Registered Functional Encryption with Certified Everlasting Security

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. Registered functional encryption removes the trusted authority bottleneck of standard functional encryption by allowing users to register their own public keys, with a transparent aggregation procedure producing a compact master public key. On the other hand, certified everlasting deletion enables a recipient to produce a classical certificate guaranteeing that a ciphertext has been information-theoretically destroyed, even against computationally unbounded adversaries. In this work, we initiate the study of registered functional encryption with certified everlasting deletion (RFE-CED), combining both properties for the first time. We define the syntax and security of RFE-CED, capturing two complementary guarantees: everlasting privacy, which bounds the trace distance between residual adversarial states after certified deletion, and post-verification indistinguishability, which ensures the adversary cannot distinguish the encrypted message even after the deletion certificate is verified. To construct RFE-CED, we also introduce public-slot registered functional encryption (PS-RFE) as a building block, in which each registered function takes both a secret and an arbitrary public input. Combining PS-RFE with a pseudorandom and authenticated BB84 encodings, we obtain a secure slotted RFE-CED scheme with the help of an interactive version of certified everlasting lemma.

Keywords: Functional Encryption · Registered Cryptography · Certified Deletion · Everlasting Security

1 Introduction

Functional encryption (FE) [2] enables fine-grained access to encrypted data: given a functional key sk_f, a recipient learns only $f(m)$ from an encryption of a message m. This functionality makes FE a natural framework for computing on sensitive data while preserving privacy. However, standard FE relies on a trusted key-generation authority that derives all functional keys from a master secret key. Consequently, the authority can decrypt every ciphertext in the system, and compromising it compromises the privacy

of all encrypted data [11]. This key-escrow problem has long been a central obstacle to deploying function-based cryptography in practice.

Recent work addresses this issue through registered cryptography. Beginning with registration-based encryption [7] and extending to registered attribute-based encryption [11] and registered functional encryption [6], this line of research replaces the trusted authority with a transparent, secret-less key curator. In particular, users generate their own keys and register public information with the curator, which merely aggregates registrations into compact public parameters. As a result, key-escrow problem is eliminated while preserving the compactness and efficiency of the underlying system.

Removing the key-escrow problem, however, does not fully resolve the security concerns surrounding long-lived sensitive data. Medical records, genomic data, and other sensitive information often remain private for decades, whereas cryptographic security depends on computational assumptions that may eventually fail. An adversary can therefore store ciphertexts today and decrypt them in the future once advances in algorithms or quantum computation render those assumptions obsolete.

A promising approach to this problem comes from quantum cryptography. The notion of certified deletion, introduced by Broadbent *et al.* [4] and foreshadowed by Unruh's work on revocable timed-release encryption [15], allows a receiver to produce a classical certificate proving that the quantum information underlying a ciphertext has been destroyed. Subsequent works strengthened this guarantee to certified everlasting security [1, 10], ensuring that once deletion is successfully certified, the encrypted message remains hidden even from an adversary that later becomes computationally unbounded.

The goals of registered cryptography and certified everlasting security are naturally complementary. The former removes reliance on a trusted authority, while the latter protects against future advances in computation. Nevertheless, existing constructions achieve only one of these goals at a time. Certified everlasting FE schemes [10] continue to rely on a master secret key, whereas registered FE [6, 14] provides no mechanism for ciphertext deletion with everlasting security. Certified deletion has recently been incorporated into registered ABE [12]. In a related line of work, Sarkar [14] introduced a non-committing variant of registered functional encryption. While the non-commitment version is a useful ingredient in existing approaches to certified deletion, such as that of Hiroka *et al.* [10], it does not suffice to achieve certified everlasting security in the registered functional encryption setting. Consequently, despite these developments, the problem of realizing certified everlasting deletion for registered functional encryption remains open. This gap motivates the central question addressed in our work.

Can a functional encryption scheme simultaneously eliminate key-escrow issue and support certified everlasting deletion of ciphertexts?

Answering this question requires reconciling two seemingly conflicting requirements. Any solution must preserve the compactness that makes the registered model attractive, while providing meaningful deletion guarantees in a setting where receivers generate their own keys and may later become corrupted.

Our Results. We answer the above question in the affirmative. We construct the first *registered functional encryption with certified everlasting deletion* (RFE-CED) scheme. Our construction preserves the compactness and succinctness of the registered model

while enabling receivers to certify the deletion of encrypted information. To formalize security in this setting, we introduce a notion of adaptive certified everlasting security for registered functional encryption. Our definition accounts for adaptive key generation and corruption, and guarantees that once deletion is successfully certified, the encrypted message remains hidden even against adversaries that later become computationally unbounded.

As a key technical ingredient, we introduce public-slot registered functional encryption (PS-RFE), a variant of registered FE that supports decryption with an auxiliary public input. It is the registered analogue of the FE with an additional public input to the function used by Hiroka *et al.* [10]. Using PS-RFE, BB84-style quantum encodings, and a length-doubling pseudorandom generator, we construct an RFE-CED scheme and prove its correctness, compactness, and certified everlasting security. Instantiating PS-RFE by our modification of the slotted registered FE of [6] (Section 4), we obtain the first escrow-free functional encryption scheme for all polynomial-size circuits with certified everlasting deletion, assuming an indistinguishability obfuscator secure against QPT adversaries, a post-quantum somewhere statistically binding hash function (e.g., from LWE), and post-quantum one-way functions (for the pseudorandom generator and the secret-key encryption scheme). We work in the slotted setting of [11], where all N registrations are fixed in advance and aggregated into the master public key. Accordingly, the construction of Section 5.3 (using Section 4) is technically a slotted RFE-CED, which we simply call RFE-CED. All our results (Theorems 2, 3 and 5) are stated and proved for slotted RFE-CED, and we leave dynamic registration to future work. We note that the slotted-to-registered transformation of [7, 11] (also used in [6]) does not only modify registration: a ciphertext of the resulting scheme consists of several slotted ciphertexts, one for each active slotted instance. Lifting certified deletion through this transformation therefore requires deleting every component ciphertext, accepting only if every component certificate verifies, and a composition argument for everlasting privacy.

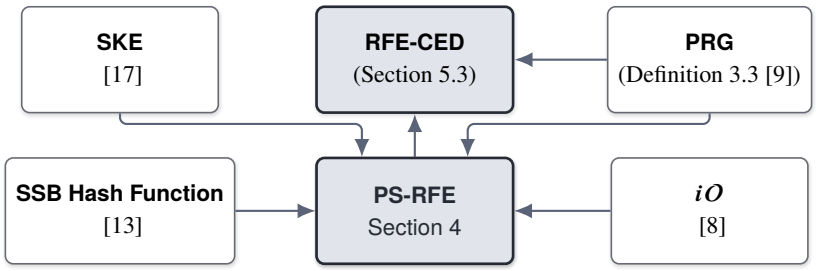

Fig. 1: High level overview of our construction

Our deletion layer follows Hiroka *et al.* [10]. We inherit the use of a public input to feed measurement outcomes into functional decryption, the PRG-authenticated BB84 encodings, the two-copy encoding of the challenge messages with a selector bit, and the overall structure of the deletion hybrids, which build on the certified everlasting lemma

of Bartusek and Khurana [1]. Our contribution is to carry this approach over to the registered setting: we define PS-RFE, construct it by adapting the obfuscation-based slotted registered FE of [6], and discharge the proof obligations that arise only in the presence of registration. Compared with registered ABE with certified deletion [12], our scheme supports arbitrary polynomial-size functions rather than predicates, but its deletion certificates are only privately verifiable, whereas [12] achieves public verifiability.

2 Technical Overview

This section presents a high-level overview of our construction of registered functional encryption with certified everlasting deletion (RFE-CED), depicted in Fig. 1.

Notations. The notation $x \leftarrow X$ denotes that x is selected uniformly at random from a finite set X. For a distribution D, we write $x \leftarrow D$ to denote sampling according to D. A function $f : \mathbb{N} \to \mathbb{R}$ is said to be negligible if, for every constant $c > 0$, there exists $\lambda_0 \in \mathbb{N}$ such that $f(\lambda) < \lambda^{-c}$ for all $\lambda > \lambda_0$. We use the notation $f(\lambda) \leq \mathsf{negl}(\lambda)$ to express that f is a negligible function. For a bit string x, we write x_j to mean the j-th bit of x. For bit strings $x, \theta \in \{0,1\}^\ell$, we write $|x\rangle_\theta$ to mean the BB84 state $\bigotimes_{j\in[\ell]} H^{\theta_j} |x_j\rangle$ where H is the Hadamard operator. The trace distance, denoted by $\mathsf{TD}(.,.)$, between two states ρ and σ is defined as $\mathsf{TD}(\rho,\sigma) = \|\rho - \sigma\|_{\mathrm{tr}}$, where $\|A\|_{\mathrm{tr}} := \mathrm{Tr}\left(\sqrt{A^\dagger A}\right)$. We use the symbol $\mathsf{Im}(f)$ to denote image of the function f.

Challenge. In a certified-deletion system, the information needed for decryption must be irreversibly destroyed once a valid deletion certificate is produced. In contrast, registered functional encryption must allow a registered receiver to compute $f(m)$ for an encrypted message m. The central challenge is therefore to enable functional evaluation while ensuring that the information required for decryption cannot survive a successful deletion procedure.

Public-Slot Registered FE. To address this issue, we introduce public-slot registered functional encryption (PS-RFE). Instead of evaluating functions of the form $f_i(m)$, decryption keys are associated with functions $g_i(m, \mathsf{pub})$, where pub denotes ciphertext-dependent information supplied only during decryption. We obtain PS-RFE by extending the slotted registered FE framework of [6] with an additional public-input wire while preserving the original registration procedures. This procedure allows decryption to depend on measurement outcomes without revealing the underlying message.

Message Encoding. For each message bit m_i, the encryptor samples random strings z_i and basis information θ_i, and computes

$$\beta_i = m_i \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}.$$

The values z_i are embedded using a BB84-style encoding determined by θ_i. In addition, for every encoded component, the encryptor samples seeds $u_{i,j,b}$ and publishes $v_{i,j,b} = \mathsf{PRG}(u_{i,j,b})$. The tuple (β, θ, v) is encrypted under PS-RFE, while the quantum encoding forms the remaining part of the ciphertext.

Functional Decryption. For each function f, we define an associated circuit $h[f]$. Upon receiving a ciphertext, the receiver measures the encoded registers and obtains

outcomes $(b_{i,j}, u_{i,j})$. These outcomes are supplied as the public input to PS-RFE. The circuit first verifies $\mathsf{PRG}(u_{i,j}) = v_{i,j,b_{i,j}}$, ensuring that the supplied values arise from a valid measurement. It then reconstructs

$$m_i = \beta_i \oplus \bigoplus_{j:\theta_{i,j}=0} b_{i,j}$$

and outputs $f(m)$. Thus, the plaintext is recovered only inside the protected functionality.

Two-Copy Structure. To facilitate the security proof, the ciphertext contains encodings of both challenge messages $m^{(0)}$ and $m^{(1)}$, together with a selector bit. The selector determines which message is evaluated during decryption. This structure enables a sequence of hybrid transitions that gradually replace one challenge message with the other while preserving functional equivalence on all corrupted slots, namely,

$$f_i(m^{(0)}) = f_i(m^{(1)}).$$

Deletion is performed by measuring all encoded registers in the Hadamard basis and producing a certificate consisting of the measurement outcomes. Verification checks consistency between the certificate, the secret basis information, and the published PRG values. Since the deletion measurement is incompatible with the information required to reconstruct the masks z_i, generating a valid certificate necessarily destroys the information needed for future recovery of the message.

