Black-Box Registered ABE from Lattices

Ziqi Zhu[1], Kai Zhang[2], Zhili Chen[1], Junqing Gong[1,3], and Haifeng Qian[1]

[1] East China Normal University, Shanghai, China
[2] Shanghai University of Electric Power, Shanghai, China
[3] Shanghai Qi Zhi Institute, Shanghai China

Abstract. This paper presents the first *black-box* registered ABE for circuit from lattices. The selective security is based on evasive LWE assumption [EUROCRYPT'22, CRYPTO'22]. The unique prior Reg-ABE scheme from lattices is derived from non-black-box construction based on function-binding hash and witness encryption [CRYPTO'23]. Technically, we first extend the black-box registration-based encryption from standard LWE [CRYPTO'23] so that we can register a public key with a function; this yields a LWE-based Reg-ABE with ciphertexts of size $L \cdot \mathsf{polylog}(L)$ where L is the number of users. We then make use of the special structure of its ciphertext to reduce its size to $\mathsf{polylog}(L)$ via an *algebraic* obfuscator based on evasive LWE [CRYPTO'24].

1 Introduction

Registered Attribute-Based Encryption (Reg-ABE) [25] is an authority-free variant of attribute-based encryption (ABE) [34,24]. In a Reg-ABE, a user generates public-secret key pair $(\mathsf{pk}, \mathsf{sk})$ on his own. A *curator*, who works *deterministically* and holds no secret, is responsible for registering pk along with a policy f to a *compact* master public key mpk. A ciphertext under mpk for attribute x can be decrypted using sk if $f(x) = 0$. Typically, the curator starts from a common reference string crs and sends each user a helper key hsk for decryption, which may be updated later. It is crucial that each user's helper key should not be updated too many times during the lifetime of the system.

Early Reg-ABE for identity check (i.e., registration-based encryption, RBE) relies on non-black-box technique based on garbling scheme or general obfuscation [19,20,23,9]. Recent pairing-based constructions [22,25,14,42,16,41,1,17] use black-box technique and support complex functionalities at the cost of large common reference string. Under lattice assumptions, we only see black-box construction for RBE [12,13] and non-black-box construction for Reg-ABE for circuits [15] via witness encryption [18,7,37] — *there is no black-box Reg-ABE (beyond RBE) from lattice!*

Result. In this work, we propose the first black-box construction for Reg-ABE for circuits. The selective security is based on evasive LWE assumption [39,36,37]. The scheme achieves the following profile:

$$|\mathsf{crs}| = \mathsf{polylog}(L), \quad |\mathsf{mpk}| = \mathsf{polylog}(L), \quad |\mathsf{ct}| = \mathsf{polylog}(L),$$

and the number of updates for each user is roughly $\mathsf{polylog}(L)$. We preserve the following advantages over pairing-based Reg-ABE constructions that has been achieved by [15,12,13]:

- Our scheme is unbounded, which means it supports an arbitrary number of users, with an implicit bound of $L = 2^\lambda$ (c.f. Remark 6.12 in [15]).
- Our scheme enjoys a transparent setup, i.e., the common reference string is simply a uniform random string.

Furthermore, our black-box technique allows us to achieve more:

- Our scheme is conceptually simpler and concretely more efficient than the non-black-box scheme in [15]. In particular, we avoid costly transformations between different computation models or languages.

– Our scheme enjoys user corruption in the standard model. In [15], their basic scheme in the standard model does not support corruption while a generic transformation that adds corruption to it relies on random oracles.

See Figure 1 for a detailed comparison. We clarify that both Reg-ABE in [15] and this work are *generic*: the former uses function-binding hash (FBH) and witness encryption while ours uses algebraic obfuscator for matrix PRF. Given the state of the art, FBH can be built from standard LWE [26,15] while the others rely on evasive LWE [7,37], respectively. We finally remark that the notion of Reg-ABE has been generalized to *Registered Functional Encryption* (Reg-FE) [10,5,41] but all known Reg-FE schemes do not subsume our result.

ref	**function**	**black-box**	**corruption**	**assumption**
[12,13]	identity check	✓	✓	LWE
FWW [15], §6	circuit	×	×	evasive LWE
FWW [15], §C	circuit	×	✓	evasive LWE + RO
Ours	circuit	✓	✓	evasive LWE

Fig. 1: Summary of Reg-ABE from Lattices. "RO" stands for "random oracle".

Implication & Discussion. By the generic transformation in [15], we immediately obtain a distributed broadcast encryption (DBE) that is unbounded and enjoys transparent setup from evasive LWE. A very recent work [6] described a DBE scheme with structured crs of size $L^2 \cdot \mathsf{polylog}(L)$ from ℓ-succinct LWE assumption [40] (ℓ depends on L). Compared with evasive LWE [39,36,37], ℓ-succinct LWE assumption is falsifiable and desirable [40]; we leave it as an open problem to build Reg-ABE (and thus DBE) that achieves unboundedness and transparent setup from falsifiable lattice assumptions such as ℓ-succinct LWE.

Strategy. Following the blueprint in [25], we consider a weaker primitive called *L-slotted* Reg-ABE. By this, we can work with a simpler scenario without one-by-one registration: public keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_L$ and associated functions $f_1, \ldots, f_L$ are given to the curator at a single time, it is then asked to produce mpk and helper keys $\mathsf{hsk}_1, \ldots, \mathsf{hsk}_L$. In this case, the curator is called *aggregator*. It is proved that slotted Reg-ABE implies (full-fledged) Reg-ABE via "power-of-two" technique [19,25] (c.f. Section B in **Appendix**). In the remaining of the Introduction, we focus on our slotted Reg-ABE for circuit with formal treatment in Section 3 and 4.

1.1 Warm-up: Zero-Slotted Reg-ABE

As a warm-up, we start with a weak primitive called *zero-slotted Reg-ABE* that does not involve any user: The aggregator simply embeds function f into mpk_f, a ciphertext under mpk_f for $\mathbf{x}$ should be *publicly* decryptable when $f(\mathbf{x}) = 0$. (The decryptor should know f and $\mathbf{x}$.) Security means that $\mathsf{ct}_\mathbf{x}$ hides the message when $f(\mathbf{x}) = 1$; here we assume $\mathbf{x}$ is claimed before crs is generated.

Homomorphic Computation. Let $n, q, \ell \in \mathbb{N}$ and $m = n \log q$. Let $\mathbf{F} \in \mathbb{Z}_q^{n \times m\ell}$ and $\mathbf{G} = \mathbf{I}_n \otimes (1, 2, 2^2, \ldots, 2^{\log q})$. For function $f : \{0,1\}^\ell \to \{0,1\}$ and input $\mathbf{x} \in \{0,1\}^\ell$, there exist two low-norm matrices $\mathbf{H}_{f,\mathbf{x}}$ and $\mathbf{H}_f$ such that

$$(\mathbf{F} - \mathbf{x}^\top \otimes \mathbf{G}) \cdot \mathbf{H}_{f,\mathbf{x}} = \mathbf{F}\mathbf{H}_f - f(\mathbf{x}) \cdot \mathbf{G}$$

In the literature, $\mathbf{A}$ is commonly used in place of $\mathbf{F}$; here we use $\mathbf{F}$ to indicate that this part is talking about "function". Note that $\mathbf{H}_f$ solely depends on f while $\mathbf{H}_{f,x}$ depends on both f and $\mathbf{x}$.

Zero-slotted Reg-ABE via Laconic Function Evaluation. We observe that the zero-slotted Reg-ABE is syntactically similar to *attribute-based laconic functional evaluation* (AB-LFE) [32] where server sends a digest digest_f of a function f, client encrypts the message under digest_f for $\mathbf{x}$, one can decrypt with the knowledge of f when $f(\mathbf{x}) = 0$. Adapting the AB-LFE scheme in [32] readily gives us an zero-slotted Reg-ABE:

$$\begin{aligned}\mathsf{crs} &= \mathbf{F}, \mathbf{v}\\ \mathsf{mpk}_f &= \mathbf{F}\mathbf{H}_f\mathbf{G}^{-1}(\mathbf{v})\\ \mathsf{ct}_\mathbf{x} &= \overbrace{\underset{\sim}{\mathbf{s}^\top(\mathbf{F} - \mathbf{x}^\top \otimes \mathbf{G})}}^{\mathbf{c}^\top}, \overbrace{\underset{\sim}{\mathbf{s}^\top\mathbf{F}\mathbf{H}_f\mathbf{G}^{-1}(\mathbf{v})} + \lfloor q/2 \rfloor \cdot \mathsf{m}}^{c}\end{aligned}$$

where $\mathbf{s}, \mathbf{v} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{F}, \mathbf{H}_f$ are defined as before and operation "$\underset{\sim}{\cdot}$" indicates a noised version of the input. Observe that

$$\overbrace{\mathbf{s}^\top\mathbf{F}\mathbf{H}_f\mathbf{G}^{-1}(\mathbf{v})}^{\approx c - \lfloor q/2 \rfloor \cdot \mathsf{m}} = \overbrace{\mathbf{s}^\top(\mathbf{F} - \mathbf{x}^\top \otimes \mathbf{G})}^{\approx \mathbf{c}^\top} \cdot \mathbf{H}_{f,\mathbf{x}}\mathbf{G}^{-1}(\mathbf{v}) + \mathbf{s}^\top\mathbf{v}.$$

We can see that:

- When $f(\mathbf{x}) = 0$, term $\mathbf{s}^\top\mathbf{v}$ disappears and decryption is straight-forward.
- When $f(\mathbf{x}) = 1$, we rewrite c using $\mathbf{c}$ and apply LWE assumption w.r.t. $(\mathbf{F}, \mathbf{v})$ after change of variable $\mathbf{F} \mapsto \mathbf{F} + \mathbf{x}^\top \otimes \mathbf{G}$. This proves that the ciphertext is pseudorandom in the *selective* setting.

A recent work [11] clarified that $\mathbf{F}\mathbf{H}_f$ can be seen a (functional) commitment to function f while $\mathbf{H}_{f,\mathbf{x}}$ can serve as a opening of f at the point $\mathbf{x}$.

1.2 One-slotted Scheme

Let us move from zero-slotted to one-slotted Reg-ABE where a user generates a key pair $(\mathsf{pk}, \mathsf{sk})$ and asks the aggregator to embed (pk, f) to $\mathsf{mpk}_{\mathsf{pk},f}$. A ciphertext $\mathsf{ct}_\mathbf{x}$ under $\mathsf{mpk}_{\mathsf{pk},f}$ for $\mathbf{x}$ can be decrypted by sk if $f(\mathbf{x}) = 0$. Security means that $\mathsf{ct}_\mathbf{x}$ hides the message in all the three settings:

- pk is maliciously generated by the adversary who may not even know sk;
- pk is honestly generated but sk is leaked to the adversary later;
- pk is honestly generated and sk keeps secret.

We call them *malicious*, *corrupted*, *honest* key/user, respectively, and require that $f(\mathbf{x}) = 1$ for the first two settings (to rule out trivial attacks).

