Batched Identity-Based Encryption with Succinct Publicly Verifiable Subset Deletion

Shayeef Murshid[1][0009-0008-7042-7911], Ramprasad Sarkar[2][0000-0002-9907-3319], and Mriganka Mandal[1][0000-0001-8691-4185]

[1] Cryptology and Security Research Unit
R C Bose Centre for Cryptology and Security
Indian Statistical Institute Kolkata, Kolkata, India-700108
{shayeef_r,mriganka}@isical.ac.in
[2] Department of Mathematics, Indian Institute of Information Technology Lucknow, Lucknow 226002, Uttar Pradesh, India
rpsarkar@iiitl.ac.in

Abstract. Batched identity-based encryption (BIBE) compresses a bounded set of identities into a succinct public digest, and provides a single secret key for decrypting ciphertexts associated with identities in the set. We introduce batched identity-based encryption with succinct publicly verifiable subset deletion (BIBE-SPVD), which allows a holder of polynomially many quantum ciphertexts to certify deletion of any chosen subset using a single succinct classical certificate while preserving the usability of the remaining ciphertexts. We first construct a classical BIBE from identity-based encryption with puncturable master secret key, somewhere statistically binding (SSB) hashing, and indistinguishability obfuscation. We then equip this scheme with publicly verifiable certified-everlasting deletion, using a generic compiler specialized to the BIBE functionality. Finally, individual deletion certificates are aggregated using a somewhere-extractable batch argument. For each ciphertext position, the corresponding statement requires a valid deletion certificate precisely when that position belongs to the deleted subset. The identity digest and batch key remain succinct, and for N ciphertexts the aggregate deletion certificate has size $\mathsf{poly}(\lambda, \log N)$, independent of the number deleted.

Keywords: Batched Identity-Based Encryption · Publicly Verifiable Deletion · Certified Deletion · Subset Deletion · Somewhere Statistically Binding Hashing · Batch Arguments

1 Introduction

Identity-based encryption (IBE) [7,21] eliminates the need for conventional public-key certificates by allowing arbitrary strings, such as email addresses, account identifiers, or device identifiers, to serve directly as public keys. A trusted authority holding a master secret key generates the corresponding identity secret

keys. However, when a recipient is authorized to decrypt ciphertexts associated with many identities, standard IBE requires one secret key for each identity. Consequently, the key-management overhead grows linearly with the number of authorized identities.

Batched identity-based encryption (BIBE), introduced by Agarwal *et al.* [1], addresses this problem by allowing a bounded collection of identities to be publicly compressed into a succinct digest. From the digest and a batch label, the authority computes a single decryption key that enables decryption for every identity represented by the digest. Thus, the authority-side key-issuance cost and the resulting decryption key need not grow linearly with the number of identities contained in the batch. Agarwal *et al.* also study thresholdizable batching and applications including encrypted mempools. A post-quantum lattice-based BIBE construction has subsequently been developed by Xiong *et al.* [23].

A different problem arises when encrypted data is no longer intended to remain recoverable. Removing a classical ciphertext does not guarantee that another copy has not been retained. Certified deletion, introduced by Broadbent and Islam [10], exploits quantum information to provide such a guarantee. A holder of a quantum ciphertext performs a destructive deletion procedure and produces a classical certificate. Once this certificate is accepted, the protected plaintext remains hidden even if secret key material is subsequently exposed. The notion has been developed for public-key encryption, attribute-based encryption, and other expressive cryptographic functionalities [3, 17]. Generic certified-deletion techniques have also been studied in the context of obfuscation, functional encryption, and outsourced computation [2]. More recent works give generic compilers for publicly verifiable deletion, where the deletion certificate can be checked using public verification information [4, 5, 18].

The combination of batching and deletion raises a natural multi-ciphertext problem. Consider a collection of encrypted transactions, records, or messages. At a later stage, the holder may wish to delete only some of them while keeping the rest intact and usable. The holder should also be able to provide convincing evidence that the selected items have been permanently deleted. A natural approach is to generate a separate deletion certificate for each deleted item, but this makes the total proof size grow with the number of deletions.

Can we simultaneously obtain succinct batched authorization and a single succinct publicly verifiable certificate for deletion of an arbitrary subset of ciphertexts?

There are two distinct succinctness problems. The first concerns authorization. A potentially large identity set should be represented by a succinct digest and one succinct batch key. The second concerns deletion. Potentially many individual deletion certificates should be represented by one succinct publicly verifiable object. Somewhere statistically binding hashing is naturally suited to the first task because it provides a succinct commitment together with succinct local openings. For the second task, hashing alone is insufficient. A short hash of the individual deletion certificates does not prove that the committed values are

valid certificates. We therefore use a somewhere-extractable batch argument for the public deletion-verification relation, which proves simultaneously that every position of the collection admits a valid witness and, in trapdoor mode, allows a genuine individual certificate to be extracted at one hidden position.
A further difficulty concerns where certified deletion is inserted into the construction. A direct approach would first build a puncturable IBE with certified deletion and then use it inside the batching mechanism. In a punctured-programming construction, however, the master secret key may determine the randomness of identity-specific public-key generation. If the full master secret key is revealed after deletion, then the challenge identity's key-generation randomness becomes recoverable. This conflicts with the standard puncturable pseudorandom-function hybrid. We avoid this issue entirely. Our construction first produces an ordinary classical BIBE and proves its security without any post-challenge master-key exposure. Only after this proof is complete do we apply a generic publicly verifiable deletion compiler to the finished BIBE functionality.
Our Contributions. We introduce batched identity-based encryption with succinct publicly verifiable subset deletion (BIBE-SPVD). Our contributions are summarized below.

- **A generic classical BIBE from IBE with puncturable master secret key.** We construct a classical BIBE from an identity-based encryption scheme with puncturable master secret key, an SSB hash function, and indistinguishability obfuscation. The digest is an SSB hash of a canonical padded representation of the identity batch. The batch key is an obfuscated circuit that verifies a local SSB opening before releasing the corresponding identity secret key. The security proof proceeds by progressively rejecting the challenge identity at every possible position of the committed batch and then replacing the full master secret key by one punctured at the challenge identity.
- **Publicly verifiable certified deletion for BIBE.** We give a direct instantiation of a publicly verifiable BIBE-CD scheme. The construction specializes the publicly verifiable certified-deletion transformation of Kitagawa *et al.* [18] to the BIBE functionality while preserving the native Digest and ComputeKey algorithms. The resulting scheme satisfies publicly verifiable certified-everlasting deletion and, in particular, remains computationally secure even when the complete master secret key is revealed after successful verification of a deletion certificate. This single-ciphertext guarantee is everlasting: the complete experiment outputs, including the rejection outcome, are statistically close for the two challenge messages. By contrast, the subset-deletion guarantee that we obtain from it is computational.
- **Succinct publicly verifiable subset deletion.** We extend single-ciphertext certified deletion to the deletion of an arbitrary subset D of a polynomial-size ciphertext collection. For every position $i \in [N]$, we define a public statement $x_i = (d_i, \mathsf{vk}_i)$, where $d_i = 1$ if and only if $i \in D$. For deleted positions, a valid witness is an individual deletion certificate accepted under vk_i, while

undeleted positions admit a fixed dummy witness. A somewhere-extractable batch argument proves simultaneously that all N statements admit valid witnesses. The resulting subset-deletion certificate consists of a single succinct batch proof. The subset-deletion algorithm also returns the unmeasured ciphertexts at positions outside D; these remain with the holder and are not part of the public deletion certificate.

- **Adaptive choice of the deleted subset.** The challenge identity-label pairs are fixed selectively, while the challenge messages and the deleted subset may be chosen later. If $\Delta = \{i \in [N] : m_{0,i} \neq m_{1,i}\}$, security requires $\Delta \subseteq D$. Thus every ciphertext carrying challenge-dependent information must be deleted, while ciphertexts outside D may remain usable because their two challenge messages are identical.
- **Security via somewhere extraction.** The security proof switches the challenge ciphertexts one position at a time. At every position $j \in \Delta$, the subset condition guarantees $j \in D$. We program the somewhere-extractable batch argument at position j, and from any accepting aggregate proof extract a valid individual deletion certificate for the j-th challenge ciphertext. This reduces the corresponding message-switch transition to the single-ciphertext publicly verifiable certified-deletion security of the $\mathsf{BIBE\text{-}CD}$ construction from Appendix A. The proof also explicitly handles pre-challenge batch-key queries, queries that authorize a challenge attribute other than the one currently being switched, subset inadmissibility, and aggregate-verification failure. Executions in which the aggregate proof is rejected are handled separately: no master secret key is released on them, and the adversary's final guess is bounded using the selective semantic security of the $\mathsf{BIBE\text{-}CD}$ scheme. Public verifiability allows this reduction to recognize rejection by itself. Certified-everlasting deletion is invoked only for accepted aggregate proofs, after extracting an individual certificate. This separation is necessary because the single-ciphertext certified-deletion experiment outputs its own rejection symbol rather than the adversary's final bit.
- **Succinct certificate size.** The identity digest and batch secret key retain the succinctness of the underlying classical BIBE. For a collection of N ciphertexts, the aggregate subset-deletion certificate has size $\mathsf{poly}(\lambda, \log N)$ and is independent of the number $|D|$ of deleted ciphertexts. The verifier still receives the full public verification-key collection and the subset D, and therefore our succinctness claim concerns the certificate size rather than the total verifier input or verification time.

Related Work. Batched identity-based encryption was introduced by Agarwal *et al.* [1], who also investigated thresholdizable batching. Subsequent work has explored post-quantum and epochless variants. Boneh *et al.* [8] showed that any $\mathsf{KP\text{-}ABE}$ scheme can be used to construct an *epochless* BIBE, while Xiong *et al.* [23] proposed a lattice-based post-quantum BIBE. In a separate line of work, certified deletion was introduced by Broadbent and Islam [10]. Hiroka *et al.* [17] extended the notion to public-key and attribute-based settings, and Bartusek

and Khurana [3] developed generic compilers for certified-everlasting deletion. Bartusek *et al.* further studied generic certified-deletion techniques for software, obfuscation, functional encryption, and related settings [2]. Publicly verifiable deletion was subsequently achieved through generic compilers based on target-collapsing functions and progressively weaker assumptions [4, 5, 18].

2 Preliminaries

This section recalls the cryptographic tools used in the construction.

2.1 Batched Identity-Based Encryption

Let $\mathcal{I}$ be an identity space, $\mathcal{M}$ be a plaintext space, and Γ be a batch-label space. A batched identity-based encryption scheme with maximum batch size B is a tuple $\prod_{\mathsf{BIBE}} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Digest}, \mathsf{ComputeKey}, \mathsf{Dec})$ with the following algorithms.

- $\mathsf{Setup}(1^\lambda, 1^B) \to (\mathsf{mpk}, \mathsf{msk})$: On input the security parameter and an upper bound B on the batch size, outputs a master public key mpk and a master secret key msk.
- $\mathsf{Enc}(\mathsf{mpk}, m, \mathsf{id}, \mathsf{tg}) \to \mathsf{ct}$: On input the master public key, a message $m \in \mathcal{M}$, an identity $\mathsf{id} \in \mathcal{I}$, and a batch label $\mathsf{tg} \in \Gamma$, outputs a ciphertext ct.
- $\mathsf{Digest}(\mathsf{mpk}, S) \to \mathsf{dig}$: On input the master public key and a set $S \subseteq \mathcal{I}$ with $|S| \leq B$, outputs a public batch digest dig representing S.
- $\mathsf{ComputeKey}(\mathsf{msk}, \mathsf{dig}, \mathsf{tg}) \to \mathsf{sk}$: On input the master secret key, a batch digest dig, and a batch label tg, outputs a batch secret key sk.
- $\mathsf{Dec}(\mathsf{sk}, S, (\mathsf{id}, \mathsf{tg}), \mathsf{ct}) \to m/\perp$: On input a batch secret key, the corresponding identity set S, an identity-label pair $(\mathsf{id}, \mathsf{tg})$, and a ciphertext ct, outputs a message $m \in \mathcal{M}$ or $\perp$.

We require the following correctness and succinctness properties.

- **Decryption correctness.** For every security parameter $\lambda \in \mathbb{N}$, every batch bound $B \in \mathbb{N}$, every set $S \subseteq \mathcal{I}$ with $|S| \leq B$, every identity $\mathsf{id} \in S$, every $\mathsf{tg} \in \Gamma$, and every message $m \in \mathcal{M}$,

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, S, (\mathsf{id}, \mathsf{tg}), \mathsf{ct}) = m \;\middle|\; \begin{array}{l} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^B), \\ \mathsf{dig} \leftarrow \mathsf{Digest}(\mathsf{mpk}, S), \\ \mathsf{sk} \leftarrow \mathsf{ComputeKey}(\mathsf{msk}, \mathsf{dig}, \mathsf{tg}), \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, m, \mathsf{id}, \mathsf{tg}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

- **Batch succinctness.** The size of the digest dig, the size of the batch secret key sk, and the running time of $\mathsf{ComputeKey}$ are bounded by $\mathsf{poly}(\lambda, k, \log B)$, where k is the identity length. In particular, these quantities do not grow linearly with the number of identities represented by the digest.

- **Security**. For a bit $b \in \{0,1\}$, consider the following experiment between a challenger and a QPT adversary $\mathcal{A}$.
  - ◇ **Target Identity and Label:** The adversary outputs a challenge identity $\mathsf{id}^* \in \mathcal{I}$ and a challenge batch label $\mathsf{tg}^* \in \Gamma$ before setup.
  - ◇ **Setup:** The challenger computes $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^B)$ and sends mpk to $\mathcal{A}$.
  - ◇ **Query Phase 1:** The adversary adaptively submits batch-key queries (S, tg) with $S \subseteq \mathcal{I}$ and $|S| \leq B$. For each query, the challenger computes $\mathsf{dig} \leftarrow \mathsf{Digest}(\mathsf{mpk}, S)$ and $\mathsf{sk} \leftarrow \mathsf{ComputeKey}(\mathsf{msk}, \mathsf{dig}, \mathsf{tg})$, and sends sk to the adversary.
  - ◇ **Challenge Messages:** The adversary outputs two equal-length messages $(m_0, m_1) \in \mathcal{M}^2$.
  - ◇ **Challenge Ciphertext:** The challenger samples $b \leftarrow \{0,1\}$ and computes $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, m_b, \mathsf{id}^*, \mathsf{tg}^*)$. It sends ct^* to the adversary.
  - ◇ **Query Phase 2:** The adversary continues to issue adaptive batch-key queries (S, tg), which are answered as in the first query phase.
  - ◇ **Global Admissibility:** Every batch-key query (S, tg) made in either query phase must satisfy $\mathsf{tg} \neq \mathsf{tg}^*$ or $\mathsf{id}^* \notin S$. Thus, the adversary never receives a batch secret key capable of decrypting the challenge ciphertext.
  - ◇ **Guess:** The adversary outputs a bit $b' \in \{0,1\}$.

  We say that $\prod_{\mathsf{BIBE}}$ satisfies *post-quantum selective security* if for every polynomially bounded $B = B(\lambda)$ and every QPT adversary $\mathcal{A}$ satisfying the global admissibility condition,

$$|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]| \leq \mathsf{negl}(\lambda).$$

2.2 Indistinguishability Obfuscator

Let $\{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of circuits. A PPT algorithm $i\mathcal{O}$ is an indistinguishability obfuscator for this circuit class if it satisfies the following properties.

- **Correctness.** For every $\lambda \in \mathbb{N}$, every $C \in \mathcal{C}_\lambda$, and every input x, $\Pr[C'(x) = C(x) \mid C' \leftarrow i\mathcal{O}(C)] = 1$.
- **Indistinguishability.** For every QPT distinguisher $\mathcal{D}$ and every pair of functionally equivalent circuits $C_0, C_1 \in \mathcal{C}_\lambda$ such that $C_0(x) = C_1(x)$ for every x and $|C_0| = |C_1|$, $|\Pr[\mathcal{D}(i\mathcal{O}(C_0)) = 1] - \Pr[\mathcal{D}(i\mathcal{O}(C_1)) = 1]| \leq \mathsf{negl}(\lambda)$.