Security Analysis. The proof combines the adaptive security of PS-RFE, the pseudorandomness of the PRG, and the interactive certified everlasting lemma. A sequence of hybrids gradually transforms an encryption of $m^{(0)}$ into an encryption of $m^{(1)}$. For each transition, indistinguishability follows from either PRG security or adaptive PS-RFE security on functionally equivalent plaintexts. The certified everlasting lemma then converts these computational guarantees into bounds on the trace distance of the post-deletion states. Consequently, the adversary's residual state after successful verification is statistically independent of the encrypted challenge message, establishing certified everlasting privacy. A final hybrid argument similarly yields post-verification indistinguishability.

3 Preliminaries

3.1 Indistinguishability Obfuscation

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

- **Correctness:** For every security parameter $\lambda \in \mathbb{N}$, every circuit $C \in C_\lambda$, and every input x, the obfuscated circuit preserves functionality:

$$\Pr[C'(x) = C(x) \mid C' \leftarrow i\mathcal{O}(C)] = 1.$$

- **Indistinguishability:** For any QPT distinguisher $\mathcal{D}$ and any pair of functionally equivalent circuits $C_0, C_1 \in C_\lambda$ such that $C_0(x) = C_1(x)$ for all inputs x and $|C_0| = |C_1|$, the obfuscations are computationally indistinguishable:

$$\left|\Pr[\mathcal{D}(i\mathcal{O}(C_0)) = 1] - \Pr[\mathcal{D}(i\mathcal{O}(C_1)) = 1]\right| \leq \mathsf{negl}(\lambda).$$

In recent literature, candidates for indistinguishability obfuscation secure against QPT adversaries have been proposed [3, 8, 16]. Moreover, the work of [5] proposes a post-quantum $i\mathcal{O}$ candidate based on the NTRU assumption and a novel Equivocal LWE problem, employing a trapdoor-based Gaussian hint-release mechanism that ensures correctness while preventing structural leakage, with security proven against QPT adversaries.

3.2 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 [13].

3.3 Pseudorandom Generator

A pseudorandom generator [9] $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell(\lambda)}$ with stretch $\ell(\lambda)$, for some polynomial ℓ, is a polynomial-time computable function satisfying the following. For any QPT adversary $\mathcal{A}$,

$$\left|\Pr\left[\mathcal{A}(\mathsf{PRG}(s)) = 1 : s \leftarrow \{0,1\}^\lambda\right] - \Pr\left[\mathcal{A}(r) = 1 : r \leftarrow \{0,1\}^{\lambda+\ell(\lambda)}\right]\right| \leq \mathsf{negl}(\lambda).$$

Lemma 1 (Interactive Certified Everlasting Lemma [1, 10]). *Let C be an efficient challenger, and for a QPT algorithm $\mathcal{B}$ let $\mathsf{Expt}_{\mathcal{B},C}(\lambda, \theta, \beta)$ be an interactive experiment*

between $\mathcal{B}$ and C, parameterised by $\theta \in \{0,1\}^\lambda$ and $\beta \in \{0,1\}$, at the end of which $\mathcal{B}$ outputs a bit. Suppose that for every QPT $\mathcal{B}$, every $\theta \in \{0,1\}^\lambda$, and every $\beta \in \{0,1\}$,

$$\left|\Pr[\mathsf{Expt}_{\mathcal{B},C}(\lambda,\theta,\beta)=1] - \Pr[\mathsf{Expt}_{\mathcal{B},C}(\lambda,0^\lambda,\beta)=1]\right| \le \mathsf{negl}(\lambda).$$

For a QPT algorithm $\mathcal{B}'$ and $b \in \{0,1\}$, let $\widetilde{\mathsf{Expt}}_{\mathcal{B}',C}(\lambda,b)$ be the experiment that samples $z, \theta \leftarrow \{0,1\}^\lambda$, runs $\mathcal{B}'(|z\rangle_\theta)$ in the role of $\mathcal{B}$ in $\mathsf{Expt}_{\mathcal{B},C}\big(\lambda,\theta,b\oplus\bigoplus_{j:\theta_j=0} z_j\big)$, and receives from $\mathcal{B}'$ a string $z' \in \{0,1\}^\lambda$ and a state ρ; it outputs ρ if $z'_j = z_j$ for every j with $\theta_j = 1$, and $\perp$ otherwise. Then, for every QPT $\mathcal{B}'$,

$$\mathsf{TD}\big(\widetilde{\mathsf{Expt}}_{\mathcal{B}',C}(\lambda,0), \widetilde{\mathsf{Expt}}_{\mathcal{B}',C}(\lambda,1)\big) \le \mathsf{negl}(\lambda).$$

We use the lemma in the form of [10, Lemma 3.5], which extends the certified everlasting lemma of [1] to experiments in which $\mathcal{B}$ interacts with a challenger.

4 Public-Slot Registered Functional Encryption

Let λ be the security parameter. Let $\mathcal{M}$ be the message space and $\mathcal{P}$ be the public input space. Let $\mathcal{U} = \{\mathcal{G}_\lambda\}_{\lambda\in\mathbb{N}}$ denote the universe of functions on $\mathcal{M}\times\mathcal{P}$. A Public-Slot Registered Functional Encryption scheme $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ = (Setup, KeyGen, IsValid, Agg, Enc, Dec) is a tuple of efficient algorithms with the following semantics:

$\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N)$

$(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, i)$

$b \in \{0,1\} \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i)$

$(\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[N]}) \leftarrow \mathsf{Agg}(\mathsf{crs}, \{(\mathsf{pk}_i, g_i)\}_{i\in[N]})$

$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, m)$

$y/\perp \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{hsk}, \mathsf{ct}, \mathsf{pub})$

where

crs :	common reference string
$N \in \mathbb{N}$:	number of slots
$i \in [N]$:	slot index
$g_i \in \mathcal{G}_\lambda$:	two-input function for slot i
$\mathsf{pk}_i, \mathsf{sk}_i$:	public/secret key pair for slot i
mpk :	master public key
hsk_i :	helper secret key for slot i
$m \in \mathcal{M}$:	message
ct :	ciphertext
$\mathsf{pub} \in \mathcal{P}$:	public input
y :	decrypted output

4.1 Correctness and Efficiency of PS-RFE

Let $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ denote a public-slot registered functional encryption scheme defined over $\mathcal{M}$, $\mathcal{P}$, and $\mathcal{U}$. We say $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ is *correct and efficient* if the following conditions hold.

- **Completeness.** For all $\lambda, N \in \mathbb{N}$ and all $i \in [N]$, $\Pr[\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i) = 1] = 1$, where $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N)$ and $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, i)$.

- **Correctness.** For all $\lambda, N \in \mathbb{N}$, all $i \in [N]$, all $g_i \in \mathcal{G}_\lambda$, all $\{(j, g_j, \mathsf{pk}_j)\}_{j \neq i}$ with $\mathsf{IsValid}(\mathsf{crs}, j, \mathsf{pk}_j) = 1$, and all $m \in \mathcal{M}$, where crs ranges over the support of $\mathsf{Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N)$, writing $\mathbf{pk} = \{(\mathsf{pk}_j, g_j)\}_{j \in [N]}$,

$$\Pr\left[\begin{array}{l|l} \forall\, \mathsf{pub} \in \mathcal{P} : & (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, i) \\ \mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}, \mathsf{pub}) & (\mathsf{mpk}, \{\mathsf{hsk}_j\}_{j \in [N]}) \leftarrow \mathsf{Agg}(\mathsf{crs}, \mathbf{pk}) \\ = g_i(m, \mathsf{pub}) & \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, m) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

  The quantifier over pub is inside the probability because RFE-CED (Section 5.3) evaluates $\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}, \cdot)$ coherently on a superposition of public inputs.
- **Compactness.** There exists a polynomial $\mathsf{poly}(\cdot, \cdot, \cdot)$ such that for all $\lambda \in \mathbb{N}$, all $N \leq 2^\lambda$, all crs, all $\{(\mathsf{pk}_i, g_i)\}_{i \in [N]}$ with $\mathsf{IsValid} = 1$, and all outputs $(\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i \in [N]})$ of Agg, we have $|\mathsf{mpk}| \leq \mathsf{poly}(\lambda, |\mathcal{U}|, \log N)$ and $|\mathsf{hsk}_i| \leq \mathsf{poly}(\lambda, |\mathcal{U}|, \log N)$ for every $i \in [N]$.
- **Succinctness.** There exists a polynomial $\mathsf{poly}(\cdot, \cdot, \cdot)$ such that for all $\lambda \in \mathbb{N}$, all $N \leq 2^\lambda$, all crs, all mpk, all $m \in \mathcal{M}$, and all ct output by $\mathsf{Enc}(\mathsf{mpk}, m)$, we have $|\mathsf{ct}| \leq \mathsf{poly}(\lambda, |\mathcal{U}|, \log N)$.

4.2 Security of PS-RFE

Let $\prod_{\mathsf{RFE}}^{\mathsf{PS}}$ denote a public-slot registered functional encryption scheme. We say $\prod_{\mathsf{RFE}}^{\mathsf{PS}}$ is *secure with static corruptions* (or *static-corruption secure*) if for all QPT adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$ the advantage

$$\Pr\left[b' = b \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N); \\ b \leftarrow \{0,1\}; \\ \left(m_0^*, m_1^*, \{(k_i, g_i^*, \mathsf{pk}_i^*)\}_{i \in [N]}\right) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{crs}); \\ (\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i \in [N]}) \leftarrow \mathsf{Agg}\left(\mathsf{crs}, \{(\mathsf{pk}_i^*, g_i^*)\}_{i \in [N]}\right); \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, m_b^*); \\ b' \leftarrow \mathcal{A}(\mathsf{ct}^*) \end{array}\right] - \frac{1}{2} \;=\; \mathsf{negl}(\lambda)$$

where $\mathcal{A}$ has access to oracles $\mathcal{O}_{\mathsf{KeyGen}}(\cdot)$ and $\mathcal{O}_{\mathsf{Cor}}(\cdot)$, collectively $\mathcal{O}$, only before it outputs its challenge. Initialize $\mathsf{ctr} = 0$, dictionary $D[k] = \emptyset$ for all k, and a set $\mathsf{Cor} = \emptyset$ of corrupted key indices.

- $\mathcal{O}_{\mathsf{KeyGen}}(i)$: set $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$, run $(\mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}}) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, i)$, store $D[\mathsf{ctr}] := (i, \mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}})$, and return $(\mathsf{ctr}, \mathsf{pk}_{\mathsf{ctr}})$.
- $\mathcal{O}_{\mathsf{Cor}}(k)$: let $D[k] = (i, \mathsf{pk}, \mathsf{sk})$, add k to Cor, and return sk.

with the following restrictions:

- For each slot $i \in [N]$ in the challenge, $\mathcal{A}$ specifies $k_i \in \{1, \ldots, \mathsf{ctr}\}$ to reference a challenger-generated key, or $k_i = \perp$ to reference a key outside this set.
  - ◇ If $k_i \neq \perp$, let $D[k_i] = (i', \mathsf{pk}', \mathsf{sk}')$; it must hold that $i = i'$ and $\mathsf{pk}_i^* = \mathsf{pk}'$.
  - ◇ If $k_i = \perp$, it must hold that $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*) = 1$.