Generating $(\mathsf{pk}, \mathsf{sk})$. We choose to use dual-Regev PKE [21] to generate $(\mathsf{pk}, \mathsf{sk})$. With $\mathbf{D} \leftarrow \mathbb{Z}_q^{n \times m}$ in crs, we define

$$\mathsf{sk} = \mathbf{k} \leftarrow \{0,1\}^m \quad \text{and} \quad \mathsf{pk} = \mathbf{u} = \mathbf{D}\mathbf{k}. \tag{1}$$

It is helpful to recall that a ciphertext is $(\underset{\sim}{\mathbf{s}^\top\mathbf{D}}, \underset{\sim}{\mathbf{s}^\top\mathbf{u}} + \lfloor q/2 \rfloor \cdot \mathsf{m})$ and decryption relies on the equation: $\mathbf{s}^\top\mathbf{D} \cdot \mathbf{k} = \mathbf{s}^\top\mathbf{u}$.

Binding pk and f. We bind f with $\mathsf{pk} = \mathbf{u}$ via the following commitment:

$$\mathbf{h} = \mathbf{F}\mathbf{H}_f\mathbf{G}^{-1}(\mathbf{v}) + \mathbf{P}\mathbf{G}^{-1}(\mathbf{u}) \tag{2}$$

where $\mathbf{F}\mathbf{H}_f\mathbf{G}^{-1}(\mathbf{v})$ is mpk_f from zero-slotted scheme and $\mathbf{P} \leftarrow \mathbb{Z}_q^{n \times m}$ (we use letter $\mathbf{P}$ to indicate that the latter part is talking about "public key"). Inspired by [12], this readily gives a mpk for (pk, f) along with $\mathbf{F}, \mathbf{P}, \mathbf{D}$ and a ciphertext for $\mathbf{x} \in \{0,1\}^\ell$ is basically a dual-Regev PKE ciphertext under public key

$$\mathbf{M}_\mathbf{x} = \begin{pmatrix} \mathbf{F} - \mathbf{x}^\top \otimes \mathbf{G} & \mathbf{P} \\ & \mathbf{G} \ \mathbf{D} \end{pmatrix}, \begin{pmatrix} \mathbf{h} \\ \mathbf{0} \end{pmatrix} \tag{3}$$

with trapdoor $\pi_{\mathbf{u},f} = (\mathbf{H}_f\mathbf{G}^{-1}(\mathbf{v}), \mathbf{G}^{-1}(\mathbf{u}), -\mathbf{k})$. In more details, our one-slotted Reg-ABE scheme is as follows:

$$\begin{aligned}
\mathsf{crs} &= \mathbf{F}, \mathbf{P}, \mathbf{D}, \mathbf{v} \\
(\mathsf{pk}, \mathsf{sk}) &= (\mathbf{u} = \mathbf{D}\mathbf{k}, \mathbf{k}) \\
\mathsf{mpk}_{\mathsf{pk},f} &= \mathbf{F}\mathbf{H}_f\mathbf{G}^{-1}(\mathbf{v}) + \mathbf{P}\mathbf{G}^{-1}(\mathbf{u}) \\
\mathsf{ct}_{\mathbf{x}} &= \overbrace{\mathbf{s}^{\top}(\mathbf{F} - \mathbf{x}^{\top} \otimes \mathbf{G})}^{\mathbf{c}_1^{\top}}, \overbrace{\mathbf{s}^{\top}\mathbf{P} + \mathbf{t}^{\top}\mathbf{G}}^{\mathbf{c}_2^{\top}}, \overbrace{\mathbf{t}^{\top}\mathbf{D}}^{\mathbf{c}_3^{\top}}, \overbrace{\mathbf{s}^{\top}\mathbf{h} + \lfloor q/2 \rfloor \cdot \mathsf{m}}^{c}.
\end{aligned}$$

where $\mathbf{s}^{\top}, \mathbf{t}^{\top} \leftarrow \mathbb{Z}_q^n$. When $f(\mathbf{x}) = 0$, decryption relies on:

$$\begin{aligned}
&\overbrace{\mathbf{s}^{\top}(\mathbf{F} - \mathbf{x}^{\top} \otimes \mathbf{G})}^{\approx \mathbf{c}_1^{\top}} \cdot \mathbf{H}_{f,\mathbf{x}}\mathbf{G}^{-1}(\mathbf{v}) + \overbrace{(\mathbf{s}^{\top}\mathbf{P} + \mathbf{t}^{\top}\mathbf{G})}^{\approx \mathbf{c}_2^{\top}} \cdot \mathbf{G}^{-1}(\mathbf{u}) - \overbrace{\mathbf{t}^{\top}\mathbf{D}}^{\approx \mathbf{c}_3^{\top}} \cdot \overbrace{\mathbf{k}}^{\mathsf{sk}} \\
&= \mathbf{s}^{\top}\mathbf{F}\mathbf{H}_f\mathbf{G}^{-1}(\mathbf{v}) + (\mathbf{s}^{\top}\mathbf{P}\mathbf{G}^{-1}(\mathbf{u}) + \mathbf{t}^{\top}\mathbf{u}) - \mathbf{t}^{\top}\mathbf{u} \\
&= \mathbf{s}^{\top}\underbrace{(\mathbf{F}\mathbf{H}_f\mathbf{G}^{-1}(\mathbf{v}) + \mathbf{P}\mathbf{G}^{-1}(\mathbf{u}))}_{\mathbf{h}} \approx c - \lfloor q/2 \rfloor \cdot m.
\end{aligned}$$

This is quite similar to the 1-key ABE described in [8] implicitly used in [29].

Security. For security, analogous to the analysis of correctness, we approximately rewrite $c - \lfloor q/2 \rfloor \cdot \mathsf{m}$ as:

$$\overbrace{\mathbf{s}^{\top}(\mathbf{F} - \mathbf{x}^{\top} \otimes \mathbf{G})}^{\approx \mathbf{c}_1^{\top}} \cdot \mathbf{H}_{f,\mathbf{x}}\mathbf{G}^{-1}(\mathbf{v}) + f(\mathbf{x}) \cdot \mathbf{s}^{\top}\mathbf{v} + \overbrace{(\mathbf{s}^{\top}\mathbf{P} + \mathbf{t}^{\top}\mathbf{G})}^{\approx \mathbf{c}_2^{\top}} \cdot \mathbf{G}^{-1}(\mathbf{u}) - \mathbf{t}^{\top}\mathbf{u}$$

After a change of variable $\mathbf{F} \mapsto \mathbf{F} + \mathbf{x}^{\top} \otimes \mathbf{G}$, we consider two cases:

- When $f(\mathbf{x}) = 1$, the selective security is analogous to the zero-slotted scheme under the LWE assumption w.r.t. $(\mathbf{F}, \mathbf{P}, \mathbf{v})$.
- When $f(\mathbf{x}) = 0$, the above argument does not work since the term $\mathbf{s}^{\top}\mathbf{v}$ does not appear. Fortunately, in this case, $\mathsf{pk} = \mathbf{u}$ must be honest and thus $\mathbf{k} \leftarrow \{0,1\}^m$ is secret from the adversary. The proof makes use of the entropy from $\mathbf{k}$ and consists of four steps:
  1. LWE assumption w.r.t. $(\mathbf{F}, \mathbf{P})$ ensures that $\mathbf{c}_1$ and $\mathbf{c}_2$ are pseudorandom.
  2. Rewrite term $\mathbf{t}^{\top}\mathbf{u}$ in the above expression as $\mathbf{c}_3^{\top} \cdot \mathbf{k}$; this uses the fact that $\mathbf{k}$ is always known to the simulator; here noise flooding is needed.
  3. LWE assumption w.r.t. $\mathbf{D}$ ensures that $\mathbf{c}_3$ are pseudorandom.
  4. Leftover hash lemma ensures that $(\mathbf{D}, \mathbf{c}_3^{\top}, \mathbf{D}\mathbf{k}, \mathbf{c}_3^{\top}\mathbf{k}) \approx (\mathbf{D}, \mathbf{c}_3^{\top}, \$, \$)$ when $\mathbf{D}, \mathbf{c}_3$ are random over $\mathbb{Z}_q$ and $\mathbf{k}$ are random over $\{0, 1\}$.

We highlight that the proof does not require the adversary to claim whether the user will be honest in advance.

1.3 From One-slotted to L-slotted Reg-ABE

This section explains how to build L-slotted scheme from one-slotted scheme. We illustrate the idea via an example of $L = 8$ and set $D = \log L = 3$. Namely, we have $(\mathsf{sk}_1 = \mathbf{k}_1, \mathsf{pk}_1 = \mathbf{u}_1 = \mathbf{D}\mathbf{k}_1), \ldots, (\mathsf{sk}_8 = \mathbf{k}_8, \mathsf{pk}_8 = \mathbf{u}_8 = \mathbf{D}\mathbf{k}_8)$ defined as in (1) and ask the aggregator to bind them with $f_1, \ldots, f_8$, respectively.

Aggregation via Merkle Hash. We start with the paradigm presented in [12]. We place the 8 users at the leaves of Merkle tree of depth $D = 3$ and compute the Merkle hash. In contrast to [12], we put $\mathbf{h}_1, \ldots, \mathbf{h}_8$ defined as in (2) at the leaves rather than $(\mathsf{pk}_1 = \mathbf{u}_1, \ldots, \mathsf{pk}_8 = \mathbf{u}_8)$. In more details, let $\mathbf{B}_0, \mathbf{B}_1$ be the key for hash function and compute

$$\mathbf{h}_i = \mathbf{F}\mathbf{H}_{f_i}\mathbf{G}^{-1}(\mathbf{v}) + \mathbf{P}\mathbf{G}^{-1}(\mathbf{u}_i), \quad \forall i \in \{0,1\}^3$$

where we write index i of slot in its binary form $i = (i_1, i_2, i_3)$; we compute

- $\mathbf{h}_{b_1,b_2} = \mathbf{B}_0\mathbf{G}^{-1}(\mathbf{h}_{b_1,b_2,0}) + \mathbf{B}_1\mathbf{G}^{-1}(\mathbf{h}_{b_1,b_2,1})$ for all $(b_1, b_2) \in \{0,1\}^2$;
- $\mathbf{h}_b = \mathbf{B}_0\mathbf{G}^{-1}(\mathbf{h}_{b,0}) + \mathbf{B}_1\mathbf{G}^{-1}(\mathbf{h}_{b,1})$ for all $b \in \{0,1\}$;
- $\mathbf{h}_\epsilon = \mathbf{B}_0\mathbf{G}^{-1}(\mathbf{h}_0) + \mathbf{B}_1\mathbf{G}^{-1}(\mathbf{h}_1)$.