Post-quantum candidates for indistinguishability obfuscation have been proposed in [9, 14, 16, 22].

2.3 Identity-Based Encryption with Puncturable Master Secret Key

Let $\mathcal{X}$ be an identity space and $\mathcal{M}$ be a plaintext space. We use identity-based encryption with puncturable master secret key as defined by Çakan *et al.* [11]. Such a scheme is a tuple $\prod_{\mathsf{PIBE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Punc}, \mathsf{Enc}, \mathsf{Dec})$ with the following algorithms.

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}_0, \mathsf{msk}_0)$: On input the security parameter, outputs a master public key mpk_0 and a master secret key msk_0.
- $\mathsf{KeyGen}(\mathsf{msk}_0, X) \to \mathsf{sk}_X$: On input the master secret key and an identity $X \in \mathcal{X}$, deterministically outputs the corresponding identity secret key sk_X. Determinism is needed because KeyGen is evaluated inside the obfuscated circuits of Section 4; any randomness it uses is derived from its input key, as in the construction recalled below.
- $\mathsf{Punc}(\mathsf{msk}_0, X^*) \to \mathsf{msk}_{X^*}$: On input the master secret key and a target identity $X^* \in \mathcal{X}$, outputs a master secret key msk_{X^*} punctured at X^*. The punctured master secret key can still be used with KeyGen to generate identity secret keys for every $X \neq X^*$.
- $\mathsf{Enc}(\mathsf{mpk}_0, X, m) \to \mathsf{CT}$: On input the master public key, an identity $X \in \mathcal{X}$, and a message $m \in \mathcal{M}$, outputs a ciphertext CT.
- $\mathsf{Dec}(\mathsf{sk}_X, \mathsf{CT}) \to m/\perp$: On input an identity secret key and a ciphertext, outputs a message $m \in \mathcal{M}$ or $\perp$.

We require the following properties.

- **Decryption correctness.** For every $\lambda \in \mathbb{N}$, every $X \in \mathcal{X}$, and every $m \in \mathcal{M}$,

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_X, \mathsf{CT}) = m \;\middle|\; \begin{array}{l}(\mathsf{mpk}_0, \mathsf{msk}_0) \leftarrow \mathsf{Setup}(1^\lambda),\\ \mathsf{sk}_X \leftarrow \mathsf{KeyGen}(\mathsf{msk}_0, X),\\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{mpk}_0, X, m)\end{array}\right] = 1.$$

- **Punctured-key correctness.** For every pair of identities $X, X^* \in \mathcal{X}$ such that $X \neq X^*$, if $(\mathsf{mpk}_0, \mathsf{msk}_0) \leftarrow \mathsf{Setup}(1^\lambda)$ and $\mathsf{msk}_{X^*} \leftarrow \mathsf{Punc}(\mathsf{msk}_0, X^*)$, then an identity secret key generated using msk_{X^*} correctly decrypts ciphertexts generated for X.
- **Strong punctured-key correctness.** For every $X, X^* \in \mathcal{X}$ with $X \neq X^*$, we require the stronger pointwise property $\mathsf{KeyGen}(\mathsf{msk}_{X^*}, X) = \mathsf{KeyGen}(\mathsf{msk}_0, X)$ with probability 1 over $(\mathsf{mpk}_0, \mathsf{msk}_0) \leftarrow \mathsf{Setup}(1^\lambda)$ and $\mathsf{msk}_{X^*} \leftarrow \mathsf{Punc}(\mathsf{msk}_0, X^*)$.
- **Security.** For a bit $b \in \{0, 1\}$, consider the following experiment between a challenger and a QPT adversary $\mathcal{A}$.
  - ◇ **Target Identity:** The adversary outputs a challenge identity $X^* \in \mathcal{X}$ before setup.

- ◇ **Setup and Puncturing:** The challenger computes $(\mathsf{mpk}_0, \mathsf{msk}_0) \leftarrow \mathsf{Setup}(1^\lambda)$ and $\mathsf{msk}_{X^*} \leftarrow \mathsf{Punc}(\mathsf{msk}_0, X^*)$, and sends $(\mathsf{mpk}_0, \mathsf{msk}_{X^*})$ to $\mathcal{A}$.
- ◇ **Challenge Messages:** The adversary outputs two equal-length messages $(m_0, m_1) \in \mathcal{M}^2$.
- ◇ **Challenge Ciphertext:** The challenger samples $b \leftarrow \{0,1\}$ and computes $\mathsf{CT}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}_0, X^*, m_b)$. It sends CT^* to the adversary.
- ◇ **Guess:** The adversary outputs a bit $b' \in \{0,1\}$.

We say that $\prod_{\mathsf{PIBE}}$ satisfies *post-quantum puncturable-master-secret-key security* if for every QPT adversary $\mathcal{A}$, $|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]| \leq \mathsf{negl}(\lambda)$.

Çakan *et al.* [11] construct such a scheme from indistinguishability obfuscation, a puncturable pseudorandom function, and an underlying public-key encryption scheme. In their construction, the master secret key is a puncturable pseudorandom-function key K, and puncturing the master secret key at X^* consists of computing the corresponding punctured key K_{X^*}. For every $X \neq X^*$, puncturable-pseudorandom-function correctness gives $F(K_{X^*}, X) = F(K, X)$ with probability 1. Since these values are used as the explicit randomness of the underlying public-key key-generation algorithm, the identity secret keys generated from K and K_{X^*} are identical. Consequently, their concrete construction satisfies the pointwise strong punctured-key correctness required in our $i\mathcal{O}$ hybrid. In particular, KeyGen is deterministic in this construction. Ordinary punctured-key correctness, which only guarantees that keys derived from msk_{X^*} decrypt correctly, would not suffice, since the $i\mathcal{O}$ step in Lemma 2 requires functionally identical circuits.

2.4 Somewhere Statistically Binding Hash Functions

Let $\ell_{\mathsf{blk}} = \ell_{\mathsf{blk}}(\lambda)$ denote the block length and let $\Sigma = \{0,1\}^{\ell_{\mathsf{blk}}}$. A somewhere statistically binding hash function is a tuple $\prod_{\mathsf{SSB}} = (\mathsf{Setup}, \mathsf{Hash}, \mathsf{Open}, \mathsf{Vrfy})$.

- $\mathsf{Setup}(1^\lambda, 1^{\ell_{\mathsf{blk}}}, N, i^*) \to \mathsf{hk}$: On input the security parameter, block length, number of blocks N, and index $i^* \in [N]$, outputs a hash key hk.
- $\mathsf{Hash}(\mathsf{hk}, \mathbf{x}) \to h$: On input $\mathbf{x} = (x_1, \ldots, x_N) \in \Sigma^N$, outputs a hash value h.
- $\mathsf{Open}(\mathsf{hk}, \mathbf{x}, i) \to \pi_i$: Outputs an opening for block x_i.
- $\mathsf{Vrfy}(\mathsf{hk}, h, i, x, \pi) \to 1/0$: Checks whether x is a valid opening of the i-th block of h.

• **Correctness.** For every $\mathbf{x} = (x_1, \ldots, x_N)$, every $i, i^* \in [N]$, and $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^{\ell_{\mathsf{blk}}}, N, i^*)$, if $h \leftarrow \mathsf{Hash}(\mathsf{hk}, \mathbf{x})$ and $\pi_i \leftarrow \mathsf{Open}(\mathsf{hk}, \mathbf{x}, i)$, then, $\Pr[\mathsf{Vrfy}(\mathsf{hk}, h, i, x_i, \pi_i) = 1] = 1$.

- **Index hiding.** For every QPT adversary $\mathcal{A}$, every $N \leq 2^\lambda$, and every $i_0, i_1 \in [N]$, $|\Pr[\mathcal{A}(\mathsf{hk}_0) = 1] - \Pr[\mathcal{A}(\mathsf{hk}_1) = 1]| \leq \mathsf{negl}(\lambda)$, where $\mathsf{hk}_b \leftarrow \mathsf{Setup}(1^\lambda, 1^{\ell_{\mathsf{blk}}}, N, i_b)$.
- **Somewhere statistical binding.** A key hk is statistically binding at i^* if there do not exist h, distinct blocks $x \neq x'$, and openings π, π' such that $\mathsf{Vrfy}(\mathsf{hk}, h, i^*, x, \pi) = \mathsf{Vrfy}(\mathsf{hk}, h, i^*, x', \pi') = 1$. For every $i^* \in [N]$, $\Pr[\mathsf{hk} \text{ is statistically binding at } i^* \mid \mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^{\ell_{\mathsf{blk}}}, N, i^*)] \geq 1 - \mathsf{negl}(\lambda)$.
- **Succinctness.** The lengths satisfy $|\mathsf{hk}| = \mathsf{poly}(\lambda, \ell_{\mathsf{blk}}, \log N)$ and $|h|, |\pi| = \mathsf{poly}(\lambda, \ell_{\mathsf{blk}})$.

Post-quantum secure SSB constructions are known from standard post-quantum assumptions [19, 20].

2.5 Somewhere-Extractable Batch Arguments

Let $\mathcal{L}$ be an NP language with relation $\mathcal{R}$, computed by a circuit $\mathcal{C}_\mathcal{R}$. A publicly verifiable and non-interactive somewhere-extractable batch argument $\prod_{\mathsf{seBARG}}$ consists of algorithms $\prod_{\mathsf{seBARG}} = (\mathsf{Gen}, \mathsf{P}, \mathsf{V}, \mathsf{E})$ with the following syntax.

- $\mathsf{Gen}(1^\lambda, N, 1^n, 1^m, i^*) \to (\mathsf{crs}, \mathsf{td})$: On input the security parameter, the number N of instances, the instance length n, the witness length m, and an index $i^* \in [N]$, outputs a common reference string crs and an extraction trapdoor td. The number of instances is given in binary, since the reference string is required to be short in $\log N$.
- $\mathsf{P}(\mathsf{crs}, x_1, \ldots, x_N, w_1, \ldots, w_N) \to \pi$: On input N instances and corresponding witnesses satisfying $\mathcal{R}(x_i, w_i) = 1$ for every $i \in [N]$, outputs a batch argument π.
- $\mathsf{V}(\mathsf{crs}, x_1, \ldots, x_N, \pi) \to 0/1$: On input the common reference string, the complete batch of instances, and a proof π, outputs a verification bit.
- $\mathsf{E}(\mathsf{td}, (x_i)_{i \in [N]}, \pi) \to w^*$: On input the extraction trapdoor, the batch of instances, and a proof, outputs a candidate witness for the instance at the programmed position i^*.

We require the following properties.

- **Efficiency.** The size of the common reference string and proof is bounded by $\mathsf{poly}(\lambda, \log N, |\mathcal{C}_\mathcal{R}|)$, where $|\mathcal{C}_\mathcal{R}| = \mathsf{poly}(\lambda, n, m)$.
- **Completeness.** For every collection of instance-witness pairs $(x_i, w_i)_{i \in [N]}$ satisfying $\mathcal{R}(x_i, w_i) = 1$ for every $i \in [N]$, and every $i^* \in [N]$, $\Pr[\mathsf{V}(\mathsf{crs}, x_1, \ldots, x_N, \pi) = 1] = 1$, where $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, N, 1^n, 1^m, i^*)$ and $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x_1, \ldots, x_N, w_1, \ldots, w_N)$.
- **Index hiding.** For every QPT adversary $\mathcal{A}$ and every pair of indices $i_0, i_1 \in [N]$, $|\Pr[\mathcal{A}(\mathsf{crs}_0) = 1] - \Pr[\mathcal{A}(\mathsf{crs}_1) = 1]| \leq \mathsf{negl}(\lambda)$, where $(\mathsf{crs}_b, \mathsf{td}_b) \leftarrow \mathsf{Gen}(1^\lambda, N, 1^n, 1^m, i_b)$.

- **Somewhere argument of knowledge.** For every QPT adversary $\mathcal{A}$ and every $i^* \in [N]$, if $(x_1, \ldots, x_N, \pi) \leftarrow \mathcal{A}(\mathsf{crs})$ and $w^* \leftarrow \mathsf{E}(\mathsf{td}, (x_i)_{i\in[N]}, \pi)$, then $\Pr\left[\begin{matrix}\mathsf{V}(\mathsf{crs}, x_1, \ldots, x_N, \pi) = 1 \\ \wedge\ \mathcal{R}(x_{i^*}, w^*) = 0\end{matrix}\right] \leq \mathsf{negl}(\lambda)$, where $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, N, 1^n, 1^m, i^*)$. We stress that $\mathcal{A}$ here is an arbitrary QPT algorithm taking crs as its only input. In our security proof the statements and the proof are produced by the composition of the certified-deletion challenger, the reduction, and the subset-deletion adversary, which is again a single QPT algorithm of this form.
- **Straight-line extraction.** The extractor uses only the extraction trapdoor, the public batch of instances, and the proof. It does not rewind the adversary or access any residual quantum state.

Non-interactive batch arguments for NP whose common reference string can be generated in an extraction mode programmed at a hidden index were introduced by Choudhuri *et al.* [13], whose key tool is a somewhere-extractable commitment scheme with exactly this interface. Constructions secure under the LWE assumption, and hence plausibly post-quantum secure, are given by Choudhuri *et al.* [12] and by Devadas *et al.* [15], with proof length $\mathsf{poly}(\lambda, \log N, |\mathcal{C}_\mathcal{R}|)$. We require the instantiation used here to be secure against QPT adversaries. More precisely, index hiding and the somewhere argument of knowledge must hold against QPT adversaries, and extraction must be straight-line in the sense above. We do not claim that the security analyses of the cited works carry over to QPT adversaries. Theorem 5 and Corollary 1 are stated for an seBARG that satisfies all of the above properties against QPT adversaries.

3 Technical Overview

Notations. We begin by introducing the notation used throughout the paper. The symbol $\lambda \in \mathbb{N}$ denotes the security parameter. For a positive integer n, we write $[n] := \{1, \ldots, n\}$. The notation $x \leftarrow Y$ denotes that x is sampled uniformly at random from a finite set Y, while $x \leftarrow D$ denotes sampling x according to a distribution D. If A is an algorithm, which may be classical, probabilistic, or quantum, then $y \leftarrow A(x)$ denotes the output of A on input x, and $y \leftarrow A(x; r)$ denotes execution using explicit randomness r. The assignment $y := z$ indicates that y is set or defined to be equal to z. We use PPT and QPT to denote probabilistic polynomial-time and quantum polynomial-time algorithms, respectively. A function $\mu : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible if, for every constant $c > 0$, there exists $\lambda_0 \in \mathbb{N}$ such that $\mu(\lambda) < \lambda^{-c}$ for all $\lambda > \lambda_0$. We write $\mathsf{negl}(\lambda)$ for an unspecified negligible function and use $\approx_c$ to denote computational indistinguishability. For bit strings $x, \theta \in \{0,1\}^\ell$, we write $|x\rangle_\theta := \bigotimes_{j\in[\ell]} H^{\theta_j} |x_j\rangle$, where H denotes the Hadamard operator. The trace distance between quantum states ρ and σ is denoted by $\mathsf{TD}(\rho, \sigma)$ and is defined as $\mathsf{TD}(\rho, \sigma) = \frac{1}{2}\|\rho - \sigma\|_{\mathrm{tr}}$, where $\|A\|_{\mathrm{tr}} := \mathrm{Tr}\left(\sqrt{A^\dagger A}\right)$.

Our work is motivated by two complementary developments in batched encryption and certified deletion. Boneh *et al.* [8] showed that any KP-ABE scheme can be used to construct an *epochless* BIBE. Their construction provides a generic route to batched identity-based encryption, but does not provide a compact public digest algorithm as required by the standard BIBE syntax. More recently, Xiong *et al.* [23] proposed a lattice-based post-quantum BIBE with succinct batching properties. This naturally raises the question of whether such succinct BIBE functionality can instead be obtained through a generic compiler, without relying on a particular algebraic instantiation. A second motivation comes from certified deletion. Existing generic certified-deletion compilers of Bartusek and Khurana and subsequent works provide deletion guarantees for an individual ciphertext. For a collection of ciphertexts, independently applying such a mechanism to every ciphertext in a deleted subset produces one deletion certificate per ciphertext, causing the public deletion evidence to grow linearly with the size of the subset. We therefore consider a stronger functionality in which an arbitrary subset of ciphertexts can be deleted while the remaining ciphertexts stay usable, and deletion of the entire subset is certified by a single succinct publicly verifiable certificate. This leads naturally to batched identity-based encryption with succinct publicly verifiable subset deletion. Our construction proceeds in three stages discussed below.