- For any query (k) to O_{Cor}, it holds that $D[k] \neq \emptyset$.
- The corrupted-slot set is fixed when $\mathcal{A}$ outputs its challenge: $\mathcal{F} := \{ i \in [N] : k_i = \bot \text{ or } k_i \in \mathsf{Cor} \}$.
- For every slot $i \in \mathcal{F}$ and every $\mathsf{pub} \in \mathcal{P}$, $g_i^*(m_0^*, \mathsf{pub}) = g_i^*(m_1^*, \mathsf{pub})$, except with negligible probability over the coins of the experiment.

4.3 Construction of PS-RFE

We construct a public-slot registered functional encryption scheme $\prod_{\mathsf{RFE}}^{\mathsf{PS}}$ by extending the slotted registered functional encryption construction of Datta *et al.* [6] with a public input pub threaded through the decryption circuit. Specifically, the algorithms Setup, KeyGen, $\mathsf{IsValid}$, and Agg are unchanged from Construction 8.2 of [6]. The only modifications are that the circuit C_j takes an additional input pub and outputs $g_i(m, \mathsf{pub})$ instead of $f_i(m)$, and that Dec forwards pub to $\widetilde{C}_0$. Consequently, the completeness, correctness, compactness, and succinctness guarantees carry over directly from the base construction. In particular, since the obfuscated circuit is deterministic and iO, SKE and the SSB opening are perfectly correct, $\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}, \cdot)$ agrees with $g_i(m, \cdot)$ on every public input simultaneously, as required in Section 4.1. We begin by defining the required building blocks; all of them are required to be secure against QPT adversaries.

- Let $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ be a length-doubling pseudorandom generator (Definition 3.3).
- Let $\mathsf{SKE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ be a secret-key encryption scheme [17].
- Let $\mathsf{SSB} = (\mathsf{Setup}, \mathsf{Hash}, \mathsf{Open}, \mathsf{Vrfy})$ be a somewhere statistically binding hash function, and set $\ell_{\mathsf{blk}} = |\mathcal{U}| + 2\lambda$ [13], instantiated from a post-quantum assumption such as LWE.
- Let iO be an indistinguishability obfuscator for $\mathsf{P/poly}$ [8] secure against QPT adversaries (Definition 3.1).

Decryption Circuit C_j

Constants: $\mathsf{mpk} = (\mathsf{hk}, h)$, SK_j, $V = (v_{k,\beta})_{k \in [2\ell_c], \beta \in \{0,1\}}$.

Input: $s_i \in \{0,1\}^\lambda$, $i \in [N]$, $\mathsf{pk}_i \in \{0,1\}^{2\lambda}$, $g_i \in \{0,1\}^{|\mathcal{U}|}$, $\pi_i \in \{0,1\}^{\ell_{\mathsf{open}}}$, $\mathsf{CT}_0, \mathsf{CT}_1$, $\sigma_{\mathsf{ct}} = (u_k)_{k \in [2\ell_c]}$, $\mathsf{pub} \in \{0,1\}^{\ell_p}$.

1. Parse $(\mathsf{CT}_0, \mathsf{CT}_1) = (\beta_1, \ldots, \beta_{\ell_c}, \beta_{\ell_c+1}, \ldots, \beta_{2\ell_c}) \in \{0,1\}^{2\ell_c}$.
2. If $\mathsf{SSB.Vrfy}(\mathsf{hk}, h, i, (\mathsf{pk}_i, g_i), \pi_i) = 1$, $\mathsf{PRG}(s_i) = \mathsf{pk}_i$, and $\mathsf{PRG}(u_k) = v_{k,\beta_k}$ for all $k \in [2\ell_c]$, then
   (a) Compute $m \leftarrow \mathsf{SKE.Dec}(\mathsf{SK}_j, \mathsf{CT}_j)$.
   (b) Output $g_i(m, \mathsf{pub})$.
3. Else, output $\bot$.

Fig. 2: The circuit $C_j = C_j[\mathsf{mpk}, \mathsf{SK}_j, V]$ for $j \in \{0,1\}$, for $\prod_{\mathsf{RFE}}^{\mathsf{PS}}$

$\mathsf{Enc}(\mathsf{mpk}, m)$:

- Sample $\mathsf{SK}_0, \mathsf{SK}_1 \leftarrow \mathsf{SKE.Setup}(1^\lambda)$, compute $\mathsf{CT}_0 \leftarrow \mathsf{SKE.Enc}(\mathsf{SK}_0, m)$ and $\mathsf{CT}_1 \leftarrow \mathsf{SKE.Enc}(\mathsf{SK}_1, 0^{|m|})$.
- Write $(\mathsf{CT}_0, \mathsf{CT}_1) = (\beta_1, \ldots, \beta_{\ell_c}, \beta_{\ell_c+1}, \ldots, \beta_{2\ell_c}) \in \{0,1\}^{2\ell_c}$.
- Sample $u_{k,\beta} \leftarrow \{0,1\}^\lambda$ for all $k \in [2\ell_c]$, $\beta \in \{0,1\}$, and compute $V = (v_{k,\beta} := \mathsf{PRG}(u_{k,\beta}))$.
- Construct the circuit $C_0 = C_0[\mathsf{mpk}, \mathsf{SK}_0, V]$ (Figure 2) and compute $\widetilde{C}_0 \leftarrow i\mathcal{O}(1^\lambda, C_0)$.
- Output $\mathsf{ct} := (\mathsf{CT}_0, \mathsf{CT}_1, \widetilde{C}_0, \sigma_{\mathsf{ct}} := (u_{k,\beta_k})_{k\in[2\ell_c]})$.

$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}, \mathsf{pub})$:

- Parse $\mathsf{sk}_i = s_i$, $\mathsf{hsk}_i = (i, \mathsf{pk}_i, g_i, \pi_i)$, and $\mathsf{ct} = (\mathsf{CT}_0, \mathsf{CT}_1, \widetilde{C}_0, \sigma_{\mathsf{ct}})$.
- Output $\widetilde{C}_0(s_i, i, \mathsf{pk}_i, g_i, \pi_i, \mathsf{CT}_0, \mathsf{CT}_1, \sigma_{\mathsf{ct}}, \mathsf{pub})$.

Lemma 2 (Sufficient Condition for Circuit Equivalence). *Fix circuits $C = C_j[\mathsf{mpk}, \mathsf{SK}_j, V]$ and $C' = C_{j'}[\mathsf{mpk}', \mathsf{SK}'_{j'}, V']$ as in Figure 2. Then $C \equiv C'$ in $\mathsf{P/poly}$ if the following two conditions hold. First, $\mathsf{mpk} = \mathsf{mpk}'$ and $V = V'$, so the guard in step 2 has the same value on every input. Second, for every input x on which this common guard accepts, one has $g_i(m, \mathsf{pub}) = g_i(m', \mathsf{pub})$ for every pub, where $m = \mathsf{SKE.Dec}(\mathsf{SK}_j, \mathsf{CT}_j)$ and $m' = \mathsf{SKE.Dec}(\mathsf{SK}'_{j'}, \mathsf{CT}_{j'})$.*

Proof. Assume that both conditions hold. For any input x, the common guard either rejects on both sides, in which case both circuits output $\perp$, or accepts on both sides. In the latter case, the second condition gives $g_i(m, \mathsf{pub}) = g_i(m', \mathsf{pub})$, and hence $C(x) = C'(x)$. This proves the lemma. We only need this sufficient condition; the converse does not hold in general, since, e.g., different values $V \neq V'$ may induce the same guard.

Theorem 1 (Security of $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$). *Assume that $i\mathcal{O}$ is an indistinguishability obfuscator for $\mathsf{P/poly}$ secure against QPT adversaries (Definition 3.1), SSB is a somewhere statistically binding hash function with index hiding against QPT adversaries, SKE satisfies the security notion required in [6] against QPT adversaries, and PRG is a secure pseudorandom generator (Definition 3.3). Then $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ is static-corruption secure (Section 4.2).*

Proof. By Lemma 2, the only relevant roles of g_i in C_j are its occurrence as an opaque $|\mathcal{U}|$-bit string in the SSB-hashed tuple and its use in the output value $g_i(m, \mathsf{pub})$. Thus every hybrid step from the base proof that leaves the decrypted plaintext unchanged depends only on the guard and carries over directly, with pub treated as an untouched input wire. The only modification is in the $i\mathcal{O}$ switching steps, where the base condition $f_i(m) = f_i(m')$ is replaced by $g_i(m, \mathsf{pub}) = g_i(m', \mathsf{pub})$ for every pub. We now make this precise. A black-box reduction to the security of the base scheme is not possible, since its ciphertexts take no public input; instead, we follow the proof of security of Construction 8.2 in [6] hybrid by hybrid, from the experiment with $b = 0$ to the experiment with $b = 1$, and only describe what changes.

– *Hybrids that modify only the guard of Step 2 or the values it checks.* These include the steps that use the pseudorandomness of PRG (for the public keys of honest slots and for the unused values $v_{k,1\oplus\beta_k}$), the index-hiding and somewhere statistical binding properties of SSB, and the security of SKE. Their reductions never evaluate a registered function, so they carry over verbatim, with pub treated as an additional input wire that the guard ignores. They remain valid against QPT adversaries because the underlying primitives are assumed to be secure against QPT adversaries.

– *Hybrids that change the plaintext used on the output line.* In [6], each such iO step is justified by the functional equivalence of the two circuits, which follows from the admissibility condition $f_i(m_0^*) = f_i(m_1^*)$ for the corrupted slots whose inputs can still pass the guard. We instead invoke Lemma 2: the two circuits are equivalent provided that $g_i(m_0^*, \mathsf{pub}) = g_i(m_1^*, \mathsf{pub})$ for every $\mathsf{pub} \in \mathcal{P}$ and every such slot $i \in \mathcal{F}$, which is exactly the admissibility condition of static-corruption security.

Since all corruptions occur before the challenge, $\mathcal{F}$ is fixed when the hybrids are defined, so no reduction has to decide the universally quantified condition $\forall\, \mathsf{pub} :\ g_i(m_0^*, \mathsf{pub}) = g_i(m_1^*, \mathsf{pub})$. In particular, the step of [6] that handles corruptions issued after the challenge, by corrupting all admissible keys in advance, is not needed here. Finally, admissibility, and hence functional equivalence, may fail with negligible probability; Definition 3.1 only requires equivalence with overwhelming probability, so each iO step still changes the advantage of $\mathcal{A}$ by at most $\mathsf{negl}(\lambda)$. As the number of hybrids is polynomial, $\prod_{\mathsf{RFE}}^{\mathsf{PS}}$ is static-corruption secure.

5 Registered Functional Encryption with Certified Everlasting Deletion

Let λ be the security parameter and N be the number of slots. Let $\mathcal{M}$ be the message space and $\mathcal{U} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ denote the class of functions on $\mathcal{M}$. A Registered Functional Encryption with Certified Everlasting Deletion scheme $\prod_{\mathsf{RFE}}^{\mathsf{CED}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{IsValid}, \mathsf{Agg}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ is a tuple of efficient algorithms with the following semantics:

$\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N)$

$(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, i)$

$b \in \{0,1\} \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i)$

$(\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[N]}) \leftarrow \mathsf{Agg}(\mathsf{crs}, \{(\mathsf{pk}_i, f_i)\}_{i\in[N]})$

$(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(\mathsf{mpk}, m)$

$y/\perp \leftarrow \mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct})$

$\mathsf{cert} \leftarrow \mathsf{Del}(\mathsf{ct})$

$\top/\perp \leftarrow \mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert})$

where

crs : common reference string

$N \in \mathbb{N}$: number of slots

$i \in [N]$: slot index

$f_i \in \mathcal{F}_\lambda$: function registered for slot i

$\mathsf{pk}_i, \mathsf{sk}_i$: public/secret key pair for slot i

mpk : master public key

hsk_i : helper secret key for slot i

$m \in \mathcal{M}$: message

ct : ciphertext

vk : verification key

y : decrypted output

cert : certificate of deletion

5.1 Correctness and Efficiency of RFE-CED

Let $\prod_{\mathsf{RFE}}^{\mathsf{CED}}$ denote a registered functional encryption scheme with certified everlasting deletion. We say $\prod_{\mathsf{RFE}}^{\mathsf{CED}}$ is *correct and efficient* if the following conditions hold.