We take $\mathbf{h}_\epsilon$ as mpk and, for all $i = (i_1, i_2, i_3) \in \{0,1\}^3$ and set:

$$\mathsf{hsk}_i = \pi_i = (\mathbf{G}^{-1}(\mathbf{h}_{i_1,i_2,b}), \mathbf{G}^{-1}(\mathbf{h}_{i_1,b}), \mathbf{G}^{-1}(\mathbf{h}_b))_{b\in\{0,1\}}.$$

As [12], we can build a ciphertext under $\mathbf{h}_\epsilon$ as follows: for $i \in \{0,1\}^3$ and $\mathbf{x} \in \{0,1\}^\ell$, it is a dual-Regev PKE ciphertext under the following public key

$$\overbrace{\begin{pmatrix} \mathbf{B}_0 & \mathbf{B}_1 & & & & & & \\ \bar{i}_1 \cdot \mathbf{G} & i_1 \cdot \mathbf{G} & \mathbf{B}_0 & \mathbf{B}_1 & & & & \\ & & \bar{i}_2 \cdot \mathbf{G} & i_2 \cdot \mathbf{G} & \mathbf{B}_0 & \mathbf{B}_1 & & \\ & & & & \bar{i}_3 \cdot \mathbf{G} & i_3 \cdot \mathbf{G} & \mathbf{F} - \mathbf{x}^\top \otimes \mathbf{G} & \mathbf{P} \\ & & & & & & \mathbf{G} & \mathbf{D} \end{pmatrix}}^{\mathbf{M}_{i,\mathbf{x}}}, \begin{pmatrix} \mathbf{h}_\epsilon \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}$$

with trapdoor $(\pi_i, \pi_{\mathbf{u}_i,f_i})$. Observe that this is an extension of our one-slotted scheme; the right-bottom corner of $\mathbf{M}_{i,\mathbf{x}}$ is identical to $\mathbf{M}_\mathbf{x}$ in (3). Both correctness and selective security are natural extensions of one-slotted scheme. As a summary, our eight-slotted Reg-ABE scheme is as follows:

$$\begin{aligned} \mathsf{crs} &= \mathbf{B}_0, \mathbf{B}_1, \mathbf{F}, \mathbf{P}, \mathbf{D}, \mathbf{v} \\ (\mathsf{pk}_i, \mathsf{sk}_i) &= (\mathbf{u}_i = \mathbf{D}\mathbf{k}_i, \mathbf{k}_i) \\ \mathsf{mpk}_{\mathsf{pk},f} &= \mathbf{h}_\epsilon \quad \text{// defined as above} \\ \mathsf{ct}_{i,\mathbf{x}} &= \boxed{\{\mathbf{s}_{j-1}^\top\mathbf{B}_0 + \bar{i}_j \cdot \mathbf{s}_j^\top\mathbf{G}, \mathbf{s}_{j-1}^\top\mathbf{B}_1 + i_j \cdot \mathbf{s}_j^\top\mathbf{G}\}_{j=1,2,3}} \\ &\quad \mathbf{s}_3^\top(\mathbf{F} - \mathbf{x}^\top \otimes \mathbf{G}), \mathbf{s}_3^\top\mathbf{P} + \mathbf{t}^\top\mathbf{G}, \mathbf{t}^\top\mathbf{D}, \mathbf{s}_0^\top\mathbf{h} + \lfloor q/2 \rfloor \cdot \mathsf{m} \end{aligned}$$

where $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{t} \leftarrow \mathbb{Z}_q^n$. Compared with our one-slotted Reg-ABE, the boxed terms are added to handle Merkle Hash and random coins are modified accordingly (see the gray boxes).

Issue & Large Ciphertexts. Unfortunately, this is *not* a slotted Reg-ABE:

- The above ciphertext depends on $i = (i_1, i_2, i_3)$, which is basically built for a *single* user by identifying the corresponding path. This is inherited from RBE scheme in [12] where a ciphertext targets to *exactly one* user.
- A ciphertext in L-slotted Reg-ABE is potentially decryptable by *all* L users.

A naive fix is to encrypt to all eight users: a ciphertext $\mathsf{ct}_\mathbf{x}$ for $\mathbf{x}$ consists of eight sub-ciphertexts $\mathsf{ct}_{000,\mathbf{x}}, \mathsf{ct}_{001,\mathbf{x}}, \ldots, \mathsf{ct}_{111,\mathbf{x}}$ each defined as above. Both decryption and selective security are easy to established. However this makes ciphertext size unacceptable in the setting of Reg-ABE — for general L, the ciphertext size is as large as $L \cdot \mathsf{polylog}(L)$ where $\mathsf{polylog}(L)$ is the size of each sub-cihpertext and factor L comes from "encrypting to all users".

1.4 Shaving Factor L Off

To get an acceptable Reg-ABE scheme, we want to find out a compact representation of $\mathsf{ct}_{\mathbf{x}}$. In this overview, let us focus on the first term in each sub-cipertext $\mathsf{ct}_{i,\mathbf{x}}$ that corresponds to the first column of $\mathbf{M}_{i,\mathbf{x}}$:

$$\underbrace{\mathbf{s}_i^\top \mathbf{B}_0 + \bar{i}_1 \cdot \mathbf{t}_i^\top \mathbf{G}}, \quad \forall i \in \{0,1\}^3$$

where $\mathbf{s}_i$ and $\mathbf{t}_i$ are parts of randomness of each sub-ciphertext.

Idea 1: Generating Random Coins via PRF. Since all $(\mathbf{s}_i, \mathbf{t}_i)$ are fresh, it seems impossible to compress the 8 terms from an information-theoretical point of view. Our first idea is to use *correlated* randomness. Conceptually, we employ PRF in [4,7], denoted by F, to generate those random coins: for $i \in \{0,1\}^3$, set

$$\mathbf{s}_i^\top = \mathsf{F}(\mathbf{K}_s, i) = \mathbf{t}^\top \mathbf{S}_{i_1} \mathbf{S}_{i_2} \cdots \mathbf{S}_{i_3} \mathbf{K}_s$$
$$\mathbf{t}_i^\top = \mathsf{F}(\mathbf{K}_t, i) = \mathbf{t}^\top \mathbf{S}_{i_1} \mathbf{S}_{i_2} \cdots \mathbf{S}_{i_3} \mathbf{K}_t$$

where $\mathbf{t}$ is some fixed low-norm vector, $\mathbf{S}_0$ and $\mathbf{S}_1$ are low-norm and form the definition of PRF, and $\mathbf{K}_s$ and $\mathbf{K}_t$ are key or seed of PRF. It is proved that $\{\mathbf{s}_i, \mathbf{t}_i\}$ are pseudorandom under (low-norm) LWE assumption [4]. Then, we can write the eight terms as below:

$$\underbrace{\mathbf{t}^\top \mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \mathbf{K}_s \mathbf{B}_0 + \bar{i}_1 \mathbf{t}^\top \mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \mathbf{K}_t \mathbf{G}}, \quad \forall i \in \{0,1\}^3.$$

Idea 2: Fixing the Proof via Crossing Lemma. In the proof, we will need to argue that

$$\{\mathbf{t}^\top \mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \mathbf{K}_s \mathbf{B}_0 + \mathbf{e}_i^\top\}_{i \in \{0,1\}^3}$$

are pseudorandom where we write noise terms $\mathbf{e}_i$ explicitly. Clearly, we shall use the pseudorandomness of PRF. For this, one may want to change those terms into

$$(\overbrace{\mathbf{t}^\top \mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \mathbf{K}_s + \widetilde{\mathbf{e}}_i^\top}^{\mathsf{F}(\mathbf{K}_s, i)}) \cdot \mathbf{B}_0 + \mathbf{e}_i^\top, \quad \forall i \in \{0,1\}^3$$

where $\widetilde{\mathbf{e}}_i^\top$ are noise terms used by the PRF and apply the pseudorandomness of PRF. Unfortunately, the first step is incorrect: due to the large norm of $\mathbf{B}_0$, the noise flooding technique does not work as is. We circumvent the issue with the following proof strategy:

$$\begin{aligned} &\{\mathbf{t}^\top \mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \mathbf{K}_s \mathbf{B}_0 + \mathbf{e}_i^\top\}_{i \in \{0,1\}^3} \\ \approx_s\ &\{\mathbf{t}^\top \mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} (\mathbf{K}_s \mathbf{B}_0 + \boxed{\mathbf{E}}) + \mathbf{e}_i^\top\}_{i \in \{0,1\}^3} \\ \approx_c\ &\{\mathbf{t}^\top \mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \boxed{\widetilde{\mathbf{K}}_s} + \mathbf{e}_i^\top\}_{i \in \{0,1\}^3} \approx_c \{\$\}_{i \in \{0,1\}^3} \end{aligned}$$

where $\mathbf{E}$ is noise term and $\tilde{\mathbf{K}}_s$ is uniformly random. We highlight that the first step can be ensured by the noise flooding technique due to the fact that $\mathbf{t}, \mathbf{S}_0, \mathbf{S}_1$ all have low norm. The remaining steps are standard: the second step uses LWE assumption w.r.t. $\mathbf{B}_0$ and the last step applys the pseudorandomness of PRF but with new key $\widetilde{\mathbf{K}}_s$. We establish *crossing lemma* (Lemma 2) to capture the idea and present our slotted Reg-ABE with large ciphertext in Section 3.

Idea 3: Decomposing & Obfuscating. We then rewrite those terms as:

$$\underbrace{(\mathbf{t}^\top, \mathbf{t}^\top) \overbrace{\begin{pmatrix} \mathbf{S}_{i_1} & \\ & \bar{i}_1 \cdot \mathbf{S}_{i_1} \end{pmatrix}}^{\hat{\mathbf{s}}_{1,i_1}} \overbrace{\begin{pmatrix} \mathbf{S}_{i_2} & \\ & \mathbf{S}_{i_2} \end{pmatrix}}^{\hat{\mathbf{S}}_{2,i_2}} \overbrace{\begin{pmatrix} \mathbf{S}_{i_3} & \\ & \mathbf{S}_{i_3} \end{pmatrix} \begin{pmatrix} \mathbf{K}_s \mathbf{B}_0 \\ \mathbf{K}_t \mathbf{G} \end{pmatrix}}^{\hat{\mathbf{S}}_{3,i_3}}}, \quad \forall i \in \{0,1\}^3$$

and, more importantly, the eight terms can be assembled from six matrices $\hat{\mathbf{S}}_{1,b}, \hat{\mathbf{S}}_{2,b}, \hat{\mathbf{S}}_{3,b}$ with $b \in \{0,1\}$. In general, L terms can be built from $2\log L$ blocks. This almost reaches our goal; however, we quickly argue that it is insecure to publish $\widehat{\mathbf{S}}_{j,b}$ with $j \in [3]$ and $b \in \{0,1\}$ "in the clear". To ensure that only those eight terms can be derived, we employ GGH encoding: sample $\mathbf{A}_1, \mathbf{A}_2$ with trapdoors $\mathbf{A}_1^{-1}$ and $\mathbf{A}_2^{-1}$, we publish the following terms instead:

$$\hat{\mathbf{S}}_{1,b}\mathbf{A}_1, \quad \mathbf{A}_1^{-1}(\hat{\mathbf{S}}_{2,b}\mathbf{A}_2), \quad \mathbf{A}_2^{-1}(\hat{\mathbf{S}}_{3,b}), \quad \forall b \in \{0,1\}.$$

the security is then based on evasive LWE as in [37]. Formally, we employ a recent algebraic obfuscator for relaxed matrix PRF proposed by [30] in the entirely different context. In their work, a concrete obfuscator Obf was presented based on evasive LWE assumption [37]; the construction is a natural extension of witness encryption in [37] which is based on GGH encoding. This will yield a slotted Reg-ABE basically identical to the above. Our treatment gives a generic Reg-ABE and we hope this is easier to follow. We leave a more detailed overview and our final slotted Reg-ABE in Section 4.

