$.

Certified-everlasting theorem for OWSG. To prove that our generic compiler yields PVD even when instantiated with a OWSG, it suffices to extend Theorem 3.1 as follows.

Theorem 5.2. *Let* (KeyGen, StateGen, Ver) *be a OSWG. Let $\{\mathcal{Z}_\lambda(\cdot,\cdot,\cdot,\cdot)\}_{\lambda\in\mathbb{N}}$ be a quantum operation with four arguments: a $n(\lambda)$-bit string z, two $m(\lambda)$-qubit quantum states (ψ_0, ψ_1), and a $n(\lambda)$-qubit quantum register* A. *Let $\mathscr{A}$ be a class of adversaries*[4] *such that for all $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathscr{A}$, and for any strings $z \in \{0,1\}^{n(\lambda)}$, states (ψ_0, ψ_1), and any state $|\psi\rangle^{\mathsf{A},\mathsf{C}}$ on $n(\lambda)$-qubit register* A *and arbitrary size register* C,

$$\left|\Pr[\mathcal{A}_\lambda\left(\mathcal{Z}_\lambda\left(z,\psi_0,\psi_1,\mathsf{A}\right),\mathsf{C}\right)=1] - \Pr[\mathcal{A}_\lambda\left(\mathcal{Z}_\lambda\left(0^{n(\lambda)},\psi_0,\psi_1,\mathsf{A}\right),\mathsf{C}\right)=1]\right| = \mathrm{negl}(\lambda).$$

That is, $\mathcal{Z}_\lambda$ is semantically-secure against $\mathcal{A}_\lambda$ with respect to its first input. 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 $x_0, x_1 \leftarrow \{0,1\}^{n(\lambda)}$, generate quantum states ψ_{x_0} and ψ_{x_1} by running the procedure StateGen on input x_0 and x_1, respectfully, and initialize $\mathcal{A}_\lambda$ with*

$$\mathcal{Z}_\lambda\left(x_0 \oplus x_1, \psi_{x_0}, \psi_{x_1}, \frac{1}{\sqrt{2}}\left(|x_0\rangle + (-1)^b |x_1\rangle\right)\right).$$

- *$\mathcal{A}_\lambda$'s output is parsed as a string $x' \in \{0,1\}^{n(\lambda)}$ and a residual state on register* A′.
- *If $\mathsf{Ver}(x', \psi_{x_i})$ outputs $\top$ for some $i \in \{0,1\}$, then output* A′, *and otherwise output $\bot$.*

Then,

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

Proof. The proof is analogous to Theorem 3.1, except that we invoke the security of the OWSG, rather than the one-wayness of the underlying one-way function. □

[4] 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 $\mathscr{A}$.

Construction from OWSG. We now consider the following PKE scheme with PVD. The construction is virtually identical to Construction 4.3, except that we replace one-way functions with one-way state generators. This means that the verification key is now quantum.

Construction 5.3 (PKE with PVD from OWSG)*.* Let $\lambda \in \mathbb{N}$ and let $(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Ver})$ be a OSWG, and let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a standard (post-quantum) public-key encryption scheme. Consider the following PKE scheme with PVD:

- $\mathsf{PVGen}(1^\lambda)$: Same as $\mathsf{Gen}(1^\lambda)$.
- $\mathsf{PVEnc}(\mathsf{pk}, b)$: Sample $x_0, x_1 \leftarrow \{0,1\}^{n(\lambda)}$ and generate quantum states ψ_{x_0} and ψ_{x_1} by running the procedure StateGen on input x_0 and x_1, respectfully. Then, output

$$\mathsf{vk} := (\psi_{x_0}, \psi_{x_1}), \quad |\mathsf{ct}\rangle := \left(\mathsf{Enc}(\mathsf{pk}, x_0 \oplus x_1), \frac{1}{\sqrt{2}}\left(|x_0\rangle + (-1)^b |x_1\rangle\right)\right).$$

- $\mathsf{PVDec}(\mathsf{sk}, |\mathsf{ct}\rangle)$: Parse $|\mathsf{ct}\rangle$ as a classical ciphertext ct' and a quantum state $|\psi\rangle$. Compute $z \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$, measure $|\psi\rangle$ in the Hadamard basis to obtain $w \in \{0,1\}^{n(\lambda)}$, and output the bit $b = z \cdot w$.
- $\mathsf{PVDel}(|\mathsf{ct}\rangle)$: Parse $|\mathsf{ct}\rangle$ as a classical ciphertext ct' and a quantum state $|\psi\rangle$. Measure $|\psi\rangle$ in the computational basis to obtain $x' \in \{0,1\}^{n(\lambda)}$, and output $\pi := x'$.
- $\mathsf{PVVrfy}(\mathsf{vk}, \pi)$: Parse vk as (ψ_{x_0}, ψ_{x_1}) and output $\top$ if and only if $\mathsf{Ver}(\pi, \psi_{x_i})$ outputs $\top$, for some $i \in \{0,1\}$. Otherwise, output $\bot$.