- **Completeness.** For all $\lambda, N \in \mathbb{N}$ and all $i \in [N]$:

$$\Pr\left[\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i) = 1 \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N) \\ (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, i) \end{array}\right] = 1.$$

- **Evaluation Correctness.** For all $\lambda, N \in \mathbb{N}$, all $i \in [N]$, all $f_i \in \mathcal{F}_\lambda$, all $\{(j, f_j, \mathsf{pk}_j)\}_{j\neq i}$ with $\mathsf{IsValid}(\mathsf{crs}, j, \mathsf{pk}_j) = 1$, and all $m \in \mathcal{M}$, where crs ranges over the support of $\mathsf{Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N)$, writing $\mathbf{pk} = \{(\mathsf{pk}_j, f_j)\}_{j\in[N]}$,

$$\Pr\left[\begin{array}{l}\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \\ = f_i(m)\end{array} \;\middle|\; \begin{array}{l} (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, i) \\ (\mathsf{mpk}, \{\mathsf{hsk}_j\}_{j\in[N]}) \leftarrow \mathsf{Agg}(\mathsf{crs}, \mathbf{pk}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, m) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

- **Verification Correctness.** For all $\lambda, N \in \mathbb{N}$, all $\{(j, f_j, \mathsf{pk}_j)\}_{j\in[N]}$ with $\mathsf{IsValid}(\mathsf{crs}, j, \mathsf{pk}_j) = 1$, and all $m \in \mathcal{M}$, where crs ranges over the support of $\mathsf{Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N)$,

$$\Pr\left[\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert}) \neq \top \;\middle|\; \begin{array}{l} (\mathsf{mpk}, \{\mathsf{hsk}_j\}_{j\in[N]}) \leftarrow \mathsf{Agg}(\mathsf{crs}, \mathbf{pk}) \\ (\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(\mathsf{mpk}, m) \\ \mathsf{cert} \leftarrow \mathsf{Del}(\mathsf{ct}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

- **Compactness.** There exists a polynomial $\mathsf{poly}(\cdot,\cdot,\cdot)$ such that for all $\lambda \in \mathbb{N}$, all $N \leq 2^\lambda$, all crs, all $\{(\mathsf{pk}_i, f_i)\}_{i\in[N]}$ with $\mathsf{IsValid} = 1$, and all outputs $(\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[N]})$ of Agg:

$$|\mathsf{mpk}| \leq \mathsf{poly}(\lambda, |\mathcal{U}|, \log N) \qquad \text{and} \qquad \forall\, i \in [N] :\ |\mathsf{hsk}_i| \leq \mathsf{poly}(\lambda, |\mathcal{U}|, \log N).$$

- **Succinctness.** There exists a polynomial $\mathsf{poly}(\cdot,\cdot,\cdot)$ such that for all $\lambda \in \mathbb{N}$, all $N \leq 2^\lambda$, all crs, all mpk, all $m \in \mathcal{M}$, and all $(\mathsf{ct}, \mathsf{vk})$ output by Enc:

$$|\mathsf{ct}| \leq \mathsf{poly}(\lambda, |\mathcal{U}|, \log N).$$

5.2 Certified Everlasting Indistinguishable-Security of RFE-CED

Let $\prod_{\mathsf{RFE}}^{\mathsf{CED}}$ denote a registered functional encryption with certified everlasting deletion scheme. We say that $\prod_{\mathsf{RFE}}^{\mathsf{CED}}$ is *adaptively certified everlasting indistinguishable-secure* if for all non-uniform QPT adversaries $\mathcal{A} = \{\mathcal{A}_\lambda, |\psi\rangle_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, the following hold.

Experiment (a): Everlasting Privacy.

$$\mathsf{TD}\big(\mathsf{EvExp}_0^{\mathcal{A}}(\lambda), \mathsf{EvExp}_1^{\mathcal{A}}(\lambda)\big) \leq \mathsf{negl}(\lambda),$$

where, for each $b \in \{0,1\}$, $\mathsf{EvExp}_b^{\mathcal{A}}(\lambda)$ is the output of the following experiment, which is ρ if $\mathsf{Vrfy}(\mathsf{vk}_b, \mathsf{cert}) = \top$ and $\bot$ otherwise.

$$\begin{aligned}
\mathsf{crs} &\leftarrow \mathsf{Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N);\\
\big(m_0, m_1, \{(k_i, f_i^*, \mathsf{pk}_i^*)\}_{i\in[N]}\big) &\leftarrow \mathcal{A}_\lambda^{O}(\mathsf{crs});\\
\big(\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[N]}\big) &\leftarrow \mathsf{Agg}\big(\mathsf{crs}, \{(\mathsf{pk}_i^*, f_i^*)\}_{i\in[N]}\big);\\
(\mathsf{ct}_b, \mathsf{vk}_b) &\leftarrow \mathsf{Enc}(\mathsf{mpk}, m_b);\\
(\mathsf{cert}, \rho) &\leftarrow \mathcal{A}_\lambda^{O}(\mathsf{ct}_b).
\end{aligned}$$

Experiment (b): Post-Verification Indistinguishability.

$$\Pr\left[b' = b \;\middle|\; \begin{array}{l}
\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N);\\
\big(m_0, m_1,\ \{(k_i, f_i^*, \mathsf{pk}_i^*)\}_{i\in[N]}\big) \leftarrow \mathcal{A}_\lambda^{O}(\mathsf{crs});\\
(\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[N]}) \leftarrow \mathsf{Agg}\big(\mathsf{crs}, \{(\mathsf{pk}_i^*, f_i^*)\}_{i\in[N]}\big);\\
(\mathsf{ct}_b, \mathsf{vk}_b) \leftarrow \mathsf{Enc}(\mathsf{mpk}, m_b);\\
\mathsf{cert} \leftarrow \mathcal{A}_\lambda^{O}(\mathsf{ct}_b);\\
\beta \leftarrow \mathsf{Vrfy}(\mathsf{vk}_b, \mathsf{cert});\\
b' \leftarrow \mathcal{A}_\lambda^{O}(\beta)
\end{array} \right] - \frac{1}{2} \;=\; \mathsf{negl}(\lambda)$$

In both experiments, $\mathcal{A}_\lambda$ has access to oracles $O_{\mathsf{KeyGen}}(\cdot)$ and $O_{\mathsf{Cor}}(\cdot)$, collectively O, at any point of the experiment (before the challenge, between ct_b and cert, and, in Experiment (b), after verification). Initialize $\mathsf{ctr} = 0$, dictionary $D[k] = \emptyset$ for all k, and corrupted-slot set $\mathcal{F} = \emptyset$.

– $\mathcal{O}_{\mathsf{KeyGen}}(i)$: set $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$, run $(\mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}}) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, i)$, store $D[\mathsf{ctr}] := (i, \mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}})$, and return $(\mathsf{ctr}, \mathsf{pk}_{\mathsf{ctr}})$.
– $\mathcal{O}_{\mathsf{Cor}}(k)$: let $D[k] = (i, \mathsf{pk}, \mathsf{sk})$, add i to $\mathcal{F}$, and return sk.

with the following restrictions:

– For each slot $i \in [N]$ in the challenge, $\mathcal{A}$ specifies $k_i \in \{1, \ldots, \mathsf{ctr}\}$ to reference a challenger-generated key, or $k_i = \perp$ to reference a key outside this set.
  ◇ If $k_i \neq \perp$, let $D[k_i] = (i', \mathsf{pk}', \mathsf{sk}')$; it must hold that $i = i'$ and $\mathsf{pk}_i^* = \mathsf{pk}'$.
  ◇ If $k_i = \perp$, it must hold that $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*) = 1$, and i is added to $\mathcal{F}$.
– For any query (k) to $\mathcal{O}_{\mathsf{Cor}}$, it holds that $D[k] \neq \emptyset$.
– For every slot $i \in \mathcal{F}$, $f_i^*(m_0) = f_i^*(m_1)$.

5.3 Construction of RFE-CED

In this section we provide a construction of a registered functional encryption with certified everlasting deletion scheme. Let λ be a security parameter and $\mathcal{M} = \{0,1\}^n$ where $n = n(\lambda)$. We define the following building blocks.

– Let $\prod_{\mathsf{RFE}}^{\mathsf{PS}} = \mathsf{PS\text{-}RFE}.(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{IsValid}, \mathsf{Agg}, \mathsf{Enc}, \mathsf{Dec})$ be a public-slot registered functional encryption scheme for all polynomial size circuits that is correct, efficient and static-corruption secure (Section 4).
– Let $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ be a pseudorandom generator (Definition 3.3).

The algorithms of $\prod_{\mathsf{RFE}}^{\mathsf{CED}}$ are as follows.

$\mathsf{Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N)$:
  – Run $\mathsf{crs} \leftarrow \mathsf{PS\text{-}RFE.Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N)$.
  – Output crs.

$\mathsf{KeyGen}(\mathsf{crs}, i)$:
  – Run $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{PS\text{-}RFE.KeyGen}(\mathsf{crs}, i)$.
  – Output $(\mathsf{pk}_i, \mathsf{sk}_i)$.

$\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i)$:
  – Output $\mathsf{PS\text{-}RFE.IsValid}(\mathsf{crs}, i, \mathsf{pk}_i)$.

$\mathsf{Agg}(\mathsf{crs}, \{(\mathsf{pk}_i, f_i)\}_{i\in[N]})$:
  – For each $i \in [N]$, let $h[f_i]$ be the circuit of Figure 3, instantiated with the registered function f_i.
  – Run $(\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[N]}) \leftarrow \mathsf{PS\text{-}RFE.Agg}(\mathsf{crs}, \{(\mathsf{pk}_i, h[f_i])\}_{i\in[N]})$.
  – Output mpk and $\{\mathsf{hsk}_i\}_{i\in[N]}$.