2 Preliminaries

Notations. For a finite set S, we use $s \leftarrow S$ to denote the procedure of sampling s from S uniformly. We use $y \leftarrow \mathsf{Alg}(x; r)$ to denote the procedure of running algorithm Alg on input x with random coin r and assigning the output to y. When r is irrelevant to the question, we omit it and view $\mathsf{Alg}(x)$ as a distribution. We use $[\mathsf{Alg}]$ to denote its support, i.e., the set of all possible outputs y. We use lower-case boldface to denote *column* vectors (e.g., $\mathbf{a}$) and upper-case boldface to denote matrices (e.g. $\mathbf{M}$). For $\mathbf{A}_1, \ldots, \mathbf{A}_n$, we use $\mathsf{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_n)$ to denote a matrix with $\mathbf{A}_1, \ldots, \mathbf{A}_n$ on its diagonal. For $n \in \mathbb{N}$, we use $\{0,1\}^n$ to denote the set of all binary strings of length n and define $\{0,1\}^0 = \{\epsilon\}$. For $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ and $0 \le j \le n$, define $\mathbf{x}_{|j} = (x_1, \ldots, x_j)$ with $\mathbf{x}_{|0} = \epsilon$ and write $\mathbf{x}\|b = (x_1, \ldots, x_n, b)$.

2.1 Lattice Background

Norm. Let $n, m, q \in \mathbb{N}$. For any matrix $\mathbf{A} = (a_{i,j})_{i \in [n], j \in [m]} \in \mathbb{Z}_q^{n \times m}$, we define $\|\mathbf{A}\| = \max_{i \in [n]} \sum_{j=1}^m |a_{i,j}|$, which is the *infinity norm*[4]. In particular, for row vector $\mathbf{r}^\top = (r_1, \ldots, r_m) \in \mathbb{Z}_q^{1 \times m}$, we have $\|\mathbf{r}^\top\| = \sum_{j=1}^m |r_j|$; for column vector $\mathbf{c} = (c_1, \ldots, c_n)^\top \in \mathbb{Z}_q^n$, we have $\|\mathbf{c}\| = \max_{i \in [n]} |c_i|$. For $c \in \mathbb{Z}_q$ and matrices $\mathbf{A}, \mathbf{B}$ of proper sizes, we have (1) $\|c \cdot \mathbf{A}\| = |c| \cdot \|\mathbf{A}\|$; (2) $\|\mathbf{A} + \mathbf{B}\| \le \|\mathbf{A}\| + \|\mathbf{B}\|$; (3) $\|\mathbf{AB}\| \le \|\mathbf{A}\| \cdot \|\mathbf{B}\|$.

Leftover Hash Lemma (LHL). Let $n, m, q \in \mathbb{N}$ with $m \ge 2n \log q$. We have

$$\{(\mathbf{A}, \boxed{\mathbf{Ax}}) : \mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{x} \leftarrow \{0,1\}^m\} \approx_s \{(\mathbf{A}, \mathbf{u}) : \mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{u} \leftarrow \mathbb{Z}_q^n\}.$$

Discrete Gaussians and Facts. Let $\mathcal{D}_{\mathbb{Z},\sigma}$ denote the *discrete Gaussian distribution* over $\mathbb{Z}$ with parameter $\sigma > 0$. The *Gaussian Tail Bound* [31] says that: For any $\lambda \in \mathbb{N}$, we have

$$\Pr[\|x\| > \sqrt{\lambda}\sigma : x \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}] \le 2^{-\lambda}$$

We will use *noise flooding* based on the following fact: For any $|z| \le B$, we have

$$\{x : x \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}\} \approx_s \{x + z : x \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}\} \quad \text{where} \quad \sigma = B\lambda^{\omega(1)}.$$

This is also called *smudging lemma* in the literature.

[4] The standard notation for infinity norm is $\|\cdot\|_\infty$; we omit the transcript for brevity.

Learning with Error (LWE). Let $n, m, q, \sigma \in \mathbb{N}$. The *learning with error* (LWE) assumption $\mathsf{LWE}_{n,m,q,\sigma}$ [33] says that: For all P.P.T. $\mathcal{A}$,

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{LWE}}(n) = |\Pr[\mathcal{A}(\mathbf{A}, \boxed{\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{c}^\top) = 1]| = \varepsilon(n)$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$ and $\mathbf{c} \leftarrow \mathbb{Z}_q^m$. The LWE assumption with sub-exponential hardness means: $\mathcal{A}$ is allowed to run in time 2^{n^c} and advantage can be bounded by 2^{-n^c} when $q/\sigma \leq 2^{n^c}$ for some constant $c > 0$.

Gadget Matrix. Let $n, q \in \mathbb{N}$ such that $q \geq 2$ and $m = n\lceil\log q\rceil$. Define $\mathbf{G} = \mathbf{I}_n \otimes (1, 2, \ldots, 2^{\lceil\log q\rceil - 1}) \in \mathbb{Z}_q^{n\times m}$. For any $\mathbf{z} = (z_1, \ldots, z_n)^\top \in \mathbb{Z}_q^n$, we use $\mathbf{G}^{-1}(\mathbf{z})$ to denote $(\mathsf{bin}(z_1), \ldots, \mathsf{bin}(z_n))^\top$ where $\mathsf{bin}(\cdot)$ gives the binary representation of the input. This ensures that $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{z}) = \mathbf{z}$.

Homomorphic Evaluation. Let $\ell, d, s \in \mathbb{N}$, we use $\mathcal{C}_{d,\ell}$ to denote the family of boolean circuits of depth d and input size ℓ. Let $n, q \in \mathbb{N}$ with $q \geq 2$ and $m > 2n\log q$. There exists two deterministic algorithms [3,35]:

$$\mathsf{EvalF}(\mathbf{B}, f) \to \mathbf{H}_f; \quad \mathsf{EvalFX}(\mathbf{B}, f, \mathbf{x}) \to \mathbf{H}_{f,\mathbf{x}}$$

where $\mathbf{B} \in \mathbb{Z}_q^{n\times m\ell}$, $f \in \mathcal{C}_{d,\ell}$, $\mathbf{x} \in \{0,1\}^\ell$ and $\mathbf{H}_f, \mathbf{H}_{f,\mathbf{x}} \in \mathbb{Z}^{m\ell\times m}$ such that

$$(\mathbf{B} - \mathbf{x} \otimes \mathbf{G})\mathbf{H}_{f,\mathbf{x}} = \mathbf{B}\mathbf{H}_f - f(\mathbf{x}) \cdot \mathbf{G} \quad \text{and} \quad \|\mathbf{H}_{f,\mathbf{x}}\| \leq m^{O(d)}. \tag{4}$$

2.2 Relaxed Matrix Pseudorandom Function

We review relaxed pseudorandom function (PRF) [30] with adaptation.

Algorithms. Let $q \in \mathbb{N}$. A *σ-matrix pseudorandom function (σ-mPRF) family* PRF consists of a tuple of P.P.T. algorithms with the following syntax:

	$w \in \mathbb{N}$	: width of PRF
	$\chi > 0$	: noise parameter
	$v \in \mathbb{N}$	: length of PRF
$\mathsf{PRFGen}(w, \chi) \to \mathbf{S}$	$m \in \mathbb{N}$	: length of PRF output
$\mathsf{PRFKey}(w, m, \mathsf{par}) \to \mathbf{K}$	$\mathsf{par} \in \{0,1\}^*$	: parameters for key
$\mathsf{PRFEval}(\mathbf{S}, \mathbf{K}, \mathbf{x}) = \mathbf{m}^\top \prod_{i=1}^{\ell} \mathbf{M}_{i,x_i}\mathbf{K}$	$\mathbf{S} \in \mathbb{Z}^{w\times v}$	: description of PRF
	$\mathbf{K} \in \mathbb{Z}_q^{w\times m}$	: key of PRF
	$\ell \in \mathbb{N}$	: length of input
	$\mathbf{x} \in \{0,1\}^\ell$	: input of PRF

where $\mathbf{m} \in \mathbb{Z}^w$ and $\mathbf{M}_{i,b} \in \mathbb{Z}^{w\times w}$ for all $i \in [\ell], b \in \{0,1\}$ can publicly and deterministically computed from $\mathbf{S}$, and the following *σ-pseudorandomness* [30]: if for all $\mathsf{par} \in \{0,1\}^*$ and all P.P.T. $\mathcal{A}$,

$$|\Pr[\mathcal{A}^{\mathsf{O}(\mathbf{S},\mathbf{K},\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{A}^{\mathsf{TRF}(\cdot)}(1^\lambda) = 1]| = \varepsilon(\lambda)$$

where $\mathbf{S} \leftarrow \mathsf{PRFGen}(w, \sigma)$, $\boxed{\mathbf{K} \leftarrow \mathsf{PRFKey}(w, m, \mathsf{par})}$, oracle $\mathsf{O}(\mathbf{S}, \mathbf{K}, x)$ outputs

$$\mathsf{PRFEval}(\mathbf{S}, \mathbf{K}, \mathbf{x}) + \mathbf{e}_\mathbf{x}^\top, \quad \mathbf{e}_\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m, \quad \forall \mathbf{x} \in \{0,1\}^\ell,$$

and TRF refers to a truly random function. We will consider a stronger version where $\mathcal{A}$ additionally gets aux related to $\mathbf{K}$; this is captured by replacing boxed part with $(\mathbf{K}, \mathsf{aux}) \leftarrow \mathsf{PRFKey}^*(\mathsf{par}^*)$ and sending aux to $\mathcal{A}$. Here, $\mathbf{K}$ produced by the two algorithms should have the same distribution. In this work, we focus on the setting where $\ell = \log(\lambda)$ and $\mathcal{A}$ is allowed to see evaluations at all $\mathbf{x}$.

Norm. Let $w, \ell \in \mathbb{N}$ and $\sigma > 0$. The *norm* of σ-mPRF PRF = (PRFGen, PRFKey, PRFEval) on input of length ℓ is defined as:

$$\max_{\mathbf{S}\in[\mathsf{PRFGen}(w,\sigma)],\mathbf{x}\in\{0,1\}^\ell} \|\mathsf{PRFEval}(\mathbf{S}, \mathbf{I}_w, \mathbf{x})\|$$

Note that this is independent of key $\mathbf{K}$ and parameter m as well.

Construction from LWE. The following lemma gives a LWE-based σ-mPRF.

Lemma 1 ([2,4,38]). *Let $n, q, w, \ell \in \mathbb{N}, \chi, \sigma > 0$ and*

$$w = 6n \log q,\ \chi = \Omega(\sqrt{n \log q}),\ \sigma \geq \lambda^{\ell+\omega(1)} \cdot (w\chi)^\ell.$$

Under $\mathsf{LWE}_{n,\mathrm{poly}(w,\ell,2^\ell),q,\chi}$ *assumption, for all P.P.T. $\mathcal{A}$,*

$$\begin{aligned}&\Pr[\mathcal{A}(\mathbf{M}_0, \mathbf{M}_1, \mathbf{t}, \boxed{\{\mathbf{t}^\top \mathbf{M}_{x_1} \cdots \mathbf{M}_{x_\ell} \mathbf{K} + \mathbf{e}_\mathbf{x}^\top\}_{\mathbf{x}\in\{0,1\}^\ell}}) = 1] \\ &- \Pr[\mathcal{A}(\mathbf{M}_0, \mathbf{M}_1, \mathbf{t}, \{\mathbf{u}_\mathbf{x}^\top\}_{\mathbf{x}\in\{0,1\}^\ell}) = 1] = \varepsilon(n)\end{aligned}$$

where $\mathbf{M}_0, \mathbf{M}_1 \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{w\times w}$, $\mathbf{K} \leftarrow \mathbb{Z}_q^{w\times m}$, $\mathbf{e}_\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$ and $\mathbf{t} \in \mathbb{Z}_q^w$ being the first elementary basis vector.