3.1 Designing BIBE from IBE with Puncturable Master Secret Key

Let $\prod_{\mathsf{PIBE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Punc}, \mathsf{Enc}, \mathsf{Dec})$ be an identity-based encryption scheme with puncturable master secret key with identity space $\mathcal{X} = \mathcal{I} \times \Gamma$ (cf. Appendix 2.3). Given a batch $S \subseteq \mathcal{I}$ with $|S| \leq B$, we form a canonical padded representation $\mathbf{S} = \mathsf{Pad}_B(S)$. Unused positions are filled with a reserved symbol $\mathsf{id}_\perp \notin \mathcal{I}$. We compute the digest $h_S = \mathsf{SSB.Hash}(\mathsf{hk}_{\mathsf{id}}, \mathbf{S})$.

A batch secret key is an obfuscation of a membership-gated key-release circuit. The circuit hardwires the underlying master secret key, the SSB hash key, the digest h_S, and the batch label tg. On input (id, i, π), it verifies $\mathsf{SSB.Vrfy}(\mathsf{hk}_{\mathsf{id}}, h_S, i, \mathsf{id}, \pi)$. Only if this check succeeds does it compute the pointwise secret key for $X = (\mathsf{id}, \mathsf{tg})$. The security uses a batch-position hybrid. For a selective challenge pair $X^* = (\mathsf{id}^*, \mathsf{tg}^*)$, we progressively modify the batch-key circuit so that it rejects id^* at position 1, then at positions $1, 2$, and eventually at all B positions. At transition j, the global SSB key is changed to one statistically binding at position j. For every admissible key query on tg^*, the represented batch does not contain id^*. Thus the honest j-th block differs from id^*, and statistical binding makes the new rejection condition functionally redundant. Indistinguishability obfuscation hides the circuit change. Once the challenge identity is rejected everywhere, the circuit no longer requires the ability to generate its secret key. We replace the full underlying master secret key by msk_{X^*}. Strong punctured-key correctness preserves functionality on every other identity. Final transition follows from puncturable-master-secret-key security.

3.2 Publicly Verifiable Certified Deletion

We next equip the classical BIBE with publicly verifiable certified deletion. In Appendix A, we give a direct construction of a publicly verifiable $\mathsf{BIBE\text{-}CD}$ scheme $\prod^{\mathsf{CD}}_{\mathsf{BIBE}} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Digest}, \mathsf{ComputeKey}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ from a post-quantum secure classical BIBE. The construction is obtained by specializing the publicly verifiable certified-deletion transformation of Kitagawa *et al.* [18] to the BIBE functionality. The algorithms Digest and $\mathsf{ComputeKey}$ preserve the batching functionality of the underlying BIBE, while Enc outputs a ciphertext together with a public verification key vk, and the additional algorithms Del and Vrfy provide publicly verifiable deletion.

As shown in Appendix A, the resulting $\mathsf{BIBE\text{-}CD}$ scheme satisfies publicly verifiable certified-everlasting deletion. In particular, we use its computational consequence that, after a valid deletion certificate has been produced, the complete master secret key may subsequently be revealed without compromising the deleted challenge plaintext. Our subset-deletion proof uses this guarantee only on executions in which the aggregate deletion proof is accepted. Executions in which it is rejected never release the master secret key, and for them we use the selective semantic security of $\prod^{\mathsf{CD}}_{\mathsf{BIBE}}$ (Lemma 4).

3.3 Succinct Publicly Verifiable Subset Deletion

We now extend the single-ciphertext deletion functionality of $\prod^{\mathsf{CD}}_{\mathsf{BIBE}}$ to deletion of an arbitrary subset of a polynomial-size ciphertext collection. Suppose $((\mathsf{ct}_i, \mathsf{vk}_i))_{i\in[N]}$ is a collection of ciphertexts and public verification keys, and let $D \subseteq [N]$ be the subset to be deleted. For every $i \in D$, the holder runs the destructive deletion algorithm Del on ct_i and obtains an individual deletion certificate cert_i, while the ciphertexts at positions outside D are left untouched. To aggregate these certificates, define for every position $i \in [N]$ a bit d_i such that $d_i = 1$ if and only if $i \in D$, and let $x_i = (d_i, \mathsf{vk}_i)$. We consider the NP relation $\mathcal{R}_{\mathsf{del}}$ that accepts every witness when $d_i = 0$, while for $d_i = 1$ it accepts a witness w_i exactly when $\mathsf{Vrfy}(\mathsf{vk}_i, w_i) = \top$. Thus, for deleted positions the witness is the corresponding individual deletion certificate, whereas undeleted positions use a fixed dummy witness. A somewhere-extractable batch argument is then used to prove simultaneously that all statements $x_1, \ldots, x_N$ admit valid witnesses. The resulting subset-deletion certificate consists solely of the succinct batch proof π_D, whose size is $\mathsf{poly}(\lambda, \log N)$ and is independent of $|D|$.

Security is proved by switching the challenge ciphertexts one position at a time. Consider a position j for which the two challenge messages differ. The subset-security condition requires $j \in D$, so the corresponding statement is $x_j = (1, \mathsf{vk}^*_j)$. We generate the somewhere-extractable batch-argument reference string with extraction programmed at position j. Index hiding makes this change computationally indistinguishable. From any accepting aggregate proof, the extractor obtains a witness w_j satisfying $\mathcal{R}_{\mathsf{del}}((1, \mathsf{vk}^*_j), w_j) = 1$, and hence $\mathsf{Vrfy}(\mathsf{vk}^*_j, w_j) = \top$. Thus, w_j is a valid individual deletion certificate for the j-th challenge ciphertext. The corresponding message-switch transition can therefore

be reduced directly to the publicly verifiable certified-deletion security of $\prod^{\mathsf{CD}}_{\mathsf{BIBE}}$ established in Appendix A, on executions in which the aggregate proof is accepted.

A minor complication is that batch-key queries may be made before the challenge messages are submitted, and hence before the set $\Delta = \{i \in [N] : m_{0,i} \neq m_{1,i}\}$ is known. For a fixed hybrid position j, we therefore separate the event that the adversary ever requests a batch key capable of decrypting the challenge ciphertext associated with $(\mathsf{id}^*_j, \mathsf{tg}^*_j)$. On this event the reduction halts with a fixed output. Its contribution to the two adjacent hybrids is the same in both of the two possible cases. If the challenge messages eventually satisfy $m_{0,j} \neq m_{1,j}$, then $j \in \Delta$ and global admissibility makes both hybrids output 0. If instead $m_{0,j} = m_{1,j}$, then the two adjacent hybrids are identically distributed, so they assign the same probability to every execution. In both cases the difference of the two hybrid probabilities is unaffected. Consequently, the reduction to the single-ciphertext BIBE-CD security experiment only needs to simulate the complementary event, in which every forwarded batch-key query is admissible.

A second point requires care. The experiment declares an execution inadmissible as soon as some query authorizes a challenge attribute X^*_i with $i \in \Delta$, and not only for the single index j being switched. A reduction that tested the index j alone would simulate a strictly weaker experiment, and the resulting error terms need not cancel between adjacent hybrids. Since all challenge attributes are committed selectively and Δ becomes known at challenge time, the reduction can apply the experiment's own admissibility predicate locally to its recorded query transcript before producing its output. This costs nothing, because the check does not change which queries are forwarded to the single-ciphertext challenger.

Finally, the remaining branches are simulated explicitly. If the adversary chooses a subset D with $\Delta \not\subseteq D$, the reduction outputs the prescribed abort value. When aggregate verification fails, no master secret key is released. The adversary's final bit on such executions is bounded by a reduction to the selective semantic security of $\prod^{\mathsf{CD}}_{\mathsf{BIBE}}$, which supplies $\perp$ to the subset adversary and records its final bit. This reduction can recognize rejection by itself because aggregate verification is public. A separate reduction to certified-everlasting deletion handles the accepting executions and treats rejection as the terminal $\perp$ outcome of its own experiment. The two contributions add up to the difference of the adjacent hybrid probabilities. If aggregate verification succeeds but extraction fails, the certified-deletion reduction submits $\perp$; this happens only with negligible probability. A standard hybrid over the N ciphertext positions then establishes the computational subset-deletion guarantee.

3.4 Assumptions and Instantiation

All reductions in this paper are against QPT adversaries and selectively chosen target identities. Accordingly, every primitive below must be secure against QPT adversaries in the instantiation that is actually used, and not only in its formal statement.

The classical BIBE of Section 4 uses an indistinguishability obfuscator, an identity-based encryption scheme with puncturable master secret key that is post-quantum secure and satisfies the pointwise strong punctured-key correctness of Appendix 2.3, and an SSB hash function with QPT index hiding and somewhere statistical binding. The BIBE-CD scheme of Appendix A additionally uses a deterministic signature scheme with one-time unforgeability for BB84 states [18]. The subset-deletion construction of Section 5 additionally uses a publicly verifiable seBARG with QPT index hiding and a QPT-secure somewhere argument of knowledge with straight-line extraction (Appendix 2.5).
Post-quantum candidates for $i\mathcal{O}$ are given in [9, 14, 16, 22]; these are candidates rather than constructions from standard assumptions. The puncturable-master-key IBE of Çakan *et al.* [11] is built from $i\mathcal{O}$, a puncturable pseudorandom function, and a public-key encryption scheme, and therefore admits a post-quantum instantiation from post-quantum $i\mathcal{O}$ and LWE. Its identity keys are generated using puncturable-PRF outputs as explicit randomness, so keys derived from the punctured and unpunctured master keys agree pointwise; ordinary decryptability after puncturing would not suffice for our $i\mathcal{O}$ hybrid. Kitagawa *et al.* [18] construct the required signature scheme from one-way functions. Batch arguments with a hidden extraction index are constructed from LWE in [12, 15], and their extractor uses only the trapdoor, the statements, and the proof, so extraction never rewinds the adversary. We do not claim that the security analyses of these works carry over to QPT adversaries, and we therefore state QPT security of the somewhere argument of knowledge as an explicit assumption. Under these conditions, the full construction can be instantiated from post-quantum $i\mathcal{O}$ together with LWE. We do not give a parameterized concrete instantiation.

4 BIBE from IBE with Puncturable Master Secret Key

We first construct a post quantum BIBE, in the sense of Appendix 2.1. Certified deletion is not used in this section. Fix a message space $\mathcal{M}$, identity space $\mathcal{I} = \{0,1\}^k$, and batch-label space Γ. Let $\mathsf{id}_\perp \notin \mathcal{I}$ be a reserved dummy block. The construction uses

- an identity-based encryption scheme with puncturable master secret key $\prod_{\mathsf{PIBE}} = \mathsf{PIBE}.(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Punc}, \mathsf{Enc}, \mathsf{Dec})$ with identity space $\mathcal{X} = \mathcal{I} \times \Gamma$ (cf. Appendix 2.3)
- a somewhere statistically binding hash function $\prod^{\mathsf{id}}_{\mathsf{SSB}}$ over blocks encoding elements of $\mathcal{I} \cup \{\mathsf{id}_\perp\}$ (cf. Appendix 2.4)
- an indistinguishability obfuscator $i\mathcal{O}$ (cf. Appendix 2.2).

For every $S \subseteq \mathcal{I}$ with $|S| \leq B$, let $\mathbf{S} = \mathsf{Pad}_B(S) = (\mathsf{id}_1, \ldots, \mathsf{id}_B)$ denote the canonical representation obtained by sorting the elements of S lexicographically and padding unused positions with $\mathsf{id}_\perp$. For every actual $\mathsf{id} \in S$, let $\mathrm{idx}_S(\mathsf{id})$ denote its unique position in $\mathbf{S}$. All circuits appearing below are padded to a common polynomial upper bound $\ell_{\max}(\lambda, k, \log B)$.

4.1 Construction

Following are the algorithms of $\prod_{\mathsf{BIBE}} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Digest}, \mathsf{ComputeKey}, \mathsf{Dec})$.

$\mathsf{Setup}(1^\lambda, 1^B)$:
- Compute $(\mathsf{mpk}_0, \mathsf{msk}_0) \leftarrow \mathsf{PIBE.Setup}(1^\lambda)$.
- Compute $\mathsf{hk}_{\mathsf{id}} \leftarrow \mathsf{SSB.Setup}(1^\lambda, 1^{k+1}, B, 1)$.
- Output $\mathsf{mpk} = (\mathsf{mpk}_0, \mathsf{hk}_{\mathsf{id}})$ and $\mathsf{msk} = (\mathsf{msk}_0, \mathsf{hk}_{\mathsf{id}})$.

$\mathsf{Enc}(\mathsf{mpk}, m, \mathsf{id}, \mathsf{tg})$:
- Parse $\mathsf{mpk} = (\mathsf{mpk}_0, \mathsf{hk}_{\mathsf{id}})$ and set $X = (\mathsf{id}, \mathsf{tg})$.
- Compute $\mathsf{CT} \leftarrow \mathsf{PIBE.Enc}(\mathsf{mpk}_0, X, m)$.
- Output $\mathsf{ct} = (\mathsf{id}, \mathsf{tg}, \mathsf{CT})$.

$\mathsf{Digest}(\mathsf{mpk}, S)$:
- Parse $\mathsf{mpk} = (\mathsf{mpk}_0, \mathsf{hk}_{\mathsf{id}})$ and compute the representation $\mathbf{S} = \mathsf{Pad}_B(S)$.
- Compute $h \leftarrow \mathsf{SSB.Hash}(\mathsf{hk}_{\mathsf{id}}, \mathbf{S})$.
- Output $\mathsf{dig} = h$.

$\mathsf{ComputeKey}(\mathsf{msk}, \mathsf{dig}, \mathsf{tg})$:
- Parse $\mathsf{msk} = (\mathsf{msk}_0, \mathsf{hk}_{\mathsf{id}})$ and $\mathsf{dig} = h$.
- Compute $\widetilde{\mathcal{C}} \leftarrow i\mathcal{O}\big(\mathcal{C}_{\mathsf{key}}[\mathsf{msk}_0, \mathsf{hk}_{\mathsf{id}}, h, \mathsf{tg}]\big)$, where $\mathcal{C}_{\mathsf{key}}$ is Fig. 1.
- Output $\mathsf{sk} = (\mathsf{hk}_{\mathsf{id}}, \widetilde{\mathcal{C}})$.

Membership-Gated Key-Release Circuit $\mathcal{C}_{\mathsf{key}}$

Input: (id, i, π)
Hardcoded: $\mathsf{msk}_0, \mathsf{hk}_{\mathsf{id}}, h, \mathsf{tg}$
1. If $i \notin [B]$, output $\perp$.
2. If $\mathsf{SSB.Vrfy}(\mathsf{hk}_{\mathsf{id}}, h, i, \mathsf{id}, \pi) \neq 1$, output $\perp$.
3. If $\mathsf{id} = \mathsf{id}_\perp$, output $\perp$.
4. Set $X = (\mathsf{id}, \mathsf{tg})$.
5. Compute $\mathsf{sk}_X := \mathsf{PIBE.KeyGen}(\mathsf{msk}_0, X)$.
6. Output sk_X.