Function $h[f]$

Secret Input: $V, \theta_1, \ldots, \theta_{2n+1}, \beta_1, \ldots, \beta_{2n+1}$

Public Input: $(b_{i,j}, u_{i,j})_{i\in[2n+1],j\in[\lambda]}$

1. Parse $V = (v_{i,j,b})_{i\in[2n+1],j\in[\lambda],b\in\{0,1\}}$.
2. Check whether $\mathsf{PRG}(u_{i,j}) = v_{i,j,b_{i,j}}$ holds for every $i \in [2n+1]$ and $j \in [\lambda]$. If so, proceed to the next step; otherwise output $\bot$.
3. Compute $m_i := \beta_i \oplus \bigoplus_{j:\theta_{i,j}=0} b_{i,j}$ for every $i \in [2n+1]$.
4. $$\text{Output } \begin{cases} f(m_1\|\cdots\|m_n), & \text{if } m_{2n+1} = 0, \\ f(m_{n+1}\|\cdots\|m_{2n}), & \text{otherwise.} \end{cases}$$

Fig. 3: The function $h[f]$ for $\prod^{\mathsf{CED}}_{\mathsf{RFE}}$

$\mathsf{Enc}(\mathsf{mpk}, m)$:

- Sample $z_i, \theta_i \leftarrow \{0,1\}^\lambda$ for every $i \in [2n+1]$.
- Sample $u_{i,j,b} \leftarrow \{0,1\}^\lambda$ and compute $v_{i,j,b} \leftarrow \mathsf{PRG}(u_{i,j,b})$ for every $i \in [2n+1]$, $j \in [\lambda]$, $b \in \{0,1\}$. Set $U := (u_{i,j,b})_{i,j,b}$ and $V := (v_{i,j,b})_{i,j,b}$.
- For every $i \in [2n+1]$ and $j \in [\lambda]$, generate the state
$$|\psi_{i,j}\rangle := \begin{cases} |z_{i,j}\rangle\,|u_{i,j,z_{i,j}}\rangle\,, & \text{if } \theta_{i,j} = 0, \\ \frac{1}{\sqrt{2}}\big(|0\rangle\,|u_{i,j,0}\rangle + (-1)^{z_{i,j}}\,|1\rangle\,|u_{i,j,1}\rangle\big), & \text{if } \theta_{i,j} = 1, \end{cases}$$
where $\theta_{i,j}$ (resp. $z_{i,j}$) is the j-th bit of θ_i (resp. z_i).
- Generate
$$\beta_i := \begin{cases} m_i \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n], \\ 0 \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n+1, 2n+1]. \end{cases}$$
- Compute $\mathsf{ct}_{\mathsf{ps}} \leftarrow \mathsf{PS\text{-}RFE.Enc}\big(\mathsf{mpk}, V\|\theta_1\|\cdots\|\theta_{2n+1}\|\beta_1\|\cdots\|\beta_{2n+1}\big)$.
- Output $\mathsf{ct} := \big(\mathsf{ct}_{\mathsf{ps}}, \bigotimes_{i\in[2n+1],j\in[\lambda]} |\psi_{i,j}\rangle\big)$ and $\mathsf{vk} := \big(U, (z_i, \theta_i)_{i\in[2n+1]}\big)$.

$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct})$:

- Parse $\mathsf{ct} = \big(\mathsf{ct}_{\mathsf{ps}}, \bigotimes_{i'\in[2n+1],j\in[\lambda]} |\psi_{i',j}\rangle\big)$.
- Apply $\mathsf{PS\text{-}RFE.Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}_{\mathsf{ps}}, \cdot)$ to $\bigotimes_{i',j} |\psi_{i',j}\rangle$, treating the register contents $(b_{i',j}, u_{i',j})_{i'\in[2n+1],j\in[\lambda]}$ as the public input $\mathsf{pub} \in \mathcal{P}' = (\{0,1\} \times \{0,1\}^\lambda)^{(2n+1)\lambda}$, and measure the output register to obtain y.
- Output y.

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

- Parse $\mathsf{ct} = \big(\mathsf{ct}_{\mathsf{ps}}, \bigotimes_{i\in[2n+1],j\in[\lambda]} |\psi_{i,j}\rangle\big)$.
- Measure $|\psi_{i,j}\rangle$ in the Hadamard basis to obtain $c_{i,j}\|d_{i,j} \in \{0,1\}^{\lambda+1}$, for every $i \in [2n+1]$ and $j \in [\lambda]$.

– Output $\mathsf{cert} := (c_{i,j}, d_{i,j})_{i\in[2n+1], j\in[\lambda]}$.

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

– Parse $\mathsf{vk} = \big(U, (z_i, \theta_i)_{i\in[2n+1]}\big)$ and $\mathsf{cert} = (c_{i,j}, d_{i,j})_{i\in[2n+1], j\in[\lambda]}$.
– For every $i \in [2n+1]$ and $j \in [\lambda]$ such that $\theta_{i,j} = 1$, check whether $z_{i,j} = c_{i,j} \oplus d_{i,j} \cdot (u_{i,j,0} \oplus u_{i,j,1})$, where $z_{i,j}$ is the j-th bit of z_i.
– If all checks pass, output $\top$; otherwise output $\bot$.

We prove the correctness, efficiency and security of the construction 5.3 below.

Theorem 2. *If $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ is correct and efficient then construction 5.3 of $\prod^{\mathsf{CED}}_{\mathsf{RFE}}$ is correct and efficient.*

Proof. The proof is divided into parts as follows.

- **Completeness and Compactness.** Both properties follow immediately from the corresponding properties of $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$, since KeyGen, $\mathsf{IsValid}$, and Agg are invoked verbatim.
- **Succinctness.** The ciphertext $\mathsf{ct} = \big(\mathsf{ct}_{\mathsf{ps}}, \bigotimes_{i\in[2n+1], j\in[\lambda]} |\psi_{i,j}\rangle\big)$ has two components. The classical part $\mathsf{ct}_{\mathsf{ps}}$ is an output of $\mathsf{PS\text{-}RFE.Enc}$, whose size is bounded by $\mathsf{poly}(\lambda, |\mathcal{U}|, \log N)$ by the succinctness of $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$. The quantum part $\bigotimes_{i\in[2n+1], j\in[\lambda]} |\psi_{i,j}\rangle$ consists of $(2n+1)\cdot\lambda$ registers, each over $O(\lambda)$ qubits (since each $u_{i,j,b} \in \{0,1\}^\lambda$), contributing $O((2n+1)\cdot\lambda^2) = \mathsf{poly}(\lambda)$ qubits in total.
- **Evaluation Correctness.** Let $\mathsf{ct} = \big(\mathsf{ct}_{\mathsf{ps}}, \bigotimes_{i\in[2n+1], j\in[\lambda]} |\psi_{i,j}\rangle\big)$ be an honestly generated ciphertext for message $m = m_1\|\cdots\|m_n$, and let $(\mathsf{sk}_{i^*}, \mathsf{hsk}_{i^*})$ be an honestly generated key pair for slot i^* with registered function f_{i^*}. By construction of Enc:
  – $\mathsf{ct}_{\mathsf{ps}} \leftarrow \mathsf{PS\text{-}RFE.Enc}\big(\mathsf{mpk}, V\|\theta_1\|\cdots\|\theta_{2n+1}\|\beta_1\|\cdots\|\beta_{2n+1}\big)$, and for all $i \in [2n+1]$, $j \in [\lambda]$ with $\theta_{i,j} = 0$, $|\psi_{i,j}\rangle = |z_{i,j}\rangle\,|u_{i,j,z_{i,j}}\rangle$.
  – $\mathsf{PRG}(u_{i,j,b}) = v_{i,j,b}$ for all $i \in [2n+1]$, $j \in [\lambda]$, $b \in \{0,1\}$, so the PRG check in step 2 of $h[f_{i^*}]$ (Figure 3) passes on all registers with $\theta_{i,j} = 0$, since those registers hold $|u_{i,j,z_{i,j}}\rangle$ in the $b_{i,j} = z_{i,j}$ branch, and on all registers with $\theta_{i,j} = 1$ in both branches.
  – For $i \in [n+1, 2n+1]$, the encryption sets $\beta_i = 0 \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}$, so that

$$m_i = \beta_i \oplus \bigoplus_{j:\theta_{i,j}=0} b_{i,j} = \Big(0 \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}\Big) \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j} = 0.$$

    In particular, taking $i = 2n+1$, we get $m_{2n+1} = 0$, so step 4 of $h[f_{i^*}]$ selects the branch $f_{i^*}(m_1\|\cdots\|m_n)$.
  – For $i \in [n]$, the encryption sets $\beta_i = m_i \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}$, so that

$$\beta_i \oplus \bigoplus_{j:\theta_{i,j}=0} b_{i,j} = \Big(m_i \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}\Big) \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j} = m_i.$$

    Hence step 3 of $h[f_{i^*}]$ correctly recovers m_i for every $i \in [n]$.

Applying $h[f_{i^*}]\big(V\|\theta_1\|\cdots\|\theta_{2n+1}\|\beta_1\|\cdots\|\beta_{2n+1},\cdot\big)$ to $\bigotimes_{i,j}|\psi_{i,j}\rangle$ and measuring the output therefore yields $f_{i^*}(m_1\|\cdots\|m_n) = f_{i^*}(m)$. Indeed, steps 3 and 4 of $h[f_{i^*}]$ read only registers with $\theta_{i,j}=0$, so every public input in the support of $\bigotimes_{i,j}|\psi_{i,j}\rangle$ yields this output, and by correctness of $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ for all public inputs the coherent evaluation leaves the output register in $|f_{i^*}(m)\rangle$ except with negligible probability.

- **Verification Correctness.** Let $\mathsf{ct} = \big(\mathsf{ct}_{\mathsf{ps}}, \bigotimes_{i\in[2n+1],j\in[\lambda]}|\psi_{i,j}\rangle\big)$ be an honestly generated ciphertext and $\mathsf{vk} = \big(U,(z_i,\theta_i)_{i\in[2n+1]}\big)$ the corresponding verification key. Fix any $i\in[2n+1]$ and $j\in[\lambda]$ with $\theta_{i,j}=1$. By construction of Enc, the register $|\psi_{i,j}\rangle$ was prepared as

$$|\psi_{i,j}\rangle = \tfrac{1}{\sqrt{2}}\big(|0\rangle|u_{i,j,0}\rangle + (-1)^{z_{i,j}}|1\rangle|u_{i,j,1}\rangle\big).$$

We apply the following lemma to analyze the Hadamard-basis measurement performed by Del.

Lemma 3. *Let $|\psi\rangle = |0\rangle_A|x_1\cdots x_n\rangle_B + |1\rangle_A|y_1\cdots y_n\rangle_B$. Measuring all qubits of register B in the Hadamard basis with outcome $\tilde{d}\in\{0,1\}^n$ leaves register A in the state $|0\rangle + (-1)^{\tilde{d}\cdot(x\oplus y)}|1\rangle$, up to normalization.*

Proof. For any basis state $|b\rangle$ and Hadamard-basis projector $|d_i\rangle\langle d_i|$ on a single qubit, one verifies $|d_i\rangle\langle d_i|\,|b\rangle = (-1)^{d_i\cdot b}|d_i\rangle$ (identifying $|+\rangle\leftrightarrow|0\rangle$ and $|-\rangle\leftrightarrow|1\rangle$). Applying this to all n qubits of register B, measuring register B and obtaining outcome $\tilde{d}$ collapses the state to

$$\big((-1)^{\sum_i \tilde{d}_i x_i}|0\rangle + (-1)^{\sum_i \tilde{d}_i y_i}|1\rangle\big)|\tilde{d}\rangle = (-1)^{\tilde{d}\cdot x}\big(|0\rangle + (-1)^{\tilde{d}\cdot(x\oplus y)}|1\rangle\big)|\tilde{d}\rangle,$$

so register A is left in the state $|0\rangle + (-1)^{\tilde{d}\cdot(x\oplus y)}|1\rangle$.

Applying Lemma 3 to $|\psi_{i,j}\rangle$ with $|0\rangle_A|u_{i,j,0}\rangle_B + (-1)^{z_{i,j}}|1\rangle_A|u_{i,j,1}\rangle_B$ (identifying $x=u_{i,j,0}$, $y=u_{i,j,1}$), measuring register B in the Hadamard basis with outcome $d_{i,j}\in\{0,1\}^\lambda$ leaves register A in the state

$$|0\rangle + (-1)^{z_{i,j}+d_{i,j}\cdot(u_{i,j,0}\oplus u_{i,j,1})}|1\rangle.$$

Now measuring register A in the Hadamard basis yields outcome $c_{i,j}\in\{0,1\}$, where $c_{i,j}=0$ implies $z_{i,j} = d_{i,j}\cdot(u_{i,j,0}\oplus u_{i,j,1})$, and $c_{i,j}=1$ implies $z_{i,j}\neq d_{i,j}\cdot(u_{i,j,0}\oplus u_{i,j,1})$. In both cases we conclude

$$z_{i,j} \;=\; c_{i,j}\oplus d_{i,j}\cdot(u_{i,j,0}\oplus u_{i,j,1}),$$

which is exactly the relation checked by Vrfy.