In particular, fix $\mathbf{t}$ defined in Lemma 1, the σ-mPRF PRF_0 works as follows:

- $\mathsf{PRFGen}_0(w, \chi)$: Output $\mathbf{S} = (\mathbf{M}_0|\mathbf{M}_1) \in \mathbb{Z}^{w\times 2w}$.
- $\mathsf{PRFKey}_0(w, m)$: Output $\mathbf{K} \leftarrow \mathbb{Z}_q^{w\times m}$.
- $\mathsf{PRFEval}_0(\mathbf{S}, \mathbf{K}, \mathbf{x}) = \mathbf{t}^\top \cdot \mathbf{M}_{x_1} \cdots \mathbf{M}_{x_\ell} \cdot \mathbf{K}$.

and its norm on input of length ℓ is bounded by $(w\lambda\chi)^\ell$.

2.3 Obfuscation for Matrix PRF

Let $\lambda, \ell, m, q \in \mathbb{N}$. An obfuscator for σ-mPRF (PRFGen, PRFKey, PRFEval) is a P.P.T. algorithm with the following syntax:

$$\mathsf{Obf}(1^\lambda, 1^\ell, \mathbf{S}, \mathbf{K}) \to \tilde{\mathsf{F}} \quad \left| \quad \begin{aligned} &\lambda : \text{security parameter} \\ &\ell : \text{input length} \\ &\mathbf{S} : \text{description of PRF} \\ &\mathbf{K} : \text{key of PRF} \\ &\tilde{\mathsf{F}} : \text{obfuscated PRF}\end{aligned}\right.$$

and following two properties:

- Δ-Correctness: For all $w, m, \ell \in \mathbb{N}$, all $\chi > 0$, all par $\in \{0,1\}^*$, all $\mathbf{S} \in [\mathsf{PRFGen}(w, \chi)]$, all $\mathbf{K} \in [\mathsf{PRFKey}(w, m, \mathsf{par})]$ and all $\mathbf{x} \in \{0,1\}^\ell$, we have

$$\Pr[\ |\widetilde{\mathsf{F}}(\mathbf{x}) - \mathsf{PRFEval}(\mathbf{S}, \mathbf{K}, \mathbf{x})| \leq \Delta : \widetilde{\mathsf{F}} \leftarrow \mathsf{Obf}(1^\lambda, 1^\ell, \mathbf{S}, \mathbf{K})\ ] = \varepsilon(\lambda).$$

- $\mathcal{D}$-Security: For all $w, m, \ell \in \mathbb{N}$, all $\chi > 0$, all par $\in \{0,1\}^*$, there exists a distribution $\mathcal{D}$ such that, for all $\mathcal{A}$,

$$\Pr[\mathcal{A}(1^\lambda, \boxed{\widetilde{\mathsf{F}}}) = 1 : \widetilde{\mathsf{F}} \leftarrow \mathsf{Obf}(1^\lambda, 1^\ell, \mathbf{S}, \mathbf{K})] - \Pr[\mathcal{A}(1^\lambda, \mathsf{F}_\$) = 1 : \mathsf{F}_\$ \leftarrow \mathcal{D}] = \varepsilon(\lambda)$$

  where $\mathbf{S} \leftarrow \mathsf{PRFGen}(w, \chi)$ and $\mathbf{K} \leftarrow \mathsf{PRFKey}(w, m, \mathsf{par})$.

Lattice Trapdoor & Algorithms. Let $m = 2n\log q$ and $\sigma \geq 2\sqrt{n\log q}$. We have the following P.P.T. algorithms [31]:

$$
\begin{array}{r|ll}
\mathsf{TrapGen}(1^n, q) \to (\mathbf{A}, \mathbf{A}^{-1}) & n \in \mathbb{N} & : \text{dimension} \\
\mathsf{PreSamp}(1^n, q, \mathbf{A}, \mathbf{A}^{-1}, \mathbf{z}, \sigma) \to \mathbf{k} & q \geq 2 & : \text{field} \\
 & \mathbf{A} \in \mathbb{Z}_q^{n\times m} & : \text{lattice instance} \\
 & \mathbf{A}^{-1} \in \mathbb{Z}^{m\times m} & : \text{lattice trapdoor} \\
 & \mathbf{z} \in \mathbb{Z}_q^{n} & : \text{target vector} \\
 & \sigma > 0 & : \text{noise parameter} \\
 & \mathbf{k} \in \mathbb{Z}^{m} & : \text{pre-image}
\end{array}
$$

with the following properties:

$$\mathbf{A} \sim \mathcal{U}(\mathbb{Z}_q^{n\times m}), \quad \mathbf{k} \sim \mathcal{D}_{\mathbb{Z},\sigma}^{m} \quad \text{and} \quad \mathbf{A}\mathbf{k} = \mathbf{z} \bmod q$$

This can be extended to matrix setting via column-wise extension. We use $(\mathbf{A}, \mathbf{A}^{-1}) \leftarrow \mathbb{Z}_q^{n\times m} \times \mathbb{Z}^{m\times m}$ to refer to TrapGen and use $\mathbf{k} \leftarrow \mathbf{A}^{-1}(\mathbf{z})$ to refer to PreSamp when n, q, σ is clear from the context.

Evasive LWE. Define P.P.T. algorithm:

$$
\begin{array}{r|ll}
 & \lambda & : \text{security parameter} \\
\mathsf{Samp}(1^\lambda) \to (\mathbf{S}, \mathbf{P}, \mathsf{aux}) & \mathbf{S} \in \mathbb{Z}_q^{n'\times n} & : \text{randomness} \\
 & \mathbf{P} \in \mathbb{Z}_q^{n\times t} & : \text{target} \\
 & \mathsf{aux} \in \{0,1\}^* & : \text{auxiliary information}
\end{array}
$$

The *evasive LWE assumption* $\mathsf{evLWE}_{\mathsf{Samp},\chi_0,\chi_1}$ [39,36] says that there exists polynomial Q such that for every P.P.T. $\mathcal{A}$, there exists P.P.T. $\mathcal{B}$ such that

$$
\begin{aligned}
&\overbrace{\Pr[\mathcal{A}(\boxed{\mathbf{SA}+\mathbf{E}}, \mathbf{A}_\chi^{-1}(\mathbf{P}), \mathsf{aux})] - \Pr[\mathcal{A}(\mathbf{C}, \mathbf{A}_\chi^{-1}(\mathbf{P}), \mathsf{aux})]}^{\mathsf{Adv}_{\mathcal{A}}^{\mathrm{PST}}(\lambda)} \\
\leq\ &\underbrace{\Pr[\mathcal{B}(\boxed{\mathbf{SA}+\mathbf{E}, \mathbf{SP}+\mathbf{E}'}, \mathsf{aux})] - \Pr[\mathcal{B}(\mathbf{C}, \mathbf{C}', \mathsf{aux})]}_{\mathsf{Adv}_{\mathcal{B}}^{\mathrm{PRE}}(\lambda)} \cdot Q(\lambda) + \varepsilon(\lambda)
\end{aligned}
$$

where $(\mathbf{S}, \mathbf{P}, \mathsf{aux}) \leftarrow \mathsf{Samp}(1^\lambda)$, $(\mathbf{A}, \mathbf{A}^{-1}) \leftarrow \mathbb{Z}_q^{n\times m} \times \mathbb{Z}^{m\times m}$, $\mathbf{E} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_0}^{n'\times m}$, $\mathbf{E}' \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{n'\times t}$, $\mathbf{C} \leftarrow \mathbb{Z}_q^{n'\times m}$ and $\mathbf{C}' \leftarrow \mathbb{Z}_q^{n'\times t}$. We write $\mathsf{evLWE}_{\chi_0,\chi_1}$ to indicate $\mathsf{evLWE}_{\mathsf{Samp},\chi_0,\chi_1}$ for some valid Samp.

Obfuscator from evasive LWE. We review the following theorem which ensures a concrete algebraic obfuscator for matrix PRF from evasive LWE. The reader can find their construction in Section A of **Appendix**.

Theorem 1 ([30]). *Let $n, q, w, \ell \in \mathbb{N}$ and*

$$\chi \geq \sqrt{2n}, \quad B \geq \sigma\sqrt{n}, \quad \sigma = 2^{\ell^3} \cdot (n^2\chi)^{\ell+1}, \quad W = O(w+n)\log q.$$

Under $\mathsf{LWE}_{n,\mathsf{poly}(n),q,\chi}$ and $\mathsf{evLWE}_{\sigma,\sigma}$ assumption, for σ-mPRF (PRFGen, PRFKey, PRFEval) of width $w \in \mathbb{N}$ and with entries of $\mathbf{S} \in [\mathsf{PRFGen}(w,\chi)]$ bounded by B, there exists an obfuscator Obf that achieves $\ell(WB)^\ell$-correctness for input of length ℓ and $\mathcal{D}$-security for some $\mathcal{D}$, and have

$$|\mathsf{Obf}(1^\lambda, 1^\ell, \mathbf{S}, \mathbf{K})| = O(\ell W^2 m \log q)$$

for all $\mathbf{S} \in [\mathsf{PRFGen}(w,\chi)]$ and $\mathbf{K} \in [\mathsf{PRFKey}(w, m, \mathsf{par})]$.

2.4 Slotted Registered Attribute-Based Encryption for Circuits

We review the notion of *slotted* registered attribute-based encryption (slotted Reg-ABE) adapted from [25]. See Section B of **Appendix** for more information about (full-fledged) Reg-ABE.

Algorithms. Let $s, d, \ell \in \mathbb{N}$. A *slotted registered attribute-based encryption* [25] for circuit is a tuple of algorithms with the following syntax:

$$\begin{array}{rl|rl} & & \lambda: & \text{security parameter} \\ & & d, \ell: & \text{depth/input size of circuits} \\ \mathsf{Setup}(1^\lambda, 1^d, 1^\ell) & \to \mathsf{crs} & \mathsf{crs}: & \text{common reference string} \\ \mathsf{Gen}(\mathsf{crs}) & \to (\mathsf{pk}, \mathsf{sk}) & \mathsf{pk}, \mathsf{pk}_i: & \text{user's public key} \\ \mathsf{Agg}(\mathsf{crs}, (f_i, \mathsf{pk}_i)_{i\in[L]}) & \to (\mathsf{mpk}, (\mathsf{hsk}_i)_{i\in[L]}) & \mathsf{sk}, \mathsf{sk}_i: & \text{user's secret key} \\ \mathsf{Enc}(\mathsf{mpk}, x, \mathsf{m}) & \to \mathsf{ct} & \mathsf{mpk}: & \text{master public key} \\ \mathsf{Dec}(\mathsf{sk}, \mathsf{hsk}, \mathsf{ct}) & \to \mathsf{m}/\bot & \mathsf{hsk}: & \text{helper secret key} \\ & & f_i \in \mathcal{C}_{d,\ell}: & \text{function for } i\text{-th user} \\ & & x \in \{0,1\}^\ell: & \text{input to function} \end{array}$$

We require that Agg is deterministic.