Fig. 1: Membership-Gated Key-Release Circuit

$\mathsf{Dec}(\mathsf{sk}, S, (\mathsf{id}, \mathsf{tg}), \mathsf{ct})$:
- Parse $\mathsf{ct} = (\mathsf{id}', \mathsf{tg}', \mathsf{CT})$ and $\mathsf{sk} = (\mathsf{hk}_{\mathsf{id}}, \widetilde{\mathcal{C}})$.
- If $\mathsf{id}' \neq \mathsf{id}$ or $\mathsf{tg}' \neq \mathsf{tg}$, output $\perp$.
- If $\mathsf{id} \notin S$, output $\perp$.
- Compute $\mathbf{S} = \mathsf{Pad}_B(S)$ and $i = \mathrm{idx}_S(\mathsf{id})$.
- Compute $\pi \leftarrow \mathsf{SSB.Open}(\mathsf{hk}_{\mathsf{id}}, \mathbf{S}, i)$.
- Compute $\mathsf{sk}_X \leftarrow \widetilde{\mathcal{C}}(\mathsf{id}, i, \pi)$.
- If $\mathsf{sk}_X = \perp$, output $\perp$.
- Compute and output $m' \leftarrow \mathsf{PIBE.Dec}(\mathsf{sk}_X, \mathsf{CT})$.

4.2 Correctness

Theorem 1 (Correctness). *If $\prod_{\mathsf{PIBE}}$, $\prod^{\mathsf{id}}_{\mathsf{SSB}}$, and $i\mathcal{O}$ are correct, then the construction in Section 4.1 satisfies decryption correctness.*

Proof. Fix $\lambda, B \in \mathbb{N}$, $m \in \mathcal{M}$, $\mathsf{id} \in \mathcal{I}$, $\mathsf{tg} \in \Gamma$, and $S \subseteq \mathcal{I}$ such that $|S| \leq B$ and $\mathsf{id} \in S$. Let $\mathbf{S} = \mathsf{Pad}_B(S)$ and let $i = \mathrm{idx}_S(\mathsf{id})$. The digest is $h \leftarrow \mathsf{SSB.Hash}(\mathsf{hk}_{\mathsf{id}}, \mathbf{S})$. During decryption, $\pi \leftarrow \mathsf{SSB.Open}(\mathsf{hk}_{\mathsf{id}}, \mathbf{S}, i)$. Since the i-th block of $\mathbf{S}$ is id, correctness of the SSB scheme gives $\mathsf{SSB.Vrfy}(\mathsf{hk}_{\mathsf{id}}, h, i, \mathsf{id}, \pi) = 1$. Hence $\mathcal{C}_{\mathsf{key}}$ sets $X = (\mathsf{id}, \mathsf{tg})$ and computes $\mathsf{sk}_X := \mathsf{PIBE.KeyGen}(\mathsf{msk}_0, X)$. Correctness of $i\mathcal{O}$ implies that $\widetilde{\mathcal{C}}(\mathsf{id}, i, \pi)$ returns the same sk_X. The ciphertext component CT was generated as $\mathsf{PIBE.Enc}(\mathsf{mpk}_0, X, m)$. Correctness of $\prod_{\mathsf{PIBE}}$ therefore implies $\mathsf{PIBE.Dec}(\mathsf{sk}_X, \mathsf{CT}) = m$ except with negligible probability.

4.3 Selective Security of the Classical BIBE

We use the following selective security notion, which is the notion of Appendix 2.1. The adversary commits to a challenge identity-label pair $(\mathsf{id}^*, \mathsf{tg}^*)$ before setup. The challenger generates $(\mathsf{mpk}, \mathsf{msk})$ and answers adaptive batch-key queries (S, tg). The global admissibility condition requires $\mathsf{tg} \neq \mathsf{tg}^*$ or $\mathsf{id}^* \notin S$ for every key query made at any point in the experiment. At challenge time the adversary submits (m_0, m_1), receives $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, m_b, \mathsf{id}^*, \mathsf{tg}^*)$, continues making admissible key queries, and finally guesses b.

Theorem 2 (Selective Security). *Suppose $\prod^{\mathsf{id}}_{\mathsf{SSB}}$ satisfies post-quantum index hiding and somewhere statistical binding, $i\mathcal{O}$ is secure against QPT distinguishers, and $\prod_{\mathsf{PIBE}}$ satisfies post-quantum puncturable-master-secret-key security and strong punctured-key correctness. Then the construction in Section 4.1 is selectively secure.*

Proof. Fix a QPT adversary $\mathcal{A}$ making at most $q = q(\lambda)$ batch-key queries. Let $X^* = (\mathsf{id}^*, \mathsf{tg}^*)$ be the selective challenge identity of the underlying scheme. For every $j \in \{0, \ldots, B\}$, define the circuit $\mathcal{C}_j$ shown in Fig. 2.

Position-j Key-Release Circuit $\mathcal{C}_j$

Input: (id, i, π)
Hardcoded: $\mathsf{msk}_0, \mathsf{hk}_{\mathsf{id}}, h, \mathsf{tg}, X^* = (\mathsf{id}^*, \mathsf{tg}^*)$

1. If $i \notin [B]$, output $\perp$.
2. If $\mathsf{SSB.Vrfy}(\mathsf{hk}_{\mathsf{id}}, h, i, \mathsf{id}, \pi) \neq 1$, output $\perp$.
3. If $\mathsf{id} = \mathsf{id}_{\perp}$, output $\perp$.
4. If $\mathsf{tg} = \mathsf{tg}^*$, $\mathsf{id} = \mathsf{id}^*$, and $i \leq j$, output $\perp$.
5. Set $X = (\mathsf{id}, \mathsf{tg})$ and compute $\mathsf{sk}_X := \mathsf{PIBE.KeyGen}(\mathsf{msk}_0, X)$.
6. Output sk_X.

Fig. 2: Position-j Key-Release Circuit

For $b \in \{0,1\}$, let $\mathsf{G}_j(b)$ denote the selective BIBE experiment with challenge bit b in which every batch-key query is answered using an obfuscation of $\mathcal{C}_j$. For $j = 0$, the additional rejection condition never holds and $\mathsf{G}_0(b)$ is the real experiment.

Lemma 1. *For every $j \in [B]$ and $b \in \{0,1\}$, $|\Pr[\mathsf{G}_{j-1}(b) = 1] - \Pr[\mathsf{G}_j(b) = 1]| \leq \mathsf{negl}(\lambda)$.*

Proof. Fix $j \in [B]$. Define $\widehat{\mathsf{G}}^{(j)}_{j-1}(b)$ and $\widehat{\mathsf{G}}^{(j)}_{j}(b)$ exactly as the adjacent hybrids except that the global SSB key is sampled as $\mathsf{hk}_{\mathsf{id}} \leftarrow \mathsf{SSB.Setup}(1^\lambda, 1^{k+1}, B, j)$. By index hiding, $\mathsf{G}_{j-1}(b) \approx_c \widehat{\mathsf{G}}^{(j)}_{j-1}(b)$ and $\widehat{\mathsf{G}}^{(j)}_{j}(b) \approx_c \mathsf{G}_j(b)$.
It remains to compare the two experiments under a key statistically binding at position j. Condition on the event that $\mathsf{hk}_{\mathsf{id}}$ is statistically binding at j. This event fails only with negligible probability.
Fix an arbitrary batch-key query (S, tg) and let $\mathbf{S} = \mathsf{Pad}_B(S) = (\mathsf{id}_1, \ldots, \mathsf{id}_B)$. Let $h = \mathsf{SSB.Hash}(\mathsf{hk}_{\mathsf{id}}, \mathbf{S})$. If $\mathsf{tg} \neq \mathsf{tg}^*$, the additional rejection condition never fires and the two circuits are literally identical. Assume therefore that $\mathsf{tg} = \mathsf{tg}^*$. Global admissibility gives $\mathsf{id}^* \notin S$. Since $\mathsf{id}^* \neq \mathsf{id}_\perp$, the block at position j satisfies $\mathsf{id}_j \neq \mathsf{id}^*$. By SSB correctness, there exists an honest opening π_j such that $\mathsf{SSB.Vrfy}(\mathsf{hk}_{\mathsf{id}}, h, j, \mathsf{id}_j, \pi_j) = 1$. Statistical binding at position j therefore implies that no opening π can satisfy $\mathsf{SSB.Vrfy}(\mathsf{hk}_{\mathsf{id}}, h, j, \mathsf{id}^*, \pi) = 1$. The circuits $\mathcal{C}_{j-1}$ and $\mathcal{C}_j$ can differ only on an input for which $\mathsf{tg} = \mathsf{tg}^*$, $\mathsf{id} = \mathsf{id}^*$, and $i = j$. The argument above shows that no such input can pass the SSB verification check. Hence the two circuits agree on every possible input for this batch-key query. After padding both circuits to the same size, security of $i\mathcal{O}$ implies indistinguishability of their obfuscations.
Applying a standard hybrid over the at most q batch-key queries and then removing the conditioning event by a union bound proves $\widehat{\mathsf{G}}^{(j)}_{j-1}(b) \approx_c \widehat{\mathsf{G}}^{(j)}_{j}(b)$. Combining the two index-hiding transitions gives the claim.

Applying Lemma 1 for $j = 1, \ldots, B$ gives $|\Pr[\mathsf{G}_0(b) = 1] - \Pr[\mathsf{G}_B(b) = 1]| \leq B \cdot \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda)$. In $\mathsf{G}_B(b)$, every valid circuit input with $\mathsf{tg} = \mathsf{tg}^*$ and $\mathsf{id} = \mathsf{id}^*$ is rejected before identity-key generation. We may therefore puncture the underlying master secret key at X^*.Compute $\mathsf{msk}_{X^*} \leftarrow \mathsf{PIBE.Punc}(\mathsf{msk}_0, X^*)$. Define the punctured circuit $\widehat{\mathcal{C}}_B$ shown in Fig. 3. Let $\mathsf{G}_{\mathsf{punct}}(b)$ be identical to $\mathsf{G}_B(b)$ except that batch-key queries are answered using obfuscations of $\widehat{\mathcal{C}}_B$.

Lemma 2. *For every $b \in \{0,1\}$, $|\Pr[\mathsf{G}_B(b) = 1] - \Pr[\mathsf{G}_{\mathsf{punct}}(b) = 1]| \leq \mathsf{negl}(\lambda)$.*

Proof. Fix an arbitrary batch-key query and circuit input. If any range, dummy-identity, or SSB verification check fails, both circuits output $\perp$. If $X = (\mathsf{id}, \mathsf{tg}) = X^*$, both circuits output $\perp$ before key generation. For every $X \neq X^*$, strong punctured-key correctness gives $\mathsf{PIBE.KeyGen}(\mathsf{msk}_{X^*}, X) = \mathsf{PIBE.KeyGen}(\mathsf{msk}_0, X)$ with probability one over setup and puncturing. Thus the two circuits are functionally identical. After padding to equal size, security of $i\mathcal{O}$ and a hybrid over polynomially many batch-key queries prove the claim.

Punctured Key-Release Circuit $\widehat{\mathcal{C}}_B$

Input: (id, i, π)
Hardcoded: $\mathsf{msk}_{X^*}, \mathsf{hk}_{\mathsf{id}}, h, \mathsf{tg}, X^*$

1. If $i \notin [B]$, output $\perp$.
2. If $\mathsf{SSB.Vrfy}(\mathsf{hk}_{\mathsf{id}}, h, i, \mathsf{id}, \pi) \neq 1$, output $\perp$.
3. If $\mathsf{id} = \mathsf{id}_{\perp}$, output $\perp$.
4. If $\mathsf{tg} = \mathsf{tg}^*$ and $\mathsf{id} = \mathsf{id}^*$, output $\perp$.
5. Set $X = (\mathsf{id}, \mathsf{tg})$. Compute $\mathsf{sk}_X := \mathsf{PIBE.KeyGen}(\mathsf{msk}_{X^*}, X)$.
6. Output sk_X.

Fig. 3: Punctured Key-Release Circuit

Lemma 3. *If $\prod_{\mathsf{PIBE}}$ satisfies post-quantum puncturable-master-secret-key security, then* $|\Pr[\mathsf{G}_{\mathsf{punct}}(0) = 1] - \Pr[\mathsf{G}_{\mathsf{punct}}(1) = 1]| \leq \mathsf{negl}(\lambda)$.

Proof. Suppose a QPT adversary $\mathcal{A}$ distinguishes the two experiments with non-negligible advantage. We construct an adversary $\mathcal{B}$ against puncturable-master-secret-key security of $\prod_{\mathsf{PIBE}}$. The adversary $\mathcal{B}$ commits to $X^* = (\mathsf{id}^*, \mathsf{tg}^*)$ and receives $(\mathsf{mpk}_0, \mathsf{msk}_{X^*})$ from its challenger. It samples $\mathsf{hk}_{\mathsf{id}} \leftarrow \mathsf{SSB.Setup}(1^\lambda, 1^{k+1}, B, 1)$, sets $\mathsf{mpk} = (\mathsf{mpk}_0, \mathsf{hk}_{\mathsf{id}})$, and gives mpk to $\mathcal{A}$.
For a batch-key query (S, tg), the reduction computes $\mathbf{S} = \mathsf{Pad}_B(S)$ and $h \leftarrow \mathsf{SSB.Hash}(\mathsf{hk}_{\mathsf{id}}, \mathbf{S})$. It obfuscates $\widehat{\mathcal{C}}_B[\mathsf{msk}_{X^*}, \mathsf{hk}_{\mathsf{id}}, h, \mathsf{tg}, X^*]$ and returns the resulting batch key. This requires only msk_{X^*}. When $\mathcal{A}$ submits (m_0, m_1), the reduction forwards them to its puncturable-master-secret-key security challenger and receives CT_b. It returns $\mathsf{ct}^* = (\mathsf{id}^*, \mathsf{tg}^*, \mathsf{CT}_b)$ to $\mathcal{A}$. All later key queries are answered as above. The view of $\mathcal{A}$ is exactly $\mathsf{G}_{\mathsf{punct}}(b)$. Hence any non-negligible distinguishing advantage contradicts puncturable-master-secret-key security.

Combining Lemmas 1, 2, and 3 proves Theorem 2.

4.4 Succinctness

Theorem 3 (Succinctness). *The construction in Section 4.1 is succinct.*

Proof. By SSB succinctness, $|\mathsf{dig}| = |h| = \mathsf{poly}(\lambda, k)$ and $|\mathsf{hk}_{\mathsf{id}}| = \mathsf{poly}(\lambda, k, \log B)$. The circuit $\mathcal{C}_{\mathsf{key}}$ contains one SSB verification and one identity-key generation computation. Its size is therefore $\mathsf{poly}(\lambda, k, \log B)$. The output size and running time of $i\mathcal{O}$ are polynomial in the circuit size and security parameter. Consequently, $|\mathsf{sk}|$ and the running time of $\mathsf{ComputeKey}$ are bounded by $\mathsf{poly}(\lambda, k, \log B)$. The algorithm Digest must read the entire batch and therefore may require $O(B)$ local computation. Similarly, Dec is given the batch S and may spend $O(B)$ time constructing its canonical representation or locating the requested identity. These costs are not part of the authority-side succinctness guarantee.

5 BIBE with Succinct Publicly Verifiable Subset Deletion

We now define batched identity-based encryption with succinct publicly verifiable subset deletion and provide our generic construction.

5.1 Syntax and Correctness

Let $\mathcal{I}$ be an identity space, $\mathcal{M}$ be a message space, and Γ be a batch-label space. Let B denote the maximum batch size and let N denote the maximum number of ciphertexts for which subset deletion is supported. A batched identity-based encryption scheme with succinct publicly verifiable subset deletion is a tuple $\prod_{\mathsf{BIBE}}^{\mathsf{SPVD}} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Digest}, \mathsf{ComputeKey}, \mathsf{Dec}, \mathsf{DelSet}, \mathsf{VrfySet})$ with the following algorithms.