Theorem 3 (Everlasting Privacy). *If $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ is a static-corruption secure public-slot registered functional encryption scheme and PRG is a secure pseudorandom generator, then Construction 5.3 satisfies Everlasting Privacy (Experiment (a) in Section 5.2).*

Proof. Let $\mathcal{A}$ be a QPT adversary against the everlasting privacy of $\prod_{\mathsf{RFE}}^{\mathsf{CED}}$. Our analysis relies on the interactive version of the Certified Everlasting Lemma [1] as formulated in Lemma 3.5 of [10] (Lemma 1). We consider the following sequence of hybrids.

$\mathbf{Hyb}_0$: This is the original everlasting security experiment where the challenge bit is set to be 0, i.e., $\mathsf{EvExp}_0^{\mathcal{A}}(\lambda)$.

- *Setup.* The challenger runs $\mathsf{crs} \leftarrow \mathsf{PS\text{-}RFE.Setup}(1^\lambda, 1^{|\mathcal{U}|}, 1^N)$ and sends crs to $\mathcal{A}$. Initialize $\mathsf{ctr} = 0$, $D[k] = \emptyset$ for all k, and $\mathcal{F} = \emptyset$.
- *Key-generation and corruption queries.* $\mathcal{A}$ may, at any point of the experiment, query $\mathcal{O}_{\mathsf{KeyGen}}(i)$: the challenger sets $\mathsf{ctr} \leftarrow \mathsf{ctr}+1$, runs $(\mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}}) \leftarrow \mathsf{PS\text{-}RFE.KeyGen}(\mathsf{crs}, i)$, stores $D[\mathsf{ctr}] := (i, \mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}})$, and returns $(\mathsf{ctr}, \mathsf{pk}_{\mathsf{ctr}})$. On $\mathcal{O}_{\mathsf{Cor}}(k)$ with $D[k] = (i, \mathsf{pk}, \mathsf{sk})$, the challenger adds i to $\mathcal{F}$ and returns sk.
- *Challenge phase.* $\mathcal{A}$ sends $(m^{(0)}, m^{(1)}) \in \mathcal{M}^2$ and $\{(k_i, f_i^*, \mathsf{pk}_i^*)\}_{i\in[N]}$ to the challenger, subject to the restrictions of Definition 5.2: for each $i \in [N]$, either $k_i \in \{1, \ldots, \mathsf{ctr}\}$ with $D[k_i] = (i, \mathsf{pk}_i^*, \cdot)$, or $k_i = \perp$ with $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*) = 1$ and i added to $\mathcal{F}$. Additionally, for every $i \in \mathcal{F}$, $f_i^*(m^{(0)}) = f_i^*(m^{(1)})$, must hold both before and after the challenge.
  For each $i \in [N]$, let $h[f_i^*]$ be the circuit of Figure 3, and compute

$$(\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[N]}) \leftarrow \mathsf{PS\text{-}RFE.Agg}(\mathsf{crs}, \{(\mathsf{pk}_i^*, h[f_i^*])\}_{i\in[N]}).$$

  The challenger generates $(\mathsf{ct}, \mathsf{vk}) \leftarrow \mathsf{Enc}(\mathsf{mpk}, m^{(0)})$. Specifically:
  (a) Generate $z_i, \theta_i \leftarrow \{0,1\}^\lambda$ for every $i \in [2n+1]$.
  (b) Generate $u_{i,j,b} \leftarrow \{0,1\}^\lambda$ and compute $v_{i,j,b} \leftarrow \mathsf{PRG}(u_{i,j,b})$ for every $i \in [2n+1]$, $j \in [\lambda]$, $b \in \{0,1\}$. Set $U := (u_{i,j,b})_{i,j,b}$, $V := (v_{i,j,b})_{i,j,b}$.
  (c) Generate the state

$$|\psi_{i,j}\rangle := \begin{cases} |z_{i,j}\rangle\,|u_{i,j,z_{i,j}}\rangle\,, & \text{if } \theta_{i,j} = 0, \\ \frac{1}{\sqrt{2}}\big(|0\rangle\,|u_{i,j,0}\rangle + (-1)^{z_{i,j}}\,|1\rangle\,|u_{i,j,1}\rangle\big), & \text{if } \theta_{i,j} = 1. \end{cases}$$

  (d) Generate

$$\beta_i := \begin{cases} m_i^{(0)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n], \\ 0 \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n+1, 2n+1]. \end{cases}$$

  (e) Compute $\mathsf{ct}_{\mathsf{ps}} \leftarrow \mathsf{PS\text{-}RFE.Enc}\big(\mathsf{mpk}, V\|\theta_1\|\cdots\|\theta_{2n+1}\|\beta_1\|\cdots\|\beta_{2n+1}\big)$.
  (f) Set $\mathsf{ct} = \big(\mathsf{ct}_{\mathsf{ps}}, \bigotimes_{i\in[2n+1], j\in[\lambda]} |\psi_{i,j}\rangle\big)$ and $\mathsf{vk} = \big(U, (z_i, \theta_i)_{i\in[2n+1]}\big)$.
  The challenger sends ct to $\mathcal{A}$.
- *Further oracle queries.* $\mathcal{A}$ may issue further $\mathcal{O}_{\mathsf{KeyGen}}$ and $\mathcal{O}_{\mathsf{Cor}}$ queries.
- *Deletion and verification.* $\mathcal{A}$ sends $\mathsf{cert} = (c_{i,j}, d_{i,j})_{i\in[2n+1], j\in[\lambda]}$ and its internal state ρ to the challenger. The challenger checks whether $z_{i,j} = c_{i,j} \oplus d_{i,j} \cdot (u_{i,j,0} \oplus u_{i,j,1})$ holds for every $i \in [2n+1]$ and $j \in [\lambda]$ such that $\theta_{i,j} = 1$. If not, the experiment outputs $\perp$. Otherwise, the experiment outputs ρ.

$\mathbf{Hyb}_{1,k}$, for $k = 0, 1, \ldots, n$: Same as $\mathbf{Hyb}_0$ except step (d) of the challenge phase is replaced by

$$\beta_i := \begin{cases} m_i^{(0)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n], \\ m_{i-n}^{(1)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n+1, n+k], \\ 0 \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n+k+1, 2n+1]. \end{cases}$$

Note that $\mathbf{Hyb}_{1,0}$ is identical to $\mathbf{Hyb}_0$.

$\mathbf{Hyb}_2$: Same as $\mathbf{Hyb}_{1,n}$ except β_{2n+1} is flipped. That is,

$$\beta_i := \begin{cases} m_i^{(0)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n], \\ m_{i-n}^{(1)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n+1, 2n], \\ 1 \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i = 2n+1. \end{cases}$$

$\mathbf{Hyb}_{3,k}$: For $k = 0, 1, \ldots, n$: Same as $\mathbf{Hyb}_2$ except step (d) is replaced by

$$\beta_i := \begin{cases} m_i^{(1)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [k], \\ m_i^{(0)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [k+1, n], \\ m_{i-n}^{(1)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n+1, 2n], \\ 1 \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i = 2n+1. \end{cases}$$

Note that $\mathbf{Hyb}_{3,0}$ is identical to $\mathbf{Hyb}_2$.

$\mathbf{Hyb}_4$: Same as $\mathbf{Hyb}_{3,n}$ except β_{2n+1} is flipped. That is,

$$\beta_i := \begin{cases} m_i^{(1)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n], \\ m_{i-n}^{(1)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n+1, 2n], \\ 0 \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i = 2n+1. \end{cases}$$

$\mathbf{Hyb}_{5,k}$: For $k = 0, 1, \ldots, n$: Same as $\mathbf{Hyb}_4$ except step (d) is replaced by

$$\beta_i := \begin{cases} m_i^{(1)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n], \\ 0 \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n+1, n+k], \\ m_{i-n}^{(1)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n+k+1, 2n], \\ 0 \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i = 2n+1. \end{cases}$$

Observe first that $\mathbf{Hyb}_{5,0}$ coincides with $\mathbf{Hyb}_4$, while $\mathbf{Hyb}_{5,n}$ corresponds to the everlasting-security experiment in which the challenge bit is fixed to 1, i.e., $\mathsf{EvExp}_1^{\mathcal{A}}(\lambda)$. Hence, in order to prove everlasting privacy of $\prod_{\mathsf{RFE}}^{\mathsf{CED}}$, it is enough to show that

$\mathsf{TD}(\mathbf{Hyb}_0, \mathbf{Hyb}_{5,n}) \leq \mathsf{negl}(\lambda)$. The hybrids fill the inactive second block with $m^{(1)}$, switch the selector, change the now inactive first block to $m^{(1)}$, switch the selector back and clear the second block.

Simulating corruptions. Each reduction to $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ below forwards the oracle queries of $\mathcal{A}$ until $\mathcal{A}$ outputs its challenge. Before issuing its own challenge, it corrupts every challenger-generated key that is not registered and every $k_i \neq \perp$ with $f_i^*(m^{(0)}) = f_i^*(m^{(1)})$, and stores the returned keys. By admissibility, it answers all later corruption queries from storage, and it answers later key-generation queries by running KeyGen itself. Hence the corrupted-slot set of the $\mathsf{PS\text{-}RFE}$ game is $\mathcal{F}^* = \{i \in [N] \mid f_i^*(m^{(0)}) = f_i^*(m^{(1)})\}$, and the reduction is admissible whenever $h[f_i^*]$ agrees on its two challenge plaintexts for every $i \in \mathcal{F}^*$ and every public input.

Embedding the target bit. In each of Lemmas 4 to 8, the two adjacent hybrids differ only in one masked bit, which encodes some c in the earlier hybrid and some c' in the later one. If $c = c'$ the hybrids are identical. Otherwise the auxiliary challenger sets this masked bit to $c \oplus \beta$, so that $b = 0$ and $b = 1$ in the reduction yield the earlier and the later hybrid, respectively.

We establish this bound through a standard hybrid argument. Specifically, we analyze each pair of adjacent hybrid distributions and prove their indistinguishability in Lemmas 4–8.

Lemma 4. *If $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ is static-corruption secure and PRG is a secure pseudorandom generator, then for any $k \in [n]$, $\mathsf{TD}(\mathbf{Hyb}_{1,k-1}, \mathbf{Hyb}_{1,k}) \leq \mathsf{negl}(\lambda)$. Note this shows $\mathsf{TD}(\mathbf{Hyb}_0, \mathbf{Hyb}_{1,n}) \leq \mathsf{negl}(\lambda)$*

Proof. The only difference between $\mathbf{Hyb}_{1,k-1}$ and $\mathbf{Hyb}_{1,k}$ is that β_{n+k} is set to $0 \oplus \bigoplus_{j:\theta_{n+k,j}=0} z_{n+k,j}$ in $\mathbf{Hyb}_{1,k-1}$ and to $m_k^{(1)} \oplus \bigoplus_{j:\theta_{n+k,j}=0} z_{n+k,j}$ in $\mathbf{Hyb}_{1,k}$. If $m_k^{(1)} = 0$ the two hybrids are identical, so assume $m_k^{(1)} = 1$.