Correctness and Compactness. For all $\lambda, L, d, \ell \in \mathbb{N}$, all $i^* \in [L]$, all $f_1, \ldots, f_L \in \mathcal{C}_{d,\ell}$, all $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^d, 1^\ell)$, all $(\mathsf{pk}_{i^*}, \mathsf{sk}_{i^*}) \leftarrow \mathsf{Gen}(\mathsf{crs})$, all $\{\mathsf{pk}_i\}_{i\in[L]\setminus\{i^*\}}$, all $x \in \{0,1\}^\ell$ such that $f_{i^*}(x) = 0$, and all $\mathsf{m} \in \mathcal{M}$, *correctness* requires that

$$\Pr[\mathsf{Dec}(\mathsf{sk}_{i^*}, \mathsf{hsk}_{i^*}, \mathsf{Enc}(\mathsf{mpk}, x, \mathsf{m})) = \mathsf{m}] = 1$$

where $(\mathsf{mpk}, (\mathsf{hsk}_i)_{i\in[L]}) \leftarrow \mathsf{Agg}(\mathsf{crs}, (f_i, \mathsf{pk}_i)_{i\in[L]})$ and *compactness* requires that

$$|\mathsf{mpk}| = \mathsf{poly}(\lambda, d, \ell, \log L) \quad \text{and} \quad |\mathsf{hsk}_i| = \mathsf{poly}(\lambda, d, \ell, \log L) \quad \forall i \in [L].$$

Security. The *(adaptive) security* requires that, for all P.P.T. adversary $\mathcal{A}$,

$$\Pr\left[\beta = \beta' \,\middle|\, \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^d, 1^\ell) \\ (x, (\mathsf{pk}_i, f_i)_{i\in[L]}, \mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathcal{A}^{\mathsf{OGen},\mathsf{OCor}}(\mathsf{crs}) \\ (\mathsf{mpk}, (\mathsf{hsk}_i)_{i\in[L]}) \leftarrow \mathsf{Agg}(\mathsf{crs}, (f_i, \mathsf{pk}_i)_{i\in[L]}) \\ \beta \leftarrow \{0,1\},\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x, \mathsf{m}_\beta),\ \beta' \leftarrow \mathcal{A}(\mathsf{ct}) \end{array}\right] - \frac{1}{2} = \varepsilon(\lambda)$$

where the oracles work as follows with initial setting $\mathcal{C} = \emptyset$ and $\mathcal{L} = \emptyset$:

- $\mathsf{OGen}()$: run $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{crs})$, set $\mathcal{L}[\mathsf{pk}] = \mathsf{sk}$ and return pk;
- $\mathsf{OCor}(\mathsf{pk})$: return $\mathcal{L}[\mathsf{pk}]$ and update $\mathcal{C} = \mathcal{C} \cup \{\mathsf{pk}\}$;

and condition:

$$\overbrace{\mathsf{pk}_i \in \mathcal{C}}^{\text{corrupted key}} \quad \vee \quad \overbrace{\mathcal{L}[\mathsf{pk}_i] = \bot}^{\text{malicious key}} \quad \Longrightarrow \quad f_i(x) = 1 \qquad \forall i \in [L].$$

Note that [25] proved that there is no need to give mpk and $\mathsf{hsk}_1, \ldots, \mathsf{hsk}_L$ to $\mathcal{A}$ explicitly and to consider post-challenge queries. There are two orthogonal ways to adapt the definition:

- when $\mathcal{A}$ claims x before seeing crs, we get the notion of *selective* security;

– when $\mathcal{A}$ receives either $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x, \mathsf{m})$ with m chosen by $\mathcal{A}$ or a random string in the ciphertext space, we get the notion of *pseudorandom ciphertext*.

We finally remark that, among $\mathsf{pk}_1, \ldots, \mathsf{pk}_L$ that appear in the above definition, we distinguish three types of users/public keys:

– pk_i with $\mathcal{L}[\mathsf{pk}_i] = \perp$ is *malicious*; challenger does not know sk_i;
– pk_i with $\mathsf{pk}_i \in \mathcal{C}$ is *corrupted*; both challenger and adversary know sk_i;
– remaining pk_i are *honest*; adversary does not know sk_i but challenger does.

3 Slotted Registered ABE with Large Ciphertext

This section presents our slotted Reg-ABE scheme with *large* ciphertexts from LWE assumption. We will adapt it to a slotted Reg-ABE scheme with *compact* ciphertexts in Section 4.3. For this ultimate purpose, here, we divide Dec into two algorithms IndDec and Dec: IndDec is the core part of old Dec and will only be invoked by Dec.

3.1 Scheme

Assuming $\mathsf{PRF}_0 = (\mathsf{PRFGen}_0, \mathsf{PRFKey}_0, \mathsf{PRFEval}_0)$ promised by Lemma 1, our slotted Reg-ABE scheme with large ciphertext works as follow.

– $\mathsf{Setup}(1^\lambda, 1^d, 1^\ell)$: Sample

$$\mathbf{B}_0, \mathbf{B}_1, \mathbf{P}, \mathbf{D} \leftarrow \mathbb{Z}_q^{n \times m}, \quad \mathbf{F} \leftarrow \mathbb{Z}_q^{n \times m\ell}, \quad \mathbf{v} \leftarrow \mathbb{Z}_q^n.$$

Output

$$\mathsf{crs} = (\mathbf{B}_0, \mathbf{B}_1, \mathbf{F}, \mathbf{P}, \mathbf{D}, \mathbf{v}, d, \ell).$$

– $\mathsf{Gen}(\mathsf{crs})$: Sample $\mathbf{k} \leftarrow \{0,1\}^m$ and set $\mathbf{u} = \mathbf{D}\mathbf{k} \in \mathbb{Z}_q^n$. Output

$$\mathsf{pk} = \mathbf{u} \quad \text{and} \quad \mathsf{sk} = \mathbf{k}.$$

– $\mathsf{Agg}(\mathsf{crs}, \{f_i, \mathsf{pk}_i\}_{i \in [L]})$: Assume $L = 2^D$ for some $D \in \mathbb{N}$ and write

$$i = (i_1, \ldots, i_D) \in \{0,1\}^D.$$

Let

$$\mathsf{pk}_i = \mathbf{u}_i \in \mathbb{Z}_q^n, \quad \mathbf{H}_{f_i} \leftarrow \mathsf{EvalF}(\mathbf{F}, f_i) \quad \forall i \in \{0,1\}^D.$$

Compute

$$\mathbf{h}_i = \mathbf{F}\mathbf{H}_{f_i}\mathbf{G}^{-1}(\mathbf{v}) + \mathbf{P}\mathbf{G}^{-1}(\mathbf{u}_i)$$

For $j = D-1, \ldots, 0$, recursively compute

$$\mathbf{h}_\iota = \mathbf{B}_0\mathbf{G}^{-1}(\mathbf{h}_{\iota\|0}) + \mathbf{B}_1\mathbf{G}^{-1}(\mathbf{h}_{\iota\|1}), \quad \forall \iota \in \{0,1\}^j.$$

Output

$$\mathsf{mpk} = (\mathsf{crs}, \mathbf{h}_\epsilon, L), \quad \mathsf{hsk}_i = \{\overbrace{\mathbf{h}_{i_{|j-1}\|0}}^{\mathbf{h}_{i,j,0}}, \overbrace{\mathbf{h}_{i_{|j-1}\|1}}^{\mathbf{h}_{i,j,1}}\}_{j \in [D]} \quad \forall i \in \{0,1\}^D.$$

We assume that one can efficiently extract both i and f_i from hsk_i.

– $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}, \mathsf{m})$: Let $\mathsf{mpk} = (\overbrace{\mathbf{B}_0, \mathbf{B}_1, \mathbf{F}, \mathbf{P}, \mathbf{D}, \mathbf{v}, d, \ell}^{\mathsf{crs}}, \mathbf{h}_\epsilon, L = 2^D)$. Run

$$\mathbf{S} \leftarrow \mathsf{PRFGen}_0(m_0, \sigma_0) \quad \text{and} \quad \mathbf{K}_0, \mathbf{K}_1, \ldots, \mathbf{K}_{D+1} \leftarrow \mathsf{PRFKey}_0(m_0, n).$$

For each $i \in \{0,1\}^D$ and $j \in [0, D+1]$, define

$$\mathbf{s}_{i,j} = \mathsf{PRFEval}_0(\mathbf{S}, \mathbf{K}_j, i) \in \mathbb{Z}_q^n$$

and sample

$$\mathbf{e}_i = (e_{i,0}, \mathbf{e}_{i,1}, \ldots, \mathbf{e}_{i,D}, \mathbf{e}_{i,D+1}, \mathbf{e}_{i,D+2}) \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_1}^{1+(2D+\ell+2)m}$$

Compute

$$\begin{aligned}
c_{i,0} &= \mathbf{s}_{i,0}^\top \mathbf{h}_\epsilon + e_{i,0} + \lfloor q/2 \rfloor \cdot \mathsf{m} \in \mathbb{Z}_q \\
\mathbf{c}_{i,j}^\top &= -\mathbf{s}_{i,j-1}^\top (\mathbf{B}_0 \mid \mathbf{B}_1) + \mathbf{s}_{i,j}^\top (\bar{i}_j \cdot \mathbf{G} \mid i_j \cdot \mathbf{G}) + \mathbf{e}_{i,j}^\top \in \mathbb{Z}_q^{2m} \quad \forall j \in [D] \\
\mathbf{c}_{i,D+1}^\top &= -\mathbf{s}_{i,D}^\top (\mathbf{F} - \mathbf{x}^\top \otimes \mathbf{G} \mid \mathbf{P}) + \mathbf{s}_{i,D+1}^\top (\mathbf{0} \mid \mathbf{G}) + \mathbf{e}_{i,D+1}^\top \in \mathbb{Z}_q^{m(\ell+1)} \\
\mathbf{c}_{i,D+2}^\top &= -\mathbf{s}_{i,D+1}^\top \mathbf{D} + \mathbf{e}_{i,D+2}^\top \in \mathbb{Z}_q^m
\end{aligned}$$

and output

$$\mathsf{ct}_\mathbf{x} = \{ \overbrace{c_{i,0}, \mathbf{c}_{i,1}, \ldots, \mathbf{c}_{i,D}, \mathbf{c}_{i,D+1}, \mathbf{c}_{i,D+2}}^{\mathsf{ct}_{\mathbf{x},i}} \}_{i \in \{0,1\}^D}.$$

We assume that one can efficiently extract $\mathbf{x}$ from $\mathsf{ct}_\mathbf{x}$ and note that

$$|\mathsf{ct}_\mathbf{x}| = L \cdot \mathsf{polylog}(L).$$

– $\mathsf{Dec}(\mathsf{sk}_{i^*}, \mathsf{hsk}_{i^*}, \mathsf{ct}_\mathbf{x})$: Parse $i^* \in \{0,1\}^D$ from sk_{i^*} and hsk_{i^*}. Let

$$\mathsf{ct}_\mathbf{x} = \{\overbrace{c_{i,0}, \mathbf{c}_{i,1}, \ldots, \mathbf{c}_{i,D}, \mathbf{c}_{i,D+1}, \mathbf{c}_{i,D+2}}^{\mathsf{ct}_{\mathbf{x},i^*}}\}_{i \in \{0,1\}^D}$$

Output

$$\mathsf{m} \leftarrow \mathsf{IndDec}(\mathsf{sk}_{i^*}, \mathsf{hsk}_{i^*}, \mathsf{ct}_{\mathbf{x},i^*}).$$

– $\mathsf{IndDec}(\mathsf{sk}, \mathsf{hsk}, \mathsf{ct}_\mathbf{x})$: Parse

$$\mathsf{sk} = \mathbf{k} \quad \text{and} \quad \mathsf{hsk} = \{\mathbf{h}_{j,0}, \mathbf{h}_{j,1}\}_{j \in [D]} \quad \text{with} \quad f \in \mathcal{C}_{d,\ell}$$

and

$$\mathsf{ct} = (c_0, \mathbf{c}_1, \ldots, \mathbf{c}_D, \mathbf{c}_{D+1}, \mathbf{c}_{D+2}) \quad \text{with} \quad \mathbf{x} \in \{0,1\}^\ell.$$

Let $\mathbf{u} = \mathbf{D}\mathbf{k}$ and run $\mathbf{H}_{f,\mathbf{x}} \leftarrow \mathsf{EvalFX}(\mathbf{F}, f, \mathbf{x})$. Compute

$$z = c_0 + \sum_{j=1}^{D} \overbrace{\mathbf{c}_j^\top \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{j,0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{j,1}) \end{bmatrix}}^{z_j} + \overbrace{\mathbf{c}_{D+1}^\top \begin{bmatrix} \mathbf{H}_{f,\mathbf{x}} \mathbf{G}^{-1}(\mathbf{v}) \\ \mathbf{G}^{-1}(\mathbf{u}) \end{bmatrix}}^{z_{D+1}} + \overbrace{\mathbf{c}_{D+2}^\top \mathbf{k}}^{z_{D+2}}.$$

and output $\lfloor 2z/q \rceil$.