- $\mathsf{Setup}(1^\lambda, 1^B, 1^N) \to (\mathsf{mpk}, \mathsf{msk})$: On input the security parameter, the maximum batch size B, and the maximum ciphertext-collection size N, outputs a master public key mpk and a master secret key msk.
- $\mathsf{Enc}(\mathsf{mpk}, m, \mathsf{id}, \mathsf{tg}) \to (\mathsf{ct}, \mathsf{vk})$: On input the master public key, a message $m \in \mathcal{M}$, an identity $\mathsf{id} \in \mathcal{I}$, and a batch label $\mathsf{tg} \in \Gamma$, outputs a possibly quantum ciphertext ct and a classical public verification key vk.
- $\mathsf{Digest}(\mathsf{mpk}, S) \to \mathsf{dig}$: On input the master public key and a set $S \subseteq \mathcal{I}$ with $|S| \leq B$, outputs a public batch digest dig.
- $\mathsf{ComputeKey}(\mathsf{msk}, \mathsf{dig}, \mathsf{tg}) \to \mathsf{sk}$: On input the master secret key, a batch digest dig, and a batch label tg, outputs a batch secret key sk.
- $\mathsf{Dec}(\mathsf{sk}, S, (\mathsf{id}, \mathsf{tg}), \mathsf{ct}) \to m/\perp$: On input a batch secret key, a batch S, an identity-label pair $(\mathsf{id}, \mathsf{tg})$, and a ciphertext, outputs a message or $\perp$.
- $\mathsf{DelSet}(\mathsf{mpk}, \mathbf{ct}, \mathbf{vk}, D) \to (\mathsf{cert}_D, \mathbf{ct}_{\overline{D}})$: On input $\mathbf{ct} = (\mathsf{ct}_1, \ldots, \mathsf{ct}_N)$, $\mathbf{vk} = (\mathsf{vk}_1, \ldots, \mathsf{vk}_N)$, and $D \subseteq [N]$, performs deletion only on the ciphertexts indexed by D, outputs a single classical subset-deletion certificate cert_D, and returns the remaining quantum ciphertexts $\mathbf{ct}_{\overline{D}} = (\mathsf{ct}_i)_{i \notin D}$ without measuring or discarding them. The second output is kept by the ciphertext holder and is not part of the certificate or of the verifier input.
- $\mathsf{VrfySet}(\mathsf{mpk}, \mathbf{vk}, D, \mathsf{cert}_D) \to \top/\perp$: On input the master public key, the public verification-key vector, the claimed deleted subset D, and a subset-deletion certificate, publicly verifies the certificate and outputs $\top$ or $\perp$.

We require the following correctness properties.

- **Decryption correctness.** For every security parameter $\lambda \in \mathbb{N}$, every $S \subseteq \mathcal{I}$ with $|S| \leq B$, every $\mathsf{id} \in S$, every $\mathsf{tg} \in \Gamma$, and every $m \in \mathcal{M}$,

$$\Pr\left[\begin{array}{l|l} \mathsf{Dec}(\mathsf{sk}, S, (\mathsf{id}, \mathsf{tg}), & (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^B, 1^N), \\ \mathsf{ct}) = m & \mathsf{dig} \leftarrow \mathsf{Digest}(\mathsf{mpk}, S), \\ & \mathsf{sk} \leftarrow \mathsf{ComputeKey}(\mathsf{msk}, \mathsf{dig}, \mathsf{tg}), \\ & (\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(\mathsf{mpk}, m, \mathsf{id}, \mathsf{tg}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

- **Subset-deletion correctness.** For every security parameter $\lambda \in \mathbb{N}$, every collection of identity-label pairs $((\mathsf{id}_i, \mathsf{tg}_i))_{i\in[N]}$, every message vector $(m_i)_{i\in[N]}$, and every subset $D \subseteq [N]$, $\Pr[\mathsf{VrfySet}(\mathsf{mpk}, \mathbf{vk}, D, \mathsf{cert}_D) = \top] \geq 1-\mathsf{negl}(\lambda)$, where $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^B, 1^N)$, $(\mathsf{ct}_i, \mathsf{vk}_i) \leftarrow \mathsf{Enc}(\mathsf{mpk}, m_i, \mathsf{id}_i, \mathsf{tg}_i)$ for every $i \in [N]$, and $(\mathsf{cert}_D, \mathbf{ct}_{\overline{D}}) \leftarrow \mathsf{DelSet}(\mathsf{mpk}, \mathbf{ct}, \mathbf{vk}, D)$. The requirement is stated for honestly generated ciphertext collections only.
- **Preservation of undeleted ciphertexts.** The honest procedure DelSet acts only on the ciphertexts indexed by D and as the identity on all other ciphertexts, so $\mathbf{ct}_{\overline{D}}$ consists of the original undeleted ciphertexts. In the setting of subset-deletion correctness, we require that for every $i \notin D$, every $S \subseteq \mathcal{I}$ with $|S| \leq B$ and $\mathsf{id}_i \in S$, and $\mathsf{sk} \leftarrow \mathsf{ComputeKey}(\mathsf{msk}, \mathsf{Digest}(\mathsf{mpk}, S), \mathsf{tg}_i)$, the returned ciphertext ct_i satisfies $\Pr[\mathsf{Dec}(\mathsf{sk}, S, (\mathsf{id}_i, \mathsf{tg}_i), \mathsf{ct}_i) = m_i] \geq 1-\mathsf{negl}(\lambda)$.

5.2 Selective Subset-Deletion Security

We define selective security directly for $\prod_{\mathsf{BIBE}}^{\mathsf{SPVD}}$. For a bit $b \in \{0,1\}$, consider the experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{sel\text{-}subset\text{-}cd}}(\lambda, B, N, b)$ between a challenger and a QPT adversary $\mathcal{A}$.

- ◇ **Target Identities and Labels:** Before setup, the adversary outputs challenge identity-label pairs $\big((\mathsf{id}_1^*, \mathsf{tg}_1^*), \ldots, (\mathsf{id}_N^*, \mathsf{tg}_N^*)\big)$.
- ◇ **Setup:** The challenger computes $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^B, 1^N)$ and sends mpk to the adversary.
- ◇ **Query Phase 1:** The adversary adaptively submits batch-key queries (S, tg) with $S \subseteq \mathcal{I}$ and $|S| \leq B$. For every query, the challenger computes $\mathsf{dig} \leftarrow \mathsf{Digest}(\mathsf{mpk}, S)$ and $\mathsf{sk} \leftarrow \mathsf{ComputeKey}(\mathsf{msk}, \mathsf{dig}, \mathsf{tg})$, and sends sk to the adversary.
- ◇ **Challenge:** The adversary outputs two equal-length message vectors $\mathbf{m}_0 = (m_{0,1}, \ldots, m_{0,N})$ and $\mathbf{m}_1 = (m_{1,1}, \ldots, m_{1,N})$. For every $i \in [N]$, the challenger computes $(\mathsf{ct}_i^*, \mathsf{vk}_i^*) \leftarrow \mathsf{Enc}(\mathsf{mpk}, m_{b,i}, \mathsf{id}_i^*, \mathsf{tg}_i^*)$ and sends $((\mathsf{ct}_i^*, \mathsf{vk}_i^*))_{i\in[N]}$ to the adversary.
  Set $\Delta = \{i \in [N] : m_{0,i} \neq m_{1,i}\}$.
- ◇ **Query Phase 2:** The adversary may continue to issue adaptive batch-key queries (S, tg). The challenger answers them as in Query Phase 1.
- ◇ **Global Admissibility:** For every batch-key query (S, tg) made in either query phase and every $i \in \Delta$, it must hold that $\mathsf{tg} \neq \mathsf{tg}_i^*$ or $\mathsf{id}_i^* \notin S$. If this condition is violated, the experiment outputs 0.
- ◇ **Subset Deletion:** The adversary outputs a subset $D \subseteq [N]$ and a purported subset-deletion certificate cert_D.
  If $\Delta \not\subseteq D$, the experiment outputs 0.
  Otherwise, the challenger computes $\mu \leftarrow \mathsf{VrfySet}(\mathsf{mpk}, (\mathsf{vk}_i^*)_{i\in[N]}, D, \mathsf{cert}_D)$.
  If $\mu = \top$, the challenger sends the complete master secret key msk to the adversary. If $\mu = \bot$, the challenger sends $\bot$.

$\diamond$ **Guess:** The adversary outputs a bit $b' \in \{0,1\}$.

We say that $\prod_{\mathsf{BIBE}}^{\mathsf{SPVD}}$ satisfies *post-quantum selective subset-deletion security* if, for every polynomially bounded $B = B(\lambda)$ and $N = N(\lambda)$ and every QPT adversary

$$\left|\Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{sel\text{-}subset\text{-}cd}}(\lambda, B, N, 0) = 1] - \Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{sel\text{-}subset\text{-}cd}}(\lambda, B, N, 1) = 1]\right| \leq \mathsf{negl}(\lambda).$$

The challenge identity-label pairs are selective, while the challenge messages and the deleted subset D are chosen adaptively. The condition $\Delta \subseteq D$ requires deletion of every ciphertext position on which the two challenge message vectors differ. No deletion requirement is imposed on positions for which $m_{0,i} = m_{1,i}$.
This notion is computational: the adversary remains QPT after it receives msk, and only its final bit is compared. It is therefore weaker than the certified-everlasting notion of Definition 1, which is satisfied by each individual ciphertext. We do not claim everlasting security for subset deletion, since our proof switches the seBARG extraction position using computational index hiding in experiments that include the post-deletion phase.

5.3 Construction

Our construction uses the publicly verifiable $\mathsf{BIBE\text{-}CD}$ scheme $\prod_{\mathsf{BIBE}}^{\mathsf{CD}} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Digest}, \mathsf{ComputeKey}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ constructed in Appendix A. This scheme is obtained by applying the publicly verifiable certified-deletion transformation of Kitagawa *et al.* [18] directly to a post-quantum secure classical BIBE, as constructed in Section 4. The algorithms Digest and $\mathsf{ComputeKey}$ preserve the batching functionality of the underlying BIBE, while Enc outputs a ciphertext together with a public verification key vk, and Del and Vrfy provide publicly verifiable deletion. As shown in Appendix A, the resulting scheme satisfies publicly verifiable certified-everlasting deletion and, in particular, remains computationally secure when the complete master secret key is revealed after successful verification of a deletion certificate, in the joint sense of Corollary 2. It also satisfies selective semantic security even when the verification key is given (Lemma 4). The proof of Theorem 5 uses both properties.
We use a somewhere-extractable batch argument $\prod_{\mathsf{seBARG}} = (\mathsf{Gen}, \mathsf{P}, \mathsf{V}, \mathsf{E})$ as defined in Appendix 2.5. We assume that the public verification key and deletion certificate of $\prod_{\mathsf{BIBE}}^{\mathsf{CD}}$ are classical strings and that Vrfy is a deterministic polynomial-time algorithm, so that deletion verification defines an NP relation. As in Appendix A.1, we use the convention $\mathsf{Vrfy}(\mathsf{vk}, \bot) = \bot$.

Deletion Relation Let $\ell_{\mathsf{cert}} = \mathsf{poly}(\lambda)$ be a common padded length for individual deletion certificates. For $d \in \{0,1\}$, a public verification key vk, and a witness $w \in \{0,1\}^{\ell_{\mathsf{cert}}}$, define the relation $\mathcal{R}_{\mathsf{del}}$ by the circuit in Fig. 4. Let $\ell_x = \mathsf{poly}(\lambda)$ be an upper bound on the encoding length of a statement $x = (d, \mathsf{vk})$. The circuit $\mathcal{C}_{\mathcal{R}_{\mathsf{del}}}$ computing $\mathcal{R}_{\mathsf{del}}$ contains a single invocation of $\mathsf{BIBE\text{-}CD.Vrfy}$ and therefore has size $\mathsf{poly}(\lambda)$, independent of N and of B. The algorithms of $\prod_{\mathsf{BIBE}}^{\mathsf{SPVD}}$ are defined as follows.

$\mathsf{Setup}(1^\lambda, 1^B, 1^N)$:
- Compute $(\mathsf{mpk}_0, \mathsf{msk}_0) \leftarrow \mathsf{BIBE\text{-}CD.Setup}(1^\lambda, 1^B)$.
- Compute $(\mathsf{crs}, \mathsf{td}_1) \leftarrow \mathsf{seBARG.Gen}(1^\lambda, N, 1^{\ell_x}, 1^{\ell_{\mathsf{cert}}}, 1)$.
- Output $\mathsf{mpk} = (\mathsf{mpk}_0, \mathsf{crs})$ and $\mathsf{msk} = \mathsf{msk}_0$.

$\mathsf{Enc}(\mathsf{mpk}, m, \mathsf{id}, \mathsf{tg})$:
- Parse $\mathsf{mpk} = (\mathsf{mpk}_0, \mathsf{crs})$.
- Compute and output $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{BIBE\text{-}CD.Enc}(\mathsf{mpk}_0, m, \mathsf{id}, \mathsf{tg})$.

$\mathsf{Digest}(\mathsf{mpk}, S)$:
- Parse $\mathsf{mpk} = (\mathsf{mpk}_0, \mathsf{crs})$.
- Compute and output $\mathsf{dig} \leftarrow \mathsf{BIBE\text{-}CD.Digest}(\mathsf{mpk}_0, S)$.

$\mathsf{ComputeKey}(\mathsf{msk}, \mathsf{dig}, \mathsf{tg})$:
- Compute and output $\mathsf{sk} \leftarrow \mathsf{BIBE\text{-}CD.ComputeKey}(\mathsf{msk}, \mathsf{dig}, \mathsf{tg})$.

$\mathsf{Dec}(\mathsf{sk}, S, (\mathsf{id}, \mathsf{tg}), \mathsf{ct})$:
- Compute and output $m' \leftarrow \mathsf{BIBE\text{-}CD.Dec}(\mathsf{sk}, S, (\mathsf{id}, \mathsf{tg}), \mathsf{ct})$.

$\mathsf{DelSet}(\mathsf{mpk}, \mathbf{ct}, \mathbf{vk}, D)$:
- Parse $\mathsf{mpk} = (\mathsf{mpk}_0, \mathsf{crs})$, $\mathbf{ct} = (\mathsf{ct}_1, \ldots, \mathsf{ct}_N)$, and $\mathbf{vk} = (\mathsf{vk}_1, \ldots, \mathsf{vk}_N)$.
- For every $i \in [N]$, set $d_i = 1$ if $i \in D$ and set $d_i = 0$ otherwise. Set $x_i = (d_i, \mathsf{vk}_i)$.
- For every $i \in D$, compute $\mathsf{cert}_i \leftarrow \mathsf{BIBE\text{-}CD.Del}(\mathsf{ct}_i)$ and set $w_i = \mathsf{cert}_i$.
- For every $i \notin D$, set $w_i = 0^{\ell_{\mathsf{cert}}}$ and leave ct_i untouched.
- Compute $\pi_D \leftarrow \mathsf{seBARG.P}(\mathsf{crs}, x_1, \ldots, x_N, w_1, \ldots, w_N)$.
- Output $(\mathsf{cert}_D, \mathbf{ct}_{\overline{D}})$, where $\mathsf{cert}_D = \pi_D$ and $\mathbf{ct}_{\overline{D}} = (\mathsf{ct}_i)_{i \notin D}$ is returned unchanged.

$\mathsf{VrfySet}(\mathsf{mpk}, \mathbf{vk}, D, \mathsf{cert}_D)$:
- Parse $\mathsf{mpk} = (\mathsf{mpk}_0, \mathsf{crs})$, $\mathbf{vk} = (\mathsf{vk}_1, \ldots, \mathsf{vk}_N)$, and $\mathsf{cert}_D = \pi_D$.
- For every $i \in [N]$, set $d_i = 1$ if $i \in D$ and set $d_i = 0$ otherwise. Set $x_i = (d_i, \mathsf{vk}_i)$.
- Compute $\mu \leftarrow \mathsf{seBARG.V}(\mathsf{crs}, x_1, \ldots, x_N, \pi_D)$.
- Output $\top$ if $\mu = 1$. Otherwise, output $\bot$.

Subset-Deletion Relation $\mathcal{R}_{\mathsf{del}}$

Statement: $x = (d, \mathsf{vk})$
Witness: $w \in \{0,1\}^{\ell_{\mathsf{cert}}}$
1. If $d = 0$, output 1.
2. Compute $\mu \leftarrow \mathsf{BIBE\text{-}CD.Vrfy}(\mathsf{vk}, w)$.
3. If $\mu = \top$, output 1. Otherwise, output 0.