Auxiliary experiment. Fix $\theta \in \{0,1\}^\lambda$ and $\beta \in \{0,1\}$. Define the experiment $\mathsf{Expt}^{1,k}_{\mathcal{B},C}(\lambda, \theta, \beta)$ in which $\mathcal{B}$ and C play the roles of $\mathcal{A}$ and the challenger of $\mathbf{Hyb}_{1,k}$, with the following modifications: C sets $\beta_{n+k} := \beta$ and $\theta_{n+k} := \theta$, and does not prepare $|\psi_{n+k,j}\rangle$ for $j \in [\lambda]$ nor sends it to $\mathcal{B}$. Additionally C sends $\{u_{n+k,j,b}\}_{j\in[\lambda],b\in\{0,1\}}$ to $\mathcal{B}$ and $\mathcal{B}$ outputs a bit b' rather than a certificate. Before that, $\mathcal{B}$ may send $(c_{i,j}, d_{i,j})_{i\neq n+k, j\in[\lambda]}$ to C, which returns $\mathsf{flag} = \mathsf{acc}$ if these pass the check of Vrfy and $\mathsf{flag} = \mathsf{rej}$ otherwise. This check reads $u_{i,j,b}$ only where $\theta_{i,j} = 1$. In particular, β_{n+k} is set as follows in C's encryption step:

$$\beta_i := \begin{cases} m_i^{(0)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n], \\ m_{i-n}^{(1)} \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n+1, n+k-1], \\ \beta, & \text{if } i = n+k, \\ 0 \oplus \bigoplus_{j:\theta_{i,j}=0} z_{i,j}, & \text{if } i \in [n+k+1, 2n+1]. \end{cases}$$

We prove the following claim.

Claim 4. *For any QPT $\mathcal{B}$, $\left|\Pr[\mathsf{Expt}^{1,k}_{\mathcal{B},C}(\lambda,\theta,\beta) = 1] - \Pr[\mathsf{Expt}^{1,k}_{\mathcal{B},C}(\lambda, 0^\lambda, \beta) = 1]\right| \leq \mathsf{negl}(\lambda)$.*

We establish Claim 4 via the following sub-experiments in a three step process.

- $\mathsf{Expt}^{1,k,a}_{\mathcal{B},C}(\lambda,\theta,\beta)$: Same as $\mathsf{Expt}^{1,k}_{\mathcal{B},C}(\lambda,\theta,\beta)$ except that $v_{i,j,1\oplus z_{i,j}}$ is sampled uniformly from $\{0,1\}^{2\lambda}$ rather than set to $\mathsf{PRG}(u_{i,j,1\oplus z_{i,j}})$, for all $i \in [2n+1] \setminus \{n+k\}$ and $j \in [\lambda]$ with $\theta_{i,j} = 0$.
- $\mathsf{Expt}^{1,k,b}_{\mathcal{B},C}(\lambda,\theta,\beta)$: Same as $\mathsf{Expt}^{1,k,a}_{\mathcal{B},C}(\lambda,\theta,\beta)$ except $\theta_{n+k} = \theta$ is replaced with 0^λ. Note that θ_{n+k} appears only inside the PS-RFE ciphertext.

Step 1. $|\Pr[\mathsf{Expt}^{1,k}_{\mathcal{B},C}(\lambda,\theta,\beta)=1] - \Pr[\mathsf{Expt}^{1,k,a}_{\mathcal{B},C}(\lambda,\theta,\beta)=1]| \le \mathsf{negl}(\lambda)$.

Since $u_{i,j,1\oplus z_{i,j}}$ for $i \in [2n+1] \setminus \{n+k\}$, $j \in [\lambda]$, $\theta_{i,j} = 0$ is used only to generate $v_{i,j,1\oplus z_{i,j}} = \mathsf{PRG}(u_{i,j,1\oplus z_{i,j}})$ and not in computing flag, this transition is indistinguishable by the pseudorandomness of PRG.

Step 2. $|\Pr[\mathsf{Expt}^{1,k,a}_{\mathcal{B},C}(\lambda,\theta,\beta)=1] - \Pr[\mathsf{Expt}^{1,k,b}_{\mathcal{B},C}(\lambda,\theta,\beta)=1]| \le \mathsf{negl}(\lambda)$.

Let Good be the event that $v_{i,j,1\oplus z_{i,j}} \notin \mathsf{Im}(\mathsf{PRG})$ for all $i \in [2n+1] \setminus \{n+k\}$, $j \in [\lambda]$, $\theta_{i,j} = 0$; since each $v_{i,j,1\oplus z_{i,j}}$ is now uniform over $\{0,1\}^{2\lambda}$, we have $\Pr[\mathsf{Good}] \ge 1 - 2n\lambda \cdot 2^{-\lambda} = 1 - \mathsf{negl}(\lambda)$. Conditioned on Good, the only input $(b_{i,j}, u_{i,j})$ that passes the PRG check in $h[f_i^*]$ satisfies $b_{i,j} = z_{i,j}$ for all such (i,j), so for every key query f and every public input $(b_{i,j}, u_{i,j})_{i,j}$:

$$\begin{aligned} &h[f]\big(V\|\theta_1\|\cdots\|\theta_{2n+1}\|\beta_1\|\cdots\|\beta_{2n+1},\ (b_{i,j},u_{i,j})_{i,j}\big) \\ &= h[f]\big(V\|\theta_1\|\cdots\|\theta_{n+k-1}\|0^\lambda\|\theta_{n+k+1}\|\cdots\|\theta_{2n+1}\|\beta_1\|\cdots\|\beta_{2n+1},\ (b_{i,j},u_{i,j})_{i,j}\big), \end{aligned}$$

since both sides output $\perp$ if the check fails. Otherwise Good forces $b_{i,j} = z_{i,j}$ for all $i \ne n+k$ and j with $\theta_{i,j} = 0$, so the selector is $m_{2n+1} = 0$ and both sides output $f(m^{(0)})$. The index $n+k$ is excluded from the PRG programming, so m_{n+k} is unconstrained, but it lies in the second block, which is inactive when the selector is 0. The two ciphertexts therefore encrypt functionally equivalent messages for every f, and indistinguishability follows from the static-corruption security of $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$. Admissibility fails only if Good fails, which happens with negligible probability.

Step 3. $|\Pr[\mathsf{Expt}^{1,k,b}_{\mathcal{B},C}(\lambda,\theta,\beta)=1] - \Pr[\mathsf{Expt}^{1,k}_{\mathcal{B},C}(\lambda,0^\lambda,\beta)=1]| \le \mathsf{negl}(\lambda)$.

Symmetric to Step 1: $u_{i,j,1\oplus z_{i,j}}$ is used only to generate $v_{i,j,1\oplus z_{i,j}}$ in $\mathsf{Expt}^{1,k}_{\mathcal{B},C}(\lambda,0^\lambda,\beta)$, so pseudorandomness of PRG gives the claim. This establishes Claim 4.

Reduction. Define $\widetilde{\mathsf{Expt}}^{1,k}_{\mathcal{B}',C}(\lambda,b)$ as the experiment that samples $z,\theta \leftarrow \{0,1\}^\lambda$, gives $\mathcal{B}'$ the state $(1^\lambda, |z\rangle_\theta)$, runs $\mathcal{B}'$ as $\mathcal{B}$ in $\mathsf{Expt}^{1,k}_{\mathcal{B},C}(\lambda,\theta,b\oplus\bigoplus_{j:\theta_j=0} z_j)$, and outputs $\mathcal{B}'$'s state ρ if $\mathcal{B}'$'s output z' satisfies $z_j = z'_j$ for all $j \in [\lambda]$ with $\theta_j = 1$, and $\perp$ otherwise. By Claim 4 and the Interactive Certified Everlasting Lemma [10],

$$\mathsf{TD}\big(\widetilde{\mathsf{Expt}}^{1,k}_{\mathcal{B}',C}(\lambda,0),\ \widetilde{\mathsf{Expt}}^{1,k}_{\mathcal{B}',C}(\lambda,1)\big) \le \mathsf{negl}(\lambda).$$

We now build $\mathcal{B}'$ from any adversary $\mathcal{A}$ that distinguishes $\mathbf{Hyb}_{1,k-1}$ and $\mathbf{Hyb}_{1,k}$. $\mathcal{B}'(1^\lambda, |z\rangle_\theta)$ runs $\mathcal{A}$ in $\mathbf{Hyb}_{1,k}$, using the external challenger C to produce everything except $|\psi_{n+k,j}\rangle$ for $j \in [\lambda]$. It constructs these registers by applying the map $|b\rangle \mapsto |b\rangle\,|u_{n+k,j,b}\rangle$ to the j-th qubit of $|z\rangle_\theta$ (possible since C supplies $(u_{n+k,j,b})_{j,b}$). When $\mathcal{A}$

returns certificate $(c_{i,j}, d_{i,j})_{i,j}$, $\mathcal{B}'$ sets $z'_{n+k,j} = c_{n+k,j} \oplus d_{n+k,j} \cdot (u_{n+k,j,0} \oplus u_{n+k,j,1})$ for each $j \in [\lambda]$, forwards $(c_{i,j}, d_{i,j})_{i \neq n+k, j \in [\lambda]}$ to C to obtain flag, and outputs $z' := z'_{n+k,1} \| \cdots \| z'_{n+k,\lambda}$ and the state (flag, ρ), where ρ is the internal state of $\mathcal{A}$. $\mathcal{B}'$ perfectly simulates $\mathbf{Hyb}_{1,k-1}$ when $b = 0$ and $\mathbf{Hyb}_{1,k}$ when $b = 1$ (recall $m_k^{(1)} = 1$). Let Φ be the CPTP map with $\Phi(\perp) = \Phi(\mathsf{rej}, \rho) = \perp$ and $\Phi(\mathsf{acc}, \rho) = \rho$. Then Φ maps $\widetilde{\mathsf{Expt}}^{1,k}_{\mathcal{B}',C}(\lambda, 0)$ to $\mathbf{Hyb}_{1,k-1}$ and $\widetilde{\mathsf{Expt}}^{1,k}_{\mathcal{B}',C}(\lambda, 1)$ to $\mathbf{Hyb}_{1,k}$, and the trace distance does not increase under Φ. Therefore,

$$\mathsf{TD}(\mathbf{Hyb}_{1,k-1}, \mathbf{Hyb}_{1,k}) \;\leq\; \mathsf{TD}\big(\widetilde{\mathsf{Expt}}^{1,k}_{\mathcal{B}',C}(\lambda, 0),\, \widetilde{\mathsf{Expt}}^{1,k}_{\mathcal{B}',C}(\lambda, 1)\big) \;\leq\; \mathsf{negl}(\lambda).$$

Lemma 5. *If* $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ *is static-corruption secure and* PRG *is a secure pseudorandom generator, then* $\mathsf{TD}(\mathbf{Hyb}_{1,n}, \mathbf{Hyb}_2) \leq \mathsf{negl}(\lambda)$.

Proof. The proof follows the same structure as Lemma 4, with the target index $n + k$ replaced throughout by $2n + 1$. Define $\mathsf{Expt}^2_{\mathcal{B},C}(\lambda, \theta, \beta)$ identically to $\mathsf{Expt}^{1,k}_{\mathcal{B},C}(\lambda, \theta, \beta)$ but with $n + k$ replaced by $2n + 1$. By the same reduction via the Interactive Certified Everlasting Lemma, it suffices to show that for all QPT $\mathcal{B}$,

$$\big|\Pr\big[\mathsf{Expt}^2_{\mathcal{B},C}(\lambda, \theta, \beta) = 1\big] - \Pr\big[\mathsf{Expt}^2_{\mathcal{B},C}(\lambda, 0^\lambda, \beta) = 1\big]\big| \leq \mathsf{negl}(\lambda).$$

Define sub-experiments $\mathsf{Expt}^{2,a}_{\mathcal{B},C}(\lambda, \theta, \beta)$ and $\mathsf{Expt}^{2,b}_{\mathcal{B},C}(\lambda, \theta, \beta)$ analogously to $\mathsf{Expt}^{1,k,a}_{\mathcal{B},C}$ and $\mathsf{Expt}^{1,k,b}_{\mathcal{B},C}$ with $n + k$ replaced by $2n + 1$. The transitions $\mathsf{Expt}^2 \to \mathsf{Expt}^{2,a}$ and $\mathsf{Expt}^{2,b} \to \mathsf{Expt}^2(\lambda, 0^\lambda, \beta)$ follow immediately from the pseudorandomness of PRG, by the same argument as Steps 1 and 3 of Claim 4. It remains to argue the transition $\mathsf{Expt}^{2,a} \to \mathsf{Expt}^{2,b}$ via the static-corruption security of $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$.