3.2 Parameters

We set $n, m, m_0, q, \sigma_0, \sigma_1$ so that they satisfy the following conditions:

$$
\begin{aligned}
& q/4 \geq D\ell m^{O(d)}\sqrt{\lambda}\sigma_1 && \text{// correctness} \\
& m > 2n\log q && \text{// homomorphic evaluation, LHL} \\
& \sigma_1 \geq \lambda^{\omega(1)} \cdot \chi_0 && \text{// } \mathsf{G}_0 \\
& \chi_0 \geq D\ell \cdot m^{O(d)}\lambda^{\omega(1)}\chi_1 && \text{// } \mathsf{G}_0 \approx_s \mathsf{G}_1 \\
& \sigma_0 = \Omega(\sqrt{n\log q}),\ m_0 = 6n\log q && \text{// Lemma 1} \\
& \chi_1 \geq O(\lambda^{D+\omega(1)}(n\log q)^{2D} m\chi_2) && \text{// Lemma 2} \\
& q/\chi_2 \leq 2^{n^c},\ \chi_2 = \mathsf{poly}(n,\lambda) && \text{// LWE hardness}
\end{aligned}
$$

where χ_0, χ_1 and χ_2 are introduced in the proof as intermediate parameters. We defer parameter selection to the next section where we present our final slotted Reg-ABE scheme and two additional conditions will be added.

3.3 Correctness

Assume $\mathsf{ct}_{\mathbf{x}}$ is generated under $\mathsf{mpk} = (\mathbf{B}_0, \mathbf{B}_1, \mathbf{F}, \mathbf{P}, \mathbf{D}, \mathbf{v}, d, \ell, \mathbf{h}_\epsilon, L = 2^D)$. Recall that Dec sends those terms corresponding to i^* in $\mathsf{ct}_{\mathbf{x}}$ to IndDec:

$$
\mathsf{ct}_{\mathbf{x},i^*} = \begin{cases}
c_{i^*,0} & = \mathbf{s}_{i^*,0}^\top \mathbf{h}_\epsilon + e_{i^*,0} + \lfloor q/2 \rfloor \cdot \mathsf{m} \\
\mathbf{c}_{i^*,j}^\top & = -\mathbf{s}_{i^*,j-1}^\top (\mathbf{B}_0 \mid \mathbf{B}_1) + \mathbf{s}_{i^*,j}^\top (\bar{i}_j^* \cdot \mathbf{G} \mid i_j^* \cdot \mathbf{G}) + \mathbf{e}_{i^*,j}^\top,\ \forall j \in [D] \\
\mathbf{c}_{i^*,D+1}^\top & = -\mathbf{s}_{i^*,D}^\top (\mathbf{F} - \mathbf{x}^\top \otimes \mathbf{G} \mid \mathbf{P}) + \mathbf{s}_{i^*,D+1}^\top (\mathbf{0} \mid \mathbf{G}) + \mathbf{e}_{i^*,D+1}^\top \\
\mathbf{c}_{i^*,D+2}^\top & = -\mathbf{s}_{i^*,D+1}^\top \mathbf{D} + \mathbf{e}_{i^*,D+2}^\top
\end{cases} \tag{5}
$$

along with

$$
\mathsf{sk}_{i^*} = \mathbf{k}_{i^*}, \qquad \mathsf{hsk}_{i^*} = \{\mathbf{h}_{i^*,j,0}, \mathbf{h}_{i^*,j,1}\}_{j\in[D]}
$$

By the specification of Gen and Agg, we have the following relations:

$$
\mathbf{h}_\epsilon = \mathbf{B}_0\mathbf{G}^{-1}(\mathbf{h}_{i^*,1,0}) + \mathbf{B}_1\mathbf{G}^{-1}(\mathbf{h}_{i^*,1,1}) \tag{6}
$$

$$
\mathbf{h}_{i^*,j-1,i_{j-1}^*} = \mathbf{B}_0\mathbf{G}^{-1}(\mathbf{h}_{i^*,j,0}) + \mathbf{B}_1\mathbf{G}^{-1}(\mathbf{h}_{i^*,j,1}), \quad \forall j \in [2, D] \tag{7}
$$

$$
\mathbf{h}_{i^*,D,i_D^*} = \mathbf{F}\mathbf{H}_{f_{i^*}}\mathbf{G}^{-1}(\mathbf{v}) + \mathbf{P}\mathbf{G}^{-1}(\mathbf{u}_{i^*}) \ \text{ where } \ \mathbf{u}_{i^*} = \mathbf{D}\mathbf{k}_{i^*}. \tag{8}
$$

For brevity, we discard transcript i^* from these terms and f_{i^*}. Observe that:

$$
z_1 = (-\mathbf{s}_0^\top \mathbf{h}_\epsilon + \mathbf{s}_1^\top \mathbf{h}_{1,i_1^*}) + \mathbf{e}_1^\top \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{1,0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{1,1}) \end{bmatrix} \tag{9}
$$

$$
z_j = (-\mathbf{s}_{j-1}^\top \mathbf{h}_{j-1,i_{j-1}^*} + \mathbf{s}_j^\top \mathbf{h}_{j,i_j^*}) + \mathbf{e}_j^\top \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{j,0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{j,1}) \end{bmatrix} \qquad \forall j \in [2, D] \tag{10}
$$

$$
z_{D+1} = -\mathbf{s}_D^\top \mathbf{h}_{D,i_D^*} + f(\mathbf{x}) \cdot \mathbf{s}_D^\top \mathbf{v} + \mathbf{s}_{D+1}^\top \mathbf{u} + \mathbf{e}_{D+1}^\top \begin{bmatrix} \mathbf{H}_{f,\mathbf{x}}\mathbf{G}^{-1}(\mathbf{v}) \\ \mathbf{G}^{-1}(\mathbf{u}) \end{bmatrix} \tag{11}
$$

$$
z_{D+2} = -\mathbf{s}_{D+1}^\top \mathbf{u} + \mathbf{e}_{D+2}^\top \mathbf{k} \tag{12}
$$

We leave all details of computing $z_1, \ldots, z_{D+2}$ to Section C in **Appendix**. It is straight-forward to see that, when $f(\mathbf{x}) = 0$, we have

$$
z = c_{i^*,0} + \sum_{j=1}^{D} z_j + z_{D+1} + z_{D+2}
$$

$$= \lfloor q/2 \rfloor \cdot \mathsf{m} + \overbrace{e_0 + \sum_{j=1}^{D} \mathbf{e}_j^\top \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{j,0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{j,1}) \end{bmatrix} + \mathbf{e}_{D+1}^\top \begin{bmatrix} \mathbf{H}_{f,\mathbf{x}} \mathbf{G}^{-1}(\mathbf{v}) \\ \mathbf{G}^{-1}(\mathbf{u}) \end{bmatrix} + \mathbf{e}_{D+2}^\top \mathbf{k}}^{e}$$

Then the correctness follows from the fact that

$$\begin{aligned} |e| &\leq |e_0| + \sum_{j=1}^{D} \|\mathbf{e}_j^\top\| + \|\mathbf{e}_{D+1}^\top\| \cdot \|\mathbf{H}_{f,\mathbf{x}}\| + \|\mathbf{e}_{D+2}^\top\| \\ &\leq (1 + 2mD + m(\ell+1)m^{O(d)} + m)\sqrt{\lambda}\sigma_1 = D\ell m^{O(d)}\sqrt{\lambda}\sigma_1 \leq q/4. \end{aligned} \tag{13}$$

3.4 Security

We have the following theorem.

Theorem 2. *Under* $\mathsf{LWE}_{n,\mathsf{poly}(m_0,D,2^D),q,\sigma_0}$ *and* $\mathsf{LWE}_{n,O(m),q,\chi_2}$ *assumption that satisfy the conditions in Section 3.2, our slotted Reg-ABE presented in Section 3.1 has pseudorandom ciphertexts in the selective setting (c.f. Section 2.4).*

Useful Lemma. We prepare the following simple lemma which will be frequently used in the proof. Consider a σ-mPRF $\mathsf{PRF} = (\mathsf{PRFGen}, \mathsf{PRFKey}, \mathsf{PRFEval})$. Given $\mathbf{S} \leftarrow \mathsf{PRFGen}(w, \chi)$ and $m, \ell \in \mathbb{N}$, we write

$$\mathsf{F}(\mathbf{K}, \mathbf{x}) = \mathsf{PRFEval}(\mathbf{S}, \mathbf{K}, \mathbf{x}) \qquad \forall\, \mathbf{K} \in \mathbb{Z}_q^{w \times m}, \mathbf{x} \in \{0,1\}^\ell.$$

Then, we have

$$\mathsf{F}(\mathbf{K}, \mathbf{x}) + \mathbf{e}_\mathbf{x} \approx_c \$$$

where $\mathbf{K} \leftarrow \mathsf{PRFKey}(w, m, \mathsf{par})$ and $\mathbf{e}_\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$; here, we use \$ to refer to a random vector of proper size. However, our proof will instead need the following argument:

$$\mathsf{F}(\mathbf{K}', \mathbf{x}) \cdot \mathbf{P} + \mathbf{e}'_\mathbf{x} \approx_c \$ \tag{14}$$

where $\mathbf{K}' \leftarrow \mathsf{PRFKey}(w, m', \mathsf{par})$ with $m' < m$ and $\mathbf{P}$ is a *public random* matrix. A straight-forward idea is to make use of σ-pseudorandomness. For this, we employ noise flooding that changes L.H.S. as:

$$\boxed{(\mathsf{F}(\mathbf{K}', \mathbf{x}) + \tilde{\mathbf{e}}_\mathbf{x})} \cdot \mathbf{P} + \mathbf{e}'_\mathbf{x}$$

and argue that the boxed term is pseudorandom. Unfortunately, this actually does *not* work due to fact that $\mathbf{P}$ is *not* a low-norm matrix. Our lemma (shown below) shows that, if PRFKey simply samples a truly random matrix $\mathbf{K}$ (of size depending on the input) and PRF has low-norm (as defined in Section 2.2), we can have (14) where $\mathsf{F}(\mathbf{K}', \mathbf{x})$ is able to interplay with the noise term $\mathbf{e}'_\mathbf{x}$ across the "large" matrix $\mathbf{P}$.