Fig. 4: Subset-Deletion Relation

5.4 Correctness

Theorem 4 (Correctness). *Suppose $\prod^{\mathsf{CD}}_{\mathsf{BIBE}}$ satisfies decryption correctness and deletion correctness, and $\prod_{\mathsf{seBARG}}$ satisfies completeness. Then the construction in Section 5.3 satisfies decryption correctness, subset-deletion correctness, and preservation of undeleted ciphertexts.*

Proof. Decryption correctness follows immediately from decryption correctness of $\prod^{\mathsf{CD}}_{\mathsf{BIBE}}$, since the algorithms Digest, $\mathsf{ComputeKey}$, and Dec are inherited directly from that scheme.

It remains to prove subset-deletion correctness. Fix a ciphertext collection $((\mathsf{ct}_i, \mathsf{vk}_i))_{i\in[N]}$ and a subset $D \subseteq [N]$. For every $i \in D$, honest deletion computes $\mathsf{cert}_i \leftarrow \mathsf{BIBE\text{-}CD.Del}(\mathsf{ct}_i)$. By deletion correctness, $\mathsf{BIBE\text{-}CD.Vrfy}(\mathsf{vk}_i, \mathsf{cert}_i) = \top$ except with negligible probability. Hence $\mathcal{R}_{\mathsf{del}}((1, \mathsf{vk}_i), \mathsf{cert}_i) = 1$ for every $i \in D$ except with negligible probability.

For every $i \notin D$, we have $d_i = 0$, and therefore $\mathcal{R}_{\mathsf{del}}((0, \mathsf{vk}_i), 0^{\ell_{\mathsf{cert}}}) = 1$ by definition of the relation. Since N is polynomially bounded, a union bound implies that all N statements have valid witnesses except with negligible probability. Completeness of $\prod_{\mathsf{seBARG}}$ therefore gives $\mathsf{seBARG.V}(\mathsf{crs}, x_1, \ldots, x_N, \pi_D) = 1$ except with negligible probability. Thus $\mathsf{VrfySet}$ outputs $\top$.

For preservation, note that the only steps of DelSet that act on quantum data are the calls $\mathsf{BIBE\text{-}CD.Del}(\mathsf{ct}_i)$ for $i \in D$; the seBARG prover reads only the classical statements and witnesses. Hence the ciphertexts ct_i with $i \notin D$ are returned in their original state, and decryption correctness of $\prod^{\mathsf{CD}}_{\mathsf{BIBE}}$ gives the required bound.

5.5 Security

Theorem 5 (Selective Security). *Suppose $\prod^{\mathsf{CD}}_{\mathsf{BIBE}}$ is the publicly verifiable BIBE-CD scheme constructed in Appendix A, instantiated with a post-quantum selectively secure classical BIBE and a signature scheme satisfying one-time unforgeability for BB84 states, so that it satisfies selective semantic security (Lemma 4) and Definition 1 (Theorem 8), and suppose that $\prod_{\mathsf{seBARG}}$ satisfies post-quantum index hiding and the somewhere argument of knowledge property against QPT adversaries, with straight-line extraction. Then the construction in Section 5.3 satisfies post-quantum selective subset-deletion security.*

Proof. Fix a QPT adversary $\mathcal{A}$ and selectively committed challenge identity-label pairs $((\mathsf{id}^*_i, \mathsf{tg}^*_i))_{i\in[N]}$. Let $X^*_i = (\mathsf{id}^*_i, \mathsf{tg}^*_i)$ for every $i \in [N]$. We write $\mathbf{m}_0$ and $\mathbf{m}_1$ for the challenge message vectors that the adversary submits in the challenge phase, and $\Delta = \{i \in [N] : m_{0,i} \neq m_{1,i}\}$. Both are determined only once the challenge phase is reached. Define hybrids $\mathsf{Hyb}_0, \ldots, \mathsf{Hyb}_N$. In Hyb_j, for every $i \leq j$, the challenge ciphertext at position i encrypts $m_{1,i}$, while for every $i > j$ it encrypts $m_{0,i}$. Thus Hyb_0 is the experiment with challenge vector $\mathbf{m}_0$, while Hyb_N is the experiment with challenge vector $\mathbf{m}_1$. Fix $j \in [N]$. We compare Hyb_{j-1} and Hyb_j. Note that on every execution in which the adversary submits challenge vectors with $m_{0,j} = m_{1,j}$, the two hybrids are identically

distributed, since they differ only in the message encrypted at position j. Such executions therefore contribute equally to the two hybrid probabilities and may be disregarded when bounding their difference. On the remaining executions we have $m_{0,j} \neq m_{1,j}$ and hence $j \in \Delta$.

Queries authorizing the j-th challenge attribute. Let $\mathsf{BadAuth}_j$ be the event that the adversary issues, either before or after the challenge, a batch-key query (S, tg) satisfying $\mathsf{tg} = \mathsf{tg}_j^*$ and $\mathsf{id}_j^* \in S$. Equivalently, $P_{S,\mathsf{tg}}(X_j^*) = 1$. Executions on which $\mathsf{BadAuth}_j$ occurs split into two cases. If the challenge vectors eventually satisfy $m_{0,j} \neq m_{1,j}$, then $j \in \Delta$ and the global admissibility condition makes both adjacent hybrids output 0 on such an execution. If instead $m_{0,j} = m_{1,j}$, the two adjacent hybrids have identical continuations. To see this precisely, partition the probability space according to the transcript prefix p consisting of everything up to and including the submission of $(\mathbf{m}_0, \mathbf{m}_1)$. No challenge ciphertext has been produced at that point, so p is distributed identically in Hyb_{j-1} and in Hyb_j. Conditioned on any prefix p with $m_{0,j} = m_{1,j}$, the two hybrids encrypt the same message at every position, so their continuations are identically distributed. Combining the two cases over the partition gives

$$\Pr[\mathsf{Hyb}_{j-1} = 1 \ \wedge \ \mathsf{BadAuth}_j] = \Pr[\mathsf{Hyb}_j = 1 \ \wedge \ \mathsf{BadAuth}_j].$$

Consequently, a reduction that, as soon as $\mathsf{BadAuth}_j$ occurs, stops forwarding queries, completes its interaction with its own challenger trivially (submitting equal dummy challenge messages if the challenge phase has not yet been reached, and the invalid certificate $\perp$ wherever a certificate is expected), and contributes the output 0, changes each hybrid probability by the same amount and leaves their difference unchanged. It therefore suffices to compare the adjacent hybrids on the complementary event $\neg\mathsf{BadAuth}_j$.

Changing the seBARG extraction position. In the real setup, the seBARG reference string is generated at the fixed position 1. Define $\widehat{\mathsf{Hyb}}_{j-1}$ and $\widehat{\mathsf{Hyb}}_j$ exactly as the corresponding adjacent hybrids except that $(\mathsf{crs}_j, \mathsf{td}_j) \leftarrow \mathsf{seBARG.Gen}(1^\lambda, N, 1^{\ell_x}, 1^{\ell_{\mathsf{cert}}}, j)$ is used in place of the reference string generated at position 1. The trapdoor td_j is retained by the challenger and is not given to the adversary.
By index hiding of $\prod_{\mathsf{seBARG}}$, $\mathsf{Hyb}_{j-1} \approx_c \widehat{\mathsf{Hyb}}_{j-1}$ and $\widehat{\mathsf{Hyb}}_j \approx_c \mathsf{Hyb}_j$. These are transitions between complete experiment outputs. A distinguisher that receives either reference-string distribution can run the remainder of the subset-deletion experiment by itself, including setup of the certified-deletion scheme, all batch-key queries, the challenge phase, subset verification, and the master-secret-key release branch. Ordinary computational indistinguishability therefore applies.

Reduction to single-ciphertext certified deletion and semantic security. We next show $\widehat{\mathsf{Hyb}}_{j-1} \approx_c \widehat{\mathsf{Hyb}}_j$. We use two reductions with a common simulation: a reduction $\mathcal{B}_{\mathrm{sem}}$ to the selective semantic security of Lemma 4, which accounts for the executions in which the aggregate proof is rejected, and a reduction $\mathcal{B}_{\mathrm{cd}}$ to the certified-everlasting experiment of Definition 1, which accounts for those

in which it is accepted. On rejection, the latter experiment outputs its own rejection symbol $\perp$ rather than the adversary's final bit, so it cannot by itself bound the rejecting contribution. Both reductions select the target challenge attribute $X_j^* = (\mathsf{id}_j^*, \mathsf{tg}_j^*)$.

Each reduction receives the public parameters mpk_0 of the BIBE-CD scheme from its own challenger. The two challengers have the same key-query and challenge interfaces up to the point at which a deletion certificate is submitted. It computes $(\mathsf{crs}_j, \mathsf{td}_j) \leftarrow \mathsf{seBARG.Gen}(1^\lambda, N, 1^{\ell_x}, 1^{\ell_{\mathsf{cert}}}, j)$, sets $\mathsf{mpk} = (\mathsf{mpk}_0, \mathsf{crs}_j)$, and gives mpk to $\mathcal{A}$.

Whenever $\mathcal{A}$ submits a batch-key query (S, tg), the reduction records the query and checks whether $P_{S,\mathsf{tg}}(X_j^*) = 1$. If so, it enters the $\mathsf{BadAuth}_j$ branch and outputs the fixed bit 0, completing its interaction with its challenger trivially as described above; for $\mathcal{B}_{\mathrm{cd}}$, the submitted certificate $\perp$ yields the rejection outcome, which is mapped to 0 below. Otherwise, the reduction forwards (S, tg) to its challenger and returns the resulting batch key to $\mathcal{A}$. Every forwarded query is therefore admissible in the underlying selective experiment.

When $\mathcal{A}$ submits $\mathbf{m}_0$ and $\mathbf{m}_1$, the reduction first checks whether $m_{0,j} = m_{1,j}$. If so, it submits $(m_{0,j}, m_{0,j})$ to its challenger, completes the interaction with the certificate $\perp$ wherever one is expected, and outputs 0. As shown above, such executions contribute equally to the two adjacent hybrids, so we henceforth restrict attention to executions with $j \in \Delta$. Otherwise, the reduction submits $(m_{0,j}, m_{1,j})$ with challenge attribute X_j^* to its BIBE-CD challenger. It receives the challenge pair $(\mathsf{ct}_j^*, \mathsf{vk}_j^*)$. For every $i < j$, the reduction computes $(\mathsf{ct}_i^*, \mathsf{vk}_i^*) \leftarrow \mathsf{BIBE\text{-}CD.Enc}(\mathsf{mpk}_0,\ m_{1,i}, \mathsf{id}_i^*, \mathsf{tg}_i^*)$. For every $i > j$, it computes $(\mathsf{ct}_i^*, \mathsf{vk}_i^*) \leftarrow \mathsf{BIBE\text{-}CD.Enc}(\mathsf{mpk}_0,\ m_{0,i},\ \mathsf{id}_i^*,\ \mathsf{tg}_i^*)$. The reduction sends the complete challenge collection $((\mathsf{ct}_i^*, \mathsf{vk}_i^*))_{i \in [N]}$ to $\mathcal{A}$. Post-challenge batch-key queries are answered exactly as before. Eventually, $\mathcal{A}$ outputs a subset $D \subseteq [N]$ and a purported aggregate certificate $\mathsf{cert}_D = \pi_D$.

Enforcing global admissibility. Once the challenge message vectors have been submitted, the set Δ is determined, and the reduction knows every challenge attribute X_i^* because these were committed selectively before setup. Before producing its final output, the reduction therefore inspects its recorded query transcript and checks whether some batch-key query (S, tg) issued in either query phase satisfies $P_{S,\mathsf{tg}}(X_i^*) = 1$ for some $i \in \Delta$. If such a query exists, the target subset-deletion experiment outputs 0 by the global admissibility condition, and the reduction outputs 0 as well. This check is performed locally and does not alter which queries are forwarded to the underlying challenger, so $\mathcal{B}_{\mathrm{sem}}$ and $\mathcal{B}_{\mathrm{cd}}$ remain admissible in their respective experiments. Without this step the reduction would simulate an experiment in which admissibility is enforced only at the single position j, and the resulting discrepancy need not cancel between the two adjacent hybrids, because a violation at a position $i \in \Delta$ with $i \neq j$ may be triggered by a post-challenge query whose distribution depends on the j-th challenge ciphertext.

Subset inadmissibility. The reduction computes $\Delta = \{i \in [N] : m_{0,i} \neq m_{1,i}\}$. If $\Delta \not\subseteq D$, the target subset-deletion experiment outputs 0. The reduction therefore outputs 0 as well. The semantic-security reduction simply outputs 0. The certified-deletion reduction submits the invalid certificate $\perp$, whose rejection outcome is mapped to 0 below. Thus this branch contributes 0 in both hybrids. From now on assume $\Delta \subseteq D$. Since $j \in \Delta$, we have $j \in D$.

Decomposing acceptance and rejection. For $b \in \{0,1\}$, write $\widehat{\mathsf{Hyb}}^{(b)}$ for $\widehat{\mathsf{Hyb}}_{j-1}$ if $b = 0$ and for $\widehat{\mathsf{Hyb}}_j$ if $b = 1$; the two differ only in that position j encrypts $m_{b,j}$. For every $i \in [N]$, set $d_i = 1$ if $i \in D$ and $d_i = 0$ otherwise, let $x_i = (d_i, \mathsf{vk}_i^*)$, and let $\mu = \mathsf{seBARG.V}(\mathsf{crs}_j, x_1, \ldots, x_N, \pi_D)$. Let p_b^{rej} (resp. p_b^{acc}) be the probability that $\widehat{\mathsf{Hyb}}^{(b)}$ outputs 1 and, in addition, $j \in \Delta$, $\neg\mathsf{BadAuth}_j$, and $\mu = 0$ (resp. $\mu = 1$). These are joint probabilities, not probabilities conditioned on rejection or acceptance. Executions with $j \notin \Delta$ or with $\mathsf{BadAuth}_j$ contribute equally to both hybrids, as shown above. Hence

$$\Pr[\widehat{\mathsf{Hyb}}_{j-1} = 1] - \Pr[\widehat{\mathsf{Hyb}}_j = 1] = (p_0^{\mathrm{rej}} - p_1^{\mathrm{rej}}) + (p_0^{\mathrm{acc}} - p_1^{\mathrm{acc}}).$$

Rejecting aggregate proofs: semantic-security reduction. Definition 1 cannot be used on this branch: it terminates with its own output $\perp$ when the individual certificate is rejected, and therefore does not retain the final guess of $\mathcal{A}$. Instead, $\mathcal{B}_{\mathrm{sem}}$ runs the common simulation above in the selective semantic-security experiment of Lemma 4, obtaining the challenge ciphertext and verification key at position j from its challenger and generating all other positions itself. After $\mathcal{A}$ outputs (D, π_D), the reduction computes μ itself, which is possible because aggregate verification is public and uses only crs_j, the verification keys, and D. If $\mu = 0$, it sends $\perp$ to $\mathcal{A}$, without requesting or learning the master secret key, receives the final bit b' of $\mathcal{A}$, and outputs b' unless the admissibility checks above require output 0. If $\mu = 1$, it outputs 0 immediately. Hence $\mathcal{B}_{\mathrm{sem}}$ outputs 1 with probability exactly p_b^{rej} when its challenge bit is b. All of its forwarded key queries are admissible for X_j^*, because it stops forwarding on $\mathsf{BadAuth}_j$. Since Lemma 4 holds against QPT adversaries that also receive the public verification key,

$$|p_0^{\mathrm{rej}} - p_1^{\mathrm{rej}}| \leq \mathsf{negl}(\lambda).$$

This argument does not condition on the event $\mu = 0$; the semantic-security adversary computes that event itself.

Extraction on the accepting branch: certified-deletion reduction. Suppose $\mathsf{seBARG.V}(\mathsf{crs}_j, x_1, \ldots, x_N, \pi_D) = 1$. The reduction $\mathcal{B}_{\mathrm{cd}}$ computes $w_j \leftarrow \mathsf{seBARG.E}(\mathsf{td}_j, (x_i)_{i\in[N]}, \pi_D)$. The certified-deletion challenger, the reduction's own simulation, and $\mathcal{A}$ together form a single QPT algorithm that takes crs_j as input and outputs $(x_1, \ldots, x_N, \pi_D)$. The somewhere argument of knowledge property of Appendix 2.5 therefore applies to it. By the somewhere argument of knowledge property, except with negligible probability, $\mathcal{R}_{\mathsf{del}}(x_j, w_j) = 1$.