Let Good be the event that $v_{i,j,1\oplus z_{i,j}} \notin \mathsf{Im}(\mathsf{PRG})$ for every $i \in [2n]$ and $j \in [\lambda]$ with $\theta_{i,j} = 0$. Since each such $v_{i,j,1\oplus z_{i,j}}$ is uniform over $\{0,1\}^{2\lambda}$ in $\mathsf{Expt}^{2,a}$, we have $\Pr[\mathsf{Good}] \geq 1 - \mathsf{negl}(\lambda)$. Conditioned on Good, we claim that for every $i \in \mathcal{F}^*$ and every public input $(b_{i,j}, u_{i,j})_{i \in [2n+1], j \in [\lambda]}$,

$$\begin{aligned} &h[f_i^*]\big(V \| \theta_1 \| \cdots \| \theta_{2n+1} \| \beta_1 \| \cdots \| \beta_{2n+1},\, (b_{i,j}, u_{i,j})_{i,j}\big) \\ &= h[f_i^*]\big(V \| \theta_1 \| \cdots \| \theta_{2n} \| 0^\lambda \| \beta_1 \| \cdots \| \beta_{2n+1},\, (b_{i,j}, u_{i,j})_{i,j}\big). \end{aligned}$$

We verify this in two cases.

– If $\mathsf{PRG}(u_{i,j}) = v_{i,j,b_{i,j}}$ holds for every $i \in [2n + 1]$ and $j \in [\lambda]$, then Good forces $b_{i,j} = z_{i,j}$ for all $i \in [2n]$, $j \in [\lambda]$ with $\theta_{i,j} = 0$. Consequently,

$$\beta_i \oplus \bigoplus_{j:\theta_{i,j}=0} b_{i,j} \;=\; \begin{cases} m_i^{(0)}, & i \in [n], \\ m_{i-n}^{(1)}, & i \in [n+1, 2n]. \end{cases}$$

Let $\gamma = \beta_{2n+1} \oplus \bigoplus_{j:\theta_{2n+1,j}=0} b_{2n+1,j}$ and $\gamma' = \beta_{2n+1} \oplus \bigoplus_{j \in [\lambda]} b_{2n+1,j}$. The left-hand side outputs $f_i^*(m^{(\gamma)})$ and the right-hand side outputs $f_i^*(m^{(\gamma')})$. By the definition of $\mathcal{F}^*$, $f_i^*(m^{(0)}) = f_i^*(m^{(1)})$, so both sides are equal.

– Otherwise, both sides output $\perp$.

In both cases the two plaintexts are functionally equivalent for every slot in $\mathcal{F}^*$, and admissibility fails only if Good fails. Hence the static-corruption security of $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ gives the required indistinguishability.

Lemma 6. *If $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ is static-corruption secure and PRG is a secure pseudorandom generator, then for any $k \in [n]$, $\mathsf{TD}(\mathbf{Hyb}_{3,k-1}, \mathbf{Hyb}_{3,k}) \le \mathsf{negl}(\lambda)$.*

Proof. The proof is identical to that of Lemma 4, with the only difference that β_k is set to $m^{(0)}_k \oplus \bigoplus_{j:\theta_{k,j}=0} z_{k,j}$ in $\mathbf{Hyb}_{3,k-1}$ and to $m^{(1)}_k \oplus \bigoplus_{j:\theta_{k,j}=0} z_{k,j}$ in $\mathbf{Hyb}_{3,k}$, so the target index is k. The auxiliary experiment $\mathsf{Expt}^{3,k}_{\mathcal{B},C}(\lambda,\theta,\beta)$ and the three-step transition via PRG security, static-corruption security of $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$, and PRG security proceed verbatim as in Lemma 4. In Step 2, Good now forces the selector to 1, so the index k lies in the inactive first block.

Lemma 7. *If $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ is static-corruption secure and PRG is a secure pseudorandom generator, then $\mathsf{TD}(\mathbf{Hyb}_{3,n}, \mathbf{Hyb}_4) \le \mathsf{negl}(\lambda)$.*

Proof. The proof is identical to that of Lemma 5, with index $2n+1$ again singled out and β_{2n+1} flipped from $1 \oplus \bigoplus_{j:\theta_{2n+1,j}=0} z_{2n+1,j}$ to $0 \oplus \bigoplus_{j:\theta_{2n+1,j}=0} z_{2n+1,j}$. The auxiliary experiment $\mathsf{Expt}^{4}_{\mathcal{B},C}(\lambda,\theta,\beta)$ and the functional-equivalence argument proceed verbatim as in Lemma 5. Here both blocks encode $m^{(1)}$ in $\mathbf{Hyb}_{3,n}$ and $\mathbf{Hyb}_4$, so both sides of the functional-equivalence equation output $f(m^{(1)})$ or $\perp$ for every f.

Lemma 8. *If $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ is static-corruption secure and PRG is a secure pseudorandom generator, then for any $k \in [n]$, $\mathsf{TD}(\mathbf{Hyb}_{5,k-1}, \mathbf{Hyb}_{5,k}) \le \mathsf{negl}(\lambda)$.*

Proof. The proof is identical to that of Lemma 4. The only difference is that β_{n+k} transitions from $m^{(1)}_k \oplus \bigoplus_{j:\theta_{n+k,j}=0} z_{n+k,j}$ in $\mathbf{Hyb}_{5,k-1}$ to $0 \oplus \bigoplus_{j:\theta_{n+k,j}=0} z_{n+k,j}$ in $\mathbf{Hyb}_{5,k}$. Since the selector is 0, the index $n+k$ lies in the inactive second block. The auxiliary experiment $\mathsf{Expt}^{5,k}_{\mathcal{B},C}(\lambda,\theta,\beta)$, the three-step transition, and the reduction via Interactive Certified Everlasting Lemma all carry over verbatim.

Combining Lemmas 4–8 via the triangle inequality gives $\mathsf{TD}(\mathbf{Hyb}_0, \mathbf{Hyb}_{5,n}) \le \mathsf{negl}(\lambda)$, establishing Theorem 3.

Theorem 5 (Post-Verification Indistinguishability). *If $\prod^{\mathsf{PS}}_{\mathsf{RFE}}$ is a static-corruption secure public-slot registered functional encryption scheme and PRG is a secure pseudorandom generator, then Post-Verification Indistinguishability (Experiment (b) in Section 5.2) holds for Construction 5.3.*

Proof. The proof follows the same structure as [10], translated to the registered setting. One defines three intermediate hybrids $\mathbf{Hyb}^{\mathsf{C}}_0 \to \mathbf{Hyb}^{\mathsf{C}}_1 \to \mathbf{Hyb}^{\mathsf{C}}_2 \to \mathbf{Hyb}^{\mathsf{C}}_3$, where $\mathbf{Hyb}^{\mathsf{C}}_0$ and $\mathbf{Hyb}^{\mathsf{C}}_3$ correspond to Post-Verification Indistinguishability experiment set with bit $b=0$ and $b=1$ respectively. The transition $\mathbf{Hyb}^{\mathsf{C}}_0 \to \mathbf{Hyb}^{\mathsf{C}}_1$ replaces $v_{i,j,1\oplus z_{i,j}}$ with a uniformly random string for all $\theta_{i,j}=0$, which is indistinguishable by PRG security.

The replaced seeds are used nowhere else, since Vrfy reads U only where $\theta_{i,j} = 1$, so the reduction computes the verification outcome itself. The transition $\mathbf{Hyb}_1^{\mathsf{C}} \to \mathbf{Hyb}_2^{\mathsf{C}}$ switches the encrypted message from $m^{(0)}$ to $m^{(1)}$ and indistinguishability follows from the static-corruption security of $\prod_{\mathsf{RFE}}^{\mathsf{PS}}$. Indeed, except with negligible probability no replaced value lies in $\mathsf{Im}(\mathsf{PRG})$, so every public input passing step 2 of $h[f]$ has $b_{i,j} = z_{i,j}$ whenever $\theta_{i,j} = 0$. Thus the PRG programming ensures that $h[f_i^*]$ agrees on both plaintexts at every public input for every $i \in \mathcal{F}^*$, since $f_i^*(m^{(0)}) = f_i^*(m^{(1)})$. The reduction samples z, θ and U itself and simulates corruptions as in the proof of Theorem 3. The final transition $\mathbf{Hyb}_2^{\mathsf{C}} \to \mathbf{Hyb}_3^{\mathsf{C}}$ restores the PRG outputs, again by PRG security. Combining the three steps gives the Post-Verification Indistinguishability.

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. Bartusek, J., Khurana, D.: Cryptography with certified deletion. In: Annual International Cryptology Conference. pp. 192–223. Springer (2023)
2. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Theory of Cryptography Conference. pp. 253–273. Springer (2011)
3. Brakerski, Z., Döttling, N., Garg, S., Malavolta, G.: Factoring and pairings are not necessary for iO: Circular-secure LWE suffices. Cryptology ePrint Archive (2020)
4. Broadbent, A., Islam, R.: Quantum encryption with certified deletion. In: Theory of Cryptography Conference. pp. 92–122. Springer (2020)
5. Cini, V., Lai, R.W., Woo, I.K.: Lattice-based obfuscation from NTRU and Equivocal LWE. Cryptology ePrint Archive (2025)
6. Datta, P., Pal, T., Yamada, S.: Registered fe beyond predicates:(attribute-based) linear functions and more. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 65–104. Springer (2024)
7. Garg, S., Hajiabadi, M., Mahmoody, M., Rahimi, A.: Registration-based encryption: removing private-key generator from ibe. In: Theory of Cryptography Conference. pp. 689–718. Springer (2018)
8. 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)
9. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing **28**(4), 1364–1396 (1999)
10. Hiroka, T., Kitagawa, F., Morimae, T., Nishimaki, R., Pal, T., Yamakawa, T.: Certified everlasting secure collusion-resistant functional encryption, and more. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 434–456. Springer (2024)
11. Hohenberger, S., Lu, G., Waters, B., Wu, D.J.: Registered attribute-based encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 511–542. Springer (2023)

12. Murshid, S., Sarkar, R., Mandal, M.: Registered attribute-based encryption with publicly verifiable certified deletion, everlasting security, and more. arXiv preprint arXiv:2603.07646 (2026)
13. 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)
14. Sarkar, R.: Post-quantum secure non-committing registered functional encryption. Cryptology ePrint Archive, Paper 2026/1293 (2026), https://eprint.iacr.org/2026/1293
15. Unruh, D.: Revocable quantum timed-release encryption. Journal of the ACM (JACM) **62**(6), 1–76 (2015)
16. 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)
17. Yao, A.C.C.: How to generate and exchange secrets. In: 27th annual symposium on foundations of computer science (Sfcs 1986). pp. 162–167. IEEE (1986)