Remark 5.4. *Unlike in Construction 4.3, the verification key* vk *in Construction 5.3 is quantum. Hence, the procedure* $\mathsf{PVVrfy}(\mathsf{vk}, \pi)$ *in Construction 5.3 may potentially consume the public verification key* (ψ_{x_0}, ψ_{x_1}) *when verifying a dishonest deletion certificate* π*. However, by the security of the OWSG scheme, we can simply hand out* $(\psi_{x_0}^{\otimes t}, \psi_{x_1}^{\otimes t})$ *for any number of* $t = \mathrm{poly}(\lambda)$ *many copies without compromising security. This would allow multiple users to verify whether a (potentially dishonest) deletion certificate is valid. We focus on the case* $t = 1$ *for simplicity.*

Theorem 5.5. *If one-way state generators exist, then Construction 5.3 instantiated with any (post-quantum) public-key encryption scheme satisfies correctness of deletion (according to Definition 4.1) as well as (everlasting) certified deletion security according to Definition 4.2.*

Proof. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a standard (post-quantum) public-key encryption scheme. Then, correctness of deletion follows from the fact that measuring $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle)$ in the Hadamard basis produces a vector orthogonal to $x_0 \oplus x_1$, whereas measuring the state $\frac{1}{\sqrt{2}}(|x_0\rangle - |x_1\rangle)$ in the Hadamard basis produces a vector that is not orthogonal to $x_0 \oplus x_1$. Certified deletion security follows from Theorem 3.1, by setting $\mathcal{Z}_\lambda(x_0 \oplus x_1) = \mathsf{Enc}(\mathsf{pk}, x_0 \oplus x_1)$ and invoking the semantic security of the public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. □

Following [BK22] and using the fact that pseudorandom state generators imply one-way state generators [MY22], we immediately obtain:

Theorem 5.6. *If pseudorandom quantum state generators with* $m \geq c \cdot n$ *for some constant* $c > 1$ *exist, then there exists a generic compiler that that adds PVD to any (post-quantum) public-key encryption scheme. Moreover, we can plug various primitives into the the compiler to obtain* X *with* PVD *for* $X \in \{$*commitment, attribute-based encryption, fully-homomormphic encryption, witness encryption, timed-release encryption*$\}$.

References

[AKN+23] Shweta Agarwal, Fuyuki Kitagawa, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Public key encryption with secure key leasing. In *Eurocrypt 2023 (to appear)*, 2023.

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

[BGG+23] James Bartusek, Sanjam Garg, Vipul Goyal, Dakshita Khurana, Giulio Malavolta, Justin Raizes, and Bhaskar Roberts. Obfuscation and outsourced computation with certified deletion. Cryptology ePrint Archive, Paper 2023/265, 2023.

[BI20] Anne Broadbent and Rabib Islam. Quantum encryption with certified deletion. *Lecture Notes in Computer Science*, page 92–122, 2020.

[BK22] James Bartusek and Dakshita Khurana. Cryptography with certified deletion. Cryptology ePrint Archive, Paper 2022/1178, 2022. https://eprint.iacr.org/2022/1178.

[BKP23] James Bartusek, Dakshita Khurana, and Alexander Poremba. Publicly-verifiable deletion via target-collapsing functions, 2023.

[DS22] Marcel Dall'Agnol and Nicholas Spooner. On the necessity of collapsing. Cryptology ePrint Archive, Paper 2022/786, 2022. https://eprint.iacr.org/2022/786.

[HHK+09] Iftach Haitner, Omer Horvitz, Jonathan Katz, Chiu-Yuen Koo, Ruggero Morselli, and Ronen Shaltiel. Reducing complexity assumptions for statistically-hiding commitment. *Journal of Cryptology*, 22(3):283–310, 2009.

[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 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 606–636. Springer, 2021.

[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. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part I*, volume 13507 of *Lecture Notes in Computer Science*, pages 239–268. Springer, 2022.

[JLS18] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. Cryptology ePrint Archive, Paper 2018/544, 2018. https://eprint.iacr.org/2018/544.

[Kre21] William Kretschmer. Quantum pseudorandomness and classical complexity. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[MW23] Giulio Malavolta and Michael Walter. Non-interactive quantum key distribution. Cryptology ePrint Archive, Paper 2023/500, 2023. https://eprint.iacr.org/2023/500.

[MY22] Tomoyuki Morimae and Takashi Yamakawa. Quantum commitments and signatures without one-way functions. LNCS, pages 269–295. Springer, Heidelberg, 2022.

[Por23] Alexander Poremba. Quantum proofs of deletion for learning with errors. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 90:1–90:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[Unr15] Dominique Unruh. Revocable quantum timed-release encryption. *J. ACM*, 62(6), dec 2015.