Since $j \in D$, we have $d_j = 1$ and therefore $x_j = (1, \mathsf{vk}_j^*)$. By definition of $\mathcal{R}_{\mathsf{del}}$, it follows that $\mathsf{BIBE\text{-}CD.Vrfy}(\mathsf{vk}_j^*, w_j) = \top$.
If extraction nevertheless returns w_j with $\mathcal{R}_{\mathsf{del}}(x_j, w_j) = 0$, the reduction submits $\perp$, which yields the rejection outcome of Definition 1. By the somewhere argument of knowledge property, this event has negligible probability for either challenge bit, and since extraction is straight-line it does not disturb the state of $\mathcal{A}$. Thus w_j is a valid individual deletion certificate for the j-th challenge ciphertext. The reduction submits w_j to its certified-deletion challenger. Since $\mathsf{BIBE\text{-}CD.Vrfy}(\mathsf{vk}_j^*, w_j) = \top$, in the experiment of Definition 1 the challenger returns the complete master secret key msk_0. The reduction forwards msk_0 to $\mathcal{A}$. Since the seBARG setup contributes only the public reference string, and the trapdoor td_j is never part of the master secret key, the master secret key of the final scheme is exactly msk_0. The reduction outputs, as its residual state, a classical register holding the final bit of $\mathcal{A}$, or the bit 0 if the admissibility checks above require it.
On executions with $\mu = 0$, $\mathcal{B}_{\mathrm{cd}}$ submits $\perp$, and its experiment ends with the rejection outcome $\perp$. The final guess of $\mathcal{A}$ is not represented there, which is why this branch was handled by $\mathcal{B}_{\mathrm{sem}}$. Let ρ_b denote the output of the experiment of Definition 1 run with $\mathcal{B}_{\mathrm{cd}}$ and challenge bit b, where the rejection outcome $\perp$ is a flag orthogonal to the accepting residual states. Measuring the bit register and mapping the flag $\perp$ to 0 is a fixed quantum channel. Except on the negligible extraction-failure event, this channel outputs 1 with probability exactly p_b^{acc}. Since trace distance does not increase under quantum channels, Theorem 8 gives

$$|p_0^{\mathrm{acc}} - p_1^{\mathrm{acc}}| \le \mathsf{TD}(\rho_0, \rho_1) + \mathsf{negl}(\lambda) \le \mathsf{negl}(\lambda).$$

This bound concerns the joint accepting probabilities and the full experiment output, including the rejection flag; it neither normalizes by nor conditions on successful verification.
Combining the two branches,

$$\left|\Pr[\widehat{\mathsf{Hyb}}_{j-1} = 1] - \Pr[\widehat{\mathsf{Hyb}}_j = 1]\right| \le |p_0^{\mathrm{rej}} - p_1^{\mathrm{rej}}| + |p_0^{\mathrm{acc}} - p_1^{\mathrm{acc}}| \le \mathsf{negl}(\lambda).$$

Together with the index-hiding transitions, this gives $\mathsf{Hyb}_{j-1} \approx_c \mathsf{Hyb}_j$. Applying the argument for every $j \in [N]$ and using that N is polynomially bounded yields $\mathsf{Hyb}_0 \approx_c \mathsf{Hyb}_N$. This proves selective subset-deletion security.

5.6 Succinctness

Theorem 6 (Succinctness). *Suppose the underlying classical* $\prod_{\mathsf{BIBE}}$ *is succinct and* $\prod_{\mathsf{seBARG}}$ *has proof length* $\mathsf{poly}(\lambda, \log N, |\mathcal{C}_{\mathcal{R}}|)$. *If the public verification keys and individual deletion certificates have length* $\mathsf{poly}(\lambda)$*, then the construction in Section 5.3 has a subset-deletion certificate of size* $\mathsf{poly}(\lambda, \log N)$*, independent of* $|D|$.

Proof. The aggregate subset-deletion certificate consists solely of the seBARG proof $\mathsf{cert}_D = \pi_D$. Each seBARG statement is of the form $x_i = (d_i, \mathsf{vk}_i)$ and

therefore has length $\ell_x = \mathsf{poly}(\lambda)$. Each witness is either an individual deletion certificate or a fixed dummy string and therefore has length $\ell_{\mathsf{cert}} = \mathsf{poly}(\lambda)$. The circuit computing $\mathcal{R}_{\mathsf{del}}$ performs one invocation of $\mathsf{BIBE\text{-}CD.Vrfy}$ on inputs of these lengths, so $|\mathcal{C}_{\mathcal{R}_{\mathsf{del}}}| = \mathsf{poly}(\lambda)$. By seBARG efficiency, $|\mathsf{cert}_D| = |\pi_D| = \mathsf{poly}(\lambda, \log N, |\mathcal{C}_{\mathcal{R}_{\mathsf{del}}}|) = \mathsf{poly}(\lambda, \log N)$. In particular, the certificate size is independent of the number $|D|$ of deleted ciphertexts. The identity digest and batch secret key are inherited from the underlying BIBE. Hence their sizes retain the succinctness guarantees of that scheme, up to the $\mathsf{poly}(\lambda)$ overhead introduced by the certified-deletion compiler.

Verification and proving cost. The subset-deletion certificate is succinct, but $\mathsf{VrfySet}$ is given the complete public verification-key vector $(\mathsf{vk}_i)_{i\in[N]}$ and must construct the N public statements $x_1, \ldots, x_N$. Thus the certificate size is $\mathsf{poly}(\lambda, \log N)$, while the verifier input and verification time are in general linear in N. More precisely, the verifier input consists of mpk, the N verification keys of $\mathsf{poly}(\lambda)$ bits each, and a description of D, using N membership bits or $O(|D| \log N)$ bits in sparse form, and $\mathsf{VrfySet}$ runs in time $\mathsf{poly}(\lambda, N)$. Honest subset deletion performs $|D|$ individual deletion measurements followed by one classical seBARG proof over N statements, in time $\mathsf{poly}(\lambda, N)$. The $N - |D|$ undeleted ciphertexts remain with the holder and are not sent to the verifier. Our succinctness claim concerns the size of the public subset-deletion certificate. Compressing the verification-key vector into a succinct public commitment and proving consistency with it inside the aggregate relation would be needed to obtain sublinear public verification, and we leave this to future work.

Corollary 1. *Instantiating the underlying classical BIBE with the construction of Section 4, applying the publicly verifiable certified-deletion compiler of Appendix A.2, and using a post-quantum secure seBARG satisfying all properties of Appendix 2.5 against QPT adversaries gives a post-quantum selectively secure batched identity-based encryption scheme with succinct publicly verifiable subset deletion. The assumptions required by each component, and candidate instantiations, are discussed in Section 3.4.*

6 Conclusion

We introduced batched identity-based encryption with succinct publicly verifiable subset deletion. The construction achieves succinctness both in batching, through a compact identity digest and a single batch secret key, and in deletion, by aggregating individual deletion certificates into one publicly verifiable certificate of size $\mathsf{poly}(\lambda, \log N)$, independent of the number of deleted ciphertexts. We first build a generic post quantum BIBE, apply a publicly verifiable certified-deletion compiler, and then use a somewhere-extractable batch argument to aggregate deletion certificates. Extending the constructions to achieve adaptive security via complexity leveraging [6] is a natural direction for future work. The construction is generic, and its security is conditional on the post-quantum primitives discussed in Section 3.4; we do not give a concrete parameterized instantiation. Our subset-deletion guarantee is computational, whereas the underlying

single-ciphertext scheme is certified-everlasting, and the aggregate certificate is succinct while the verifier input and verification time remain linear in N. Obtaining everlasting security for subset deletion, and sublinear public verification by committing to the verification-key vector, are further directions for future work.

Acknowledgments. This work is supported in part by the Anusandhan National Research Foundation (ANRF), Department of Science and Technology (DST), Government of India, under File No. EEQ/2023/000164, Defence Research and Development Organisation, Government of India, under File No. SAG/4600/TCID/CARS/C-90 and the Information Security Education and Awareness (ISEA) Project Phase-III initiatives of the Ministry of Electronics and Information Technology (MeitY) under Grant No. F.No. L-14017/1/2022-HRD.

References

1. Agarwal, A., Fernando, R., Pinkas, B.: Efficiently-thresholdizable batched identity based encryption, with applications. In: Annual International Cryptology Conference. pp. 69–100. Springer (2025)
2. Bartusek, J., Goyal, V., Khurana, D., Malavolta, G., Raizes, J., Roberts, B.: Software with certified deletion. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 85–111. Springer (2024)
3. Bartusek, J., Khurana, D.: Cryptography with certified deletion. In: Annual International Cryptology Conference. pp. 192–223. Springer (2023)
4. Bartusek, J., Khurana, D., Malavolta, G., Poremba, A., Walter, M.: Weakening assumptions for publicly-verifiable deletion. In: Theory of Cryptography Conference. pp. 183–197. Springer (2023)
5. Bartusek, J., Khurana, D., Poremba, A.: Publicly-verifiable deletion via target-collapsing functions. In: Annual International Cryptology Conference. pp. 99–128. Springer (2023)
6. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Annual International Cryptology Conference. pp. 443–459. Springer (2004)
7. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Annual international cryptology conference. pp. 213–229. Springer (2001)
8. Boneh, D., Laufer, E., Tas, E.N.: Batch decryption without epochs and its application to encrypted mempools. Cryptology ePrint Archive (2025)
9. Brakerski, Z., Döttling, N., Garg, S., Malavolta, G.: Factoring and pairings are not necessary for io: Circular-secure lwe suffices. Cryptology ePrint Archive (2020)
10. Broadbent, A., Islam, R.: Quantum encryption with certified deletion. In: Theory of Cryptography Conference. pp. 92–122. Springer (2020)
11. Çakan, A., Goyal, V.: Unclonable cryptography with unbounded collusions and impossibility of hyperefficient shadow tomography. In: Theory of Cryptography Conference. pp. 225–256. Springer (2024)
12. Choudhuri, A.R., Jain, A., Jin, Z.: Snargs for {P} from lwe. In: 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS). pp. 68–79. IEEE (2022)
13. Choudhuri, A.R., Jain, A., Jin, Z.: Non-interactive batch arguments for np from standard assumptions. In: Annual International Cryptology Conference. pp. 394–423. Springer (2021)

14. Cini, V., Lai, R.W., Woo, I.K.: Lattice-based obfuscation from NTRU and Equivocal LWE. Cryptology ePrint Archive (2025), https://link.springer.com/chapter/10.1007/978-3-032-01907-3_2
15. Devadas, L., Goyal, R., Kalai, Y., Vaikuntanathan, V.: Rate-1 non-interactive arguments for batch-np and applications. In: 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS). pp. 1057–1068. IEEE (2022)
16. Gay, R., Pass, R.: Indistinguishability obfuscation from circular security. In: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing. pp. 736–749 (2021)
17. Hiroka, T., Morimae, T., Nishimaki, R., Yamakawa, T.: Quantum encryption with certified deletion, revisited: Public key, attribute-based, and classical communication. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 606–636. Springer (2021)
18. Kitagawa, F., Nishimaki, R., Yamakawa, T.: Publicly verifiable deletion from minimal assumptions. In: Theory of Cryptography Conference. pp. 228–245. Springer (2023)
19. Koppula, V., Lewko, A.B., Waters, B.: Indistinguishability obfuscation for turing machines with unbounded memory. In: Proceedings of the forty-seventh annual ACM symposium on Theory of Computing. pp. 419–428 (2015)
20. Okamoto, T., Pietrzak, K., Waters, B., Wichs, D.: New realizations of somewhere statistically binding hashing and positional accumulators. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 121–145. Springer (2015)
21. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Workshop on the theory and application of cryptographic techniques. pp. 47–53. Springer (1984)
22. Wee, H., Wichs, D.: Candidate obfuscation via oblivious lwe sampling. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 127–156. Springer (2021)
23. Xiong, S., Zhang, Y., Chen, J.: Efficient batched ibe from lattices. Cryptology ePrint Archive (2025)

Appendix

A Instantiation of BIBE-CD with Publicly Verifiable Certified-Everlasting Deletion

In this section, we show how to equip a classical BIBE with publicly verifiable certified-everlasting deletion. Our construction is a direct specialization of the publicly verifiable deletion compiler of Kitagawa, Nishimaki, and Yamakawa [18] to the BIBE functionality. In particular, their generic result applies to attribute-based encryption and therefore also to the restricted predicate family naturally induced by a BIBE.

For simplicity, we first describe the construction for the plaintext space $\{0,1\}$. As in [18], the construction extends to plaintexts of polynomial length by a standard hybrid argument. Concretely, for plaintexts of length $\ell = \mathsf{poly}(\lambda)$ the encryption algorithm samples $x, \theta \leftarrow \{0,1\}^{\lambda\ell}$, uses one block of λ positions per message bit, and masks the t-th message bit with the parity of the θ-marked positions of the t-th block. A single signed state and a single public verification key cover the entire message, so deletion still produces one certificate per ciphertext. Accordingly, Lemma 4, Definition 1, and Theorem 8 are stated for $m \in \{0,1\}$, and Section 5 uses their ℓ-bit versions obtained in this way. The proof of Lemma 4 applies verbatim to ℓ-bit messages, with $\theta \in \{0,1\}^{\lambda\ell}$ and $\beta \in \{0,1\}^{\ell}$, since it never uses that the message is a single bit.

A.1 Syntax

Let $\mathcal{I}$ be an identity space and Γ be a batch-label space. A batched identity-based encryption scheme with publicly verifiable certified deletion is a tuple $\prod^{\mathsf{CD}}_{\mathsf{BIBE}} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Digest}, \mathsf{ComputeKey}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ with the following algorithms.

- $\mathsf{Setup}(1^\lambda, 1^B) \to (\mathsf{mpk}, \mathsf{msk})$: On input the security parameter and the maximum batch size B, outputs a master public key mpk and a master secret key msk.
- $\mathsf{Enc}(\mathsf{mpk}, m, \mathsf{id}, \mathsf{tg}) \to (\mathsf{ct}, \mathsf{vk})$: On input the master public key, a message $m \in \{0,1\}$, an identity $\mathsf{id} \in \mathcal{I}$, and a batch label $\mathsf{tg} \in \Gamma$, outputs a possibly quantum ciphertext ct and a public verification key vk.
- $\mathsf{Digest}(\mathsf{mpk}, S) \to \mathsf{dig}$: On input the master public key and a set $S \subseteq \mathcal{I}$ with $|S| \leq B$, outputs a public digest dig.
- $\mathsf{ComputeKey}(\mathsf{msk}, \mathsf{dig}, \mathsf{tg}) \to \mathsf{sk}$: On input the master secret key, a digest dig, and a batch label tg, outputs a batch secret key sk.
- $\mathsf{Dec}(\mathsf{sk}, S, (\mathsf{id}, \mathsf{tg}), \mathsf{ct}) \to m/\bot$: On input a batch secret key, the corresponding set S, an identity-label pair $(\mathsf{id}, \mathsf{tg})$, and a ciphertext, outputs a message or $\bot$.

- $\mathsf{Del}(\mathsf{ct}) \to \mathsf{cert}$: On input a ciphertext, performs a destructive quantum measurement and outputs a classical deletion certificate cert.
- $\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert}) \to \top/\bot$: On input a public verification key and a deletion certificate, deterministically outputs $\top$ or $\bot$. We adopt the convention $\mathsf{Vrfy}(\mathsf{vk}, \bot) = \bot$.

A.2 Construction

Let $\prod_{\mathsf{BIBE}} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Digest}, \mathsf{ComputeKey}, \mathsf{Dec})$ be a post-quantum selectively secure classical BIBE (cf. Appendix 2.1). We require that its plaintext space contain $\{0,1\}^{|\mathsf{sigk}|+\lambda+1}$ (for ℓ-bit messages, $\{0,1\}^{|\mathsf{sigk}|+\lambda\ell+\ell}$), so that the classical payload $(\mathsf{sigk}, \theta, \beta)$ used below can be encrypted in a single ciphertext. Let $\prod_{\mathsf{SIG}} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a deterministic signature scheme satisfying one-time unforgeability for BB84 states in the sense of Kitagawa *et al.* [18]. Informally, this notion states that no efficient adversary given a coherently signed BB84 state can both output a valid signed computational-basis string and retain information about the Hadamard-marked positions of that state. It is the property that carries the deletion guarantee below, and we refer to [18] for its formal statement.

For $x, \theta \in \{0,1\}^\lambda$, write $|x\rangle_\theta = \bigotimes_{i=1}^{\lambda} H^{\theta_i} |x_i\rangle$. For a signing key sigk, let U_{sigk} denote the unitary implementation of deterministic signing defined by $U_{\mathsf{sigk}} |\mu\rangle |0\rangle = |\mu\rangle |\mathsf{SIG.Sign}(\mathsf{sigk}, \mu)\rangle$. The algorithms of $\prod_{\mathsf{BIBE}}^{\mathsf{CD}}$ are defined as follows.

$\mathsf{Setup}(1^\lambda, 1^B)$:
- Compute $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{BIBE.Setup}(1^\lambda, 1^B)$.
- Output $(\mathsf{mpk}, \mathsf{msk})$.

$\mathsf{Enc}(\mathsf{mpk}, m, \mathsf{id}, \mathsf{tg})$:
- Sample $x, \theta \leftarrow \{0,1\}^\lambda$.
- Generate $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$.
- Prepare the signed BB84 state $|\psi\rangle = U_{\mathsf{sigk}}(|x\rangle_\theta |0\rangle)$.
- Compute $\beta = m \oplus \bigoplus_{i:\theta_i=1} x_i$.
- Compute $\mathsf{ct}_0 \leftarrow \mathsf{BIBE.Enc}(\mathsf{mpk}, (\mathsf{sigk}, \theta, \beta), \mathsf{id}, \mathsf{tg})$.
- Set $\mathsf{ct} = (|\psi\rangle, \mathsf{ct}_0)$.
- Output $(\mathsf{ct}, \mathsf{vk})$.

$\mathsf{Digest}(\mathsf{mpk}, S)$:
- Compute and output $\mathsf{dig} \leftarrow \mathsf{BIBE.Digest}(\mathsf{mpk}, S)$.

$\mathsf{ComputeKey}(\mathsf{msk}, \mathsf{dig}, \mathsf{tg})$:
- Compute and output $\mathsf{sk} \leftarrow \mathsf{BIBE.ComputeKey}(\mathsf{msk}, \mathsf{dig}, \mathsf{tg})$.

$\mathsf{Dec}(\mathsf{sk}, S, (\mathsf{id}, \mathsf{tg}), \mathsf{ct})$:
- Parse $\mathsf{ct} = (\rho, \mathsf{ct}_0)$.

- Compute $(\mathsf{sigk}, \theta, \beta) \leftarrow \mathsf{BIBE.Dec}(\mathsf{sk}, S, (\mathsf{id}, \mathsf{tg}), \mathsf{ct}_0)$. If the output is $\perp$, output $\perp$.
- Using sigk, coherently erase the signature register of ρ by applying U_{sigk}^{-1}.
- Measure the first λ qubits of the resulting state in the Hadamard basis and denote the outcome by $\overline{x} \in \{0,1\}^\lambda$. Only the positions i with $\theta_i = 1$ are used below.
- Compute and output $m' = \beta \oplus \bigoplus_{i:\theta_i=1} \overline{x}_i$.

$\mathsf{Del}(\mathsf{ct})$:

- Parse $\mathsf{ct} = (\rho, \mathsf{ct}_0)$.
- Measure ρ in the computational basis, jointly on the message and signature registers, and obtain (x', σ').
- Output $\mathsf{cert} = (x', \sigma')$.

$\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert})$:

- If $\mathsf{cert} = \perp$, output $\perp$. Otherwise, parse $\mathsf{cert} = (x', \sigma')$.
- Compute $\mu \leftarrow \mathsf{SIG.Vrfy}(\mathsf{vk}, x', \sigma')$.
- Output $\top$ if $\mu = 1$. Otherwise, output $\perp$.

A.3 Correctness

Theorem 7 (Correctness). *If $\prod_{\mathsf{BIBE}}$ and $\prod_{\mathsf{SIG}}$ are correct, then $\prod_{\mathsf{BIBE}}^{\mathsf{CD}}$ satisfies decryption correctness and deletion correctness.*

Proof. We first consider decryption correctness. Fix $m \in \{0,1\}$, $\mathsf{id} \in S$, and $\mathsf{tg} \in \Gamma$. The classical component of the ciphertext is $\mathsf{ct}_0 \leftarrow \mathsf{BIBE.Enc}(\mathsf{mpk}, (\mathsf{sigk}, \theta, \beta), \mathsf{id}, \mathsf{tg})$, where $\beta = m \oplus \bigoplus_{i:\theta_i=1} x_i$. Correctness of the underlying BIBE implies that decryption recovers $(\mathsf{sigk}, \theta, \beta)$ except with negligible probability.
For an honestly generated ciphertext, the quantum component is $|\psi\rangle = U_{\mathsf{sigk}}(|x\rangle_\theta |0\rangle)$. Applying U_{sigk}^{-1} therefore recovers $|x\rangle_\theta |0\rangle$. For every position i satisfying $\theta_i = 1$, the i-th qubit of $|x\rangle_\theta$ is a Hadamard-basis encoding of x_i. Consequently, measurement in the Hadamard basis gives $\overline{x}_i = x_i$ with probability 1 for every such position. Positions with $\theta_i = 0$ yield uniformly random outcomes but do not enter the reconstruction. Hence $\beta \oplus \bigoplus_{i:\theta_i=1} \overline{x}_i = m$. This proves decryption correctness.
We next consider deletion correctness. The state $|\psi\rangle$ is obtained by coherently signing every computational-basis value in the BB84 state. Therefore, measuring $|\psi\rangle$ in the computational basis jointly on both registers produces some pair (x', σ') satisfying $\sigma' = \mathsf{SIG.Sign}(\mathsf{sigk}, x')$, since U_{sigk} maps each computational-basis vector $|\mu\rangle |0\rangle$ to $|\mu\rangle |\mathsf{SIG.Sign}(\mathsf{sigk}, \mu)\rangle$ and therefore supports only consistent pairs. By correctness of the signature scheme, $\mathsf{SIG.Vrfy}(\mathsf{vk}, x', \sigma') = 1$. Thus an honestly generated deletion certificate is accepted with probability 1.

A.4 Selective Semantic Security

Lemma 4. *If $\prod_{\mathsf{BIBE}}$ is post-quantum selectively secure, then $\prod^{\mathsf{CD}}_{\mathsf{BIBE}}$ is post-quantum selectively secure even when the public verification key vk is given together with the challenge ciphertext.*

Proof. Fix a selective challenge identity-label pair $X^* = (\mathsf{id}^*, \mathsf{tg}^*)$ and an admissible QPT adversary. Consider challenge messages $m_0, m_1 \in \{0,1\}$.
The reduction samples $x, \theta \leftarrow \{0,1\}^\lambda$ and $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$, and prepares $|\psi\rangle = U_{\mathsf{sigk}}(|x\rangle_\theta |0\rangle)$. For $b \in \{0,1\}$, define $\beta_b = m_b \oplus \bigoplus_{i:\theta_i=1} x_i$ and $M_b = (\mathsf{sigk}, \theta, \beta_b)$. The reduction submits (M_0, M_1) as its challenge messages in the selective security experiment of the underlying BIBE. The two payloads have equal length. The resulting challenge ciphertext ct^*_0 is distributed as an encryption of $(\mathsf{sigk}, \theta, \beta_b)$ under X^*. The reduction sends $((|\psi\rangle, \mathsf{ct}^*_0), \mathsf{vk})$ to the adversary. The verification key and quantum component are generated independently of the challenge bit once x, θ, and sigk have been fixed. All admissible batch-key queries are forwarded to the underlying BIBE challenger. Thus the adversary's view is distributed exactly as in the selective security experiment of $\prod^{\mathsf{CD}}_{\mathsf{BIBE}}$. Any non-negligible distinguishing advantage therefore contradicts selective security of $\prod_{\mathsf{BIBE}}$.

A.5 Publicly Verifiable Certified-Everlasting Deletion

Definition 1 (Selective Publicly Verifiable Certified-Everlasting Deletion). *For a bit $b \in \{0,1\}$, the experiment $\mathsf{Exp}^{\mathsf{cert\text{-}del}}_{\mathcal{A}}(\lambda, b)$ is defined as follows. The adversary commits to a challenge identity-label pair $X^* = (\mathsf{id}^*, \mathsf{tg}^*)$ before setup. The challenger computes $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathit{Setup}(1^\lambda, 1^B)$ and sends mpk to the adversary. The adversary may adaptively obtain batch keys for pairs (S, tg) provided that $\mathsf{tg} \neq \mathsf{tg}^*$ or $\mathsf{id}^* \notin S$. It then submits two equal-length messages (m_0, m_1). The challenger computes $(\mathsf{ct}^*, \mathsf{vk}^*) \leftarrow \mathit{Enc}(\mathsf{mpk}, m_b, \mathsf{id}^*, \mathsf{tg}^*)$ and gives both ct^* and vk^* to the adversary. Further admissible batch-key queries are allowed. The adversary then outputs a deletion certificate cert^*. If $\mathit{Vrfy}(\mathsf{vk}^*, \mathsf{cert}^*) = \top$, the challenger sends the complete master secret key msk to the adversary, which then outputs a residual quantum state ρ, and the experiment outputs ρ. Otherwise the experiment outputs $\perp$. We say that $\prod^{\mathsf{CD}}_{\mathsf{BIBE}}$ satisfies selective publicly verifiable certified-everlasting deletion if*

$$\mathsf{TD}\left(\mathsf{Exp}^{\mathsf{cert\text{-}del}}_{\mathcal{A}}(\lambda, 0), \mathsf{Exp}^{\mathsf{cert\text{-}del}}_{\mathcal{A}}(\lambda, 1)\right) \leq \mathsf{negl}(\lambda)$$

for every QPT adversary $\mathcal{A}$.

Theorem 8 (Publicly Verifiable Certified-Everlasting Deletion). *Assume that $\prod_{\mathsf{BIBE}}$ is post-quantum selectively secure and $\prod_{\mathsf{SIG}}$ satisfies one-time unforgeability for BB84 states. Then $\prod^{\mathsf{CD}}_{\mathsf{BIBE}}$ satisfies Definition 1.*

Proof. The proof is a direct specialization of the publicly verifiable certified-everlasting lemma of Kitagawa, Nishimaki, and Yamakawa [18]. That lemma

states the following. Let E be a classical encryption functionality whose ciphertexts carry a public attribute and whose keys are associated with predicates over attributes. Suppose that, for every admissible adversary, the distribution of an encryption of $(\mathsf{sigk}, \theta, \beta)$ under the challenge attribute is computationally indistinguishable both from the distribution in which sigk is replaced by an all-zero string and from the distribution in which θ is replaced by 0^λ, where in both cases all admissible key queries continue to be answered honestly. Then wrapping E with a coherently signed BB84 state, as in Appendix A.2, yields a scheme with publicly verifiable certified-everlasting deletion, where the two complete experiment outputs, namely the residual state after master-key release on acceptance and the rejection flag $\perp$ otherwise, are close in trace distance. This is a statement about the full output, including the rejection outcome, and not about normalized states conditioned on acceptance.

Fix the selective challenge pair $X^* = (\mathsf{id}^*, \mathsf{tg}^*)$. The underlying classical BIBE may equivalently be viewed as a restricted key-policy attribute-based encryption scheme whose ciphertext attributes are pairs $X = (\mathsf{id}, \mathsf{tg})$ and whose predicates are $P_{S,\mathsf{tg}_0}(\mathsf{id}, \mathsf{tg}) = 1$ if and only if $\mathsf{id} \in S$ and $\mathsf{tg} = \mathsf{tg}_0$. The admissibility condition in the selective BIBE experiment is therefore exactly the usual attribute-based admissibility condition $P_{S,\mathsf{tg}}(X^*) = 0$.

To apply the publicly verifiable certified-everlasting lemma, consider the classical payload $(\mathsf{sigk}, \theta, \beta)$ encrypted by the underlying BIBE. The two computational indistinguishability conditions required by the lemma amount to replacing, inside the challenge plaintext, either sigk by a fixed all-zero string or θ by 0^λ, while retaining all admissible batch-key queries. Both transitions follow directly from selective security of the underlying BIBE (Appendix 2.1), by reductions of the same form as in the proof of Lemma 4, since the challenge attribute remains X^* and every queried batch key is unauthorized for X^*.

Therefore the hypotheses of the publicly verifiable certified-everlasting lemma are satisfied with the underlying BIBE playing the role of the classical encryption functionality. One-time unforgeability of $\prod_{\mathsf{SIG}}$ for BB84 states, together with these two transitions, implies negligible trace distance between the full experiment outputs, including the rejection flag, in the two challenge-message worlds. A claim about states conditioned on acceptance would require a lower bound on the acceptance probability and is not made here. Hence

$$\mathsf{TD}\left(\mathsf{Exp}_{\mathcal{A}}^{\mathsf{cert\text{-}del}}(\lambda, 0), \mathsf{Exp}_{\mathcal{A}}^{\mathsf{cert\text{-}del}}(\lambda, 1)\right) \leq \mathsf{negl}(\lambda).$$

This proves publicly verifiable certified-everlasting deletion.

Corollary 2 (Post-Deletion Master-Secret-Key Exposure). *The construction above satisfies the computational post-deletion master-secret-key exposure guarantee used in the main body. Formally,* msk *is released only if the public certificate is accepted, and otherwise the experiment outputs the rejection flag* $\perp$*. For every* QPT *adversary that outputs a bit* b' *after receiving* msk*, the probability of the joint event that the certificate is accepted and* $b' = 1$ *differs between the two challenge messages by at most a negligible amount. No claim is made about the conditional distribution given acceptance.*

Proof. By Definition 1, the master secret key msk is handed to the adversary before it outputs its residual state, so msk is already part of the adversary's view in the experiments compared by Theorem 8. Any efficient adversary that distinguishes the two challenge messages after receiving msk is in particular a distinguisher for the two residual states of those experiments, and hence has advantage at most $\mathsf{TD}(\mathsf{Exp}_{\mathcal{A}}^{\mathsf{cert\text{-}del}}(\lambda, 0), \mathsf{Exp}_{\mathcal{A}}^{\mathsf{cert\text{-}del}}(\lambda, 1)) \leq \mathsf{negl}(\lambda)$ by Theorem 8. Concretely, let the residual state be the classical register holding b'. Measuring the experiment output and mapping $\perp$ to 0 outputs 1 exactly on the joint event that the certificate is accepted and $b' = 1$, so the probabilities of this event in the two worlds differ by at most the above trace distance.

A.6 Efficiency

Theorem 9 (Efficiency). *Suppose the underlying $\prod_{\mathsf{BIBE}}$ has succinct digest and batch secret key. Then $\prod_{\mathsf{BIBE}}^{\mathsf{CD}}$ preserves these succinctness guarantees. Moreover, the public verification key and individual deletion certificate have size $\mathsf{poly}(\lambda)$ independent of the batch size B.*

Proof. The algorithms Digest and $\mathsf{ComputeKey}$ are unchanged from the underlying classical BIBE. Consequently, their output sizes and running times retain exactly the same dependence on the batch bound B. In particular, the two-stage key interface of the underlying scheme is preserved, so $\mathsf{ComputeKey}$ still receives only a digest and a batch label and its running time remains independent of B. The deletion wrapper adds one signed BB84 state of $\mathsf{poly}(\lambda)$ qubits and one classical encryption of $(\mathsf{sigk}, \theta, \beta)$. The public verification key is a signature verification key and the deletion certificate consists of one signed computational-basis outcome. Both therefore have length $\mathsf{poly}(\lambda)$ independent of B.