Lemma 2 (Crossing Lemma). *Let* $w, \ell, m, n, B \in \mathbb{N}$ *and* $\sigma, \chi > 0$. *Assume a* σ*-mPRF* $\mathsf{PRF} = (\mathsf{PRFGen}, \mathsf{PRFKey}, \mathsf{PRFEval})$ *of norm* B *with*

- $\mathsf{PRFKey}(w, m)$ *outputs* $\tilde{\mathbf{K}} \leftarrow \mathbb{Z}_q^{w \times m}$.

Under $\mathsf{LWE}_{n,m,q,\chi}$ *with* $\lambda^{\omega(1)} B m \chi \leq \sigma$, *we have*

$$\mathbf{P}, \{\mathsf{PRFEval}(\mathbf{S}, \mathbf{K}, \mathbf{x}) \cdot \mathbf{P} + \mathbf{e}_\mathbf{x}^\top\}_{\mathbf{x} \in \{0,1\}^\ell} \approx_c \mathbf{P}, \{\mathbf{u}_\mathbf{x}^\top\}_{\mathbf{x} \in \{0,1\}^\ell}$$

where $\mathbf{S} \leftarrow \mathsf{PRFGen}(w, \chi')$ *(for some* $\chi' > 0$*),* $\mathbf{K} \leftarrow \mathsf{PRFKey}(w, n)$ *with* $n < m$, $\mathbf{P} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{e}_\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$ *and* $\mathbf{u}_\mathbf{x} \leftarrow \mathbb{Z}_q^m$ *for all* $\mathbf{x} \in \{0,1\}^\ell$.

Proof. By definition of σ-mPRF, we have

$$\{\mathbf{t}^\top \mathbf{M}_{1,x_1} \cdots \mathbf{M}_{\ell,x_\ell} \tilde{\mathbf{K}} + \mathbf{e}_{\mathbf{x}}^\top\}_{\mathbf{x}\in\{0,1\}^\ell} \approx_c \{\mathbf{u}_{\mathbf{x}}^\top\}_{\mathbf{x}\in\{0,1\}^\ell} \tag{15}$$

when $\tilde{\mathbf{K}} \leftarrow \mathsf{PRFKey}(w, m)$, $\mathbf{e}_{\mathbf{x}} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{u}_{\mathbf{x}} \leftarrow \mathbb{Z}_q^m$ and

$$\|\mathbf{t}^\top \mathbf{M}_{1,x_1} \cdots \mathbf{M}_{\ell,x_\ell}\| \le B \quad \forall \mathbf{x} \in \{0,1\}^\ell. \tag{16}$$

Recall that $\mathbf{M}_{i,b}$ for all $i \in [\ell]$ and $b \in \{0,1\}$ are derived from $\mathbf{S}$. The lemma follows from the following hybrid arguments:

$$\begin{aligned} \text{L.H.S.} &= \mathbf{P}, \{\mathbf{t}^\top \mathbf{M}_{1,x_1} \cdots \mathbf{M}_{\ell,x_\ell} \mathbf{K}\mathbf{P} + \mathbf{e}_{\mathbf{x}}^\top\}_{\mathbf{x}\in\{0,1\}^\ell} \\ &\approx_s \mathbf{P}, \{\mathbf{t}^\top \mathbf{M}_{1,x_1} \cdots \mathbf{M}_{\ell,x_\ell} (\mathbf{K}\mathbf{P} + \mathbf{E}) + \mathbf{e}_{\mathbf{x}}^\top\}_{\mathbf{x}\in\{0,1\}^\ell} \\ &\approx_c \mathbf{P}, \{\mathbf{t}^\top \mathbf{M}_{1,x_1} \cdots \mathbf{M}_{\ell,x_\ell} \widetilde{\mathbf{K}} + \mathbf{e}_{\mathbf{x}}^\top\}_{\mathbf{x}\in\{0,1\}^\ell} \\ &\approx_c \mathbf{P}, \{\mathbf{u}_{\mathbf{x}}^\top\}_{\mathbf{x}\in\{0,1\}^\ell} = \text{R.H.S.} \end{aligned}$$

where $\mathbf{E} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{w\times m}$. Here,

- the first $\approx_s$ uses noise flooding and (16) with condition $\lambda^{\omega(1)} B m \chi \le \sigma$;
- the second $\approx_c$ uses $\mathsf{LWE}_{n,m,q,\chi}$ assumption: $(\mathbf{P}, \mathbf{K}\mathbf{P} + \mathbf{E}) \approx_c (\mathbf{P}, \widetilde{\mathbf{K}})$;
- the third $\approx_c$ follows from σ-pseudorandomness of PRF, i.e., (15).

This readily proves the lemma. □

Game Sequence. Let $L = 2^D$ be the number slots chosen by the adversary. Let $\mathbf{x}$ be the selective challenge that are given by $\mathcal{A}$ before seeing crs. Let $(\mathsf{pk}_i, f_i)_{i\in\{0,1\}^D}$ be the key-policy pairs to be aggregated. Let

$$\mathcal{L}_{\text{hon}} = \{\mathsf{pk}_i : \mathcal{L}[\mathsf{pk}_i] \ne \bot \wedge \mathsf{pk}_i \notin \mathcal{C}\}$$

be the set of public keys for honest users. For all $i \in \{0,1\}^D$, we have that

$$f_i(\mathbf{x}) = 0 \implies \mathsf{pk}_i \in \mathcal{L}_{\text{hon}}.$$

Our proof uses the following game sequence.

- G_0: This is the real game. We have $\mathsf{crs} = (\mathbf{B}_0, \mathbf{B}_1, \mathbf{F}, \mathbf{P}, \mathbf{D}, \mathbf{v}, d, \ell)$. For each $i \in \{0,1\}^D$, we have $\mathsf{pk}_i = \mathbf{u}_i \in \mathbb{Z}_q^n$. With

  $$\mathbf{S} \leftarrow \mathsf{PRFGen}_0(m_0, \sigma_0) \quad \text{and} \quad \mathbf{K}_0, \mathbf{K}_1, \ldots, \mathbf{K}_{D+1} \leftarrow \mathsf{PRFKey}_0(m_0, n),$$

  we write

  $$\mathsf{F}(\mathbf{K}_j, i) = \mathsf{PRFEval}_0(\mathbf{S}, \mathbf{K}_j, i) \quad \forall i \in \{0,1\}^D, j \in [D+1].$$

  The challenge ciphertext is in the following form

  $$\mathsf{ct}_{\mathbf{x}} = \{c_{i,0}, \mathbf{c}_{i,1}, \ldots, \mathbf{c}_{i,D}, \mathbf{c}_{i,D+1}, \mathbf{c}_{i,D+2}\}_{i\in\{0,1\}^D}$$

  where

  $$\begin{aligned} c_{i,0} &= \mathsf{F}(\mathbf{K}_0, i) \cdot \mathbf{h}_\epsilon + e_{i,0} \\ \mathbf{c}_{i,j}^\top &= -\mathsf{F}(\mathbf{K}_{j-1}, i) \cdot (\mathbf{B}_0 \mid \mathbf{B}_1) + \mathsf{F}(\mathbf{K}_j, i) \cdot (\bar{i}_j \cdot \mathbf{G} \mid i_j \cdot \mathbf{G}) + \mathbf{e}_{i,j}^\top, \ \forall j \in [D] \\ \mathbf{c}_{i,D+1}^\top &= -\mathsf{F}(\mathbf{K}_D, i) \cdot (\mathbf{F} - \mathbf{x}^\top \otimes \mathbf{G} \mid \mathbf{P}) + \mathsf{F}(\mathbf{K}_{D+1}, i) \cdot (\mathbf{0} \mid \mathbf{G}) + \mathbf{e}_{i,D+1}^\top \\ \mathbf{c}_{i,D+2}^\top &= -\mathsf{F}(\mathbf{K}_{D+1}, i) \cdot \mathbf{D} + \mathbf{e}_{i,D+2}^\top \end{aligned}$$

  For simplicity, we omit $\lfloor q/2 \rfloor \cdot \mathsf{m}$ in $c_{i,0}$ and consider the following noises:

  $$e_{i,0} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_0},\ \mathbf{e}_{i,1}, \ldots, \mathbf{e}_{i,D} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{2m},\ \mathbf{e}_{i,D+1} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{m(\ell+1)},\ \mathbf{e}_{i,D+2} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{m}.$$

  Smudging lemma with condition $\sigma_1 \ge \lambda^{\omega(1)} \cdot \chi_0$ gives the same distribution as the real scheme. Looking ahead, our proof will require that $\chi_0 \ge \chi_1$, this implies $\sigma_1 \ge \lambda^{\omega(1)} \cdot \chi_1$.

– G_1: Identical to G_0 except that we rewrite $c_{i,0}$ for all $i \in \{0,1\}^D$ as follows:

$$c_{i,0} = -\sum_{j=1}^{D} \mathbf{c}_{i,j}^\top \mathbf{w}_{i,j} + \mathbf{c}_{i,D+1}^\top \mathbf{w}_{i,D+1} + f_i(\mathbf{x}) \cdot \mathsf{F}(\mathbf{K}_D, i) \cdot \mathbf{v} + \mathsf{F}(\mathbf{K}_{D+1}, i) \cdot \mathbf{u}_i + e_{i,0}$$

where

$$\mathbf{w}_{i,j} = \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{i_{|j-1}\|0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{i_{|j-1}\|1}) \end{bmatrix} \quad \forall j \in [D] \quad \text{and} \quad \mathbf{w}_{i,D+1} = \begin{bmatrix} \mathbf{H}_{f_i,\mathbf{x}} \mathbf{G}^{-1}(\mathbf{v}) \\ \mathbf{G}^{-1}(\mathbf{u}_i) \end{bmatrix}$$

We claim that $\mathsf{G}_0 \approx_s \mathsf{G}_1$. This follows (1) Gaussian tail bound which ensures that, with probability $1 - 2^{-\lambda}$, we have

$$\Bigg\| \overbrace{\sum_{j=1}^{D} \mathbf{e}_{i,j}^\top \mathbf{w}_{i,j} + \mathbf{e}_{i,D+1}^\top \mathbf{w}_{i,D+1}}^{e^*} \Bigg\| \le \sum_{j=1}^{D} \overbrace{\|\mathbf{e}_{i,j}^\top\|}^{2m\cdot\sqrt{\lambda}\chi_1} + \overbrace{\|\mathbf{e}_{i,D+1}^\top\|}^{m(\ell+1)\cdot\sqrt{\lambda}\chi_1} \cdot \overbrace{\|\mathbf{H}_{f_i,\mathbf{x}}\|}^{m^{O(d)}}$$
$$\le D\ell m^{O(d)} \sqrt{\lambda}\chi_1$$

and (2) smudging lemma with condition $\chi_0 \ge D\ell m^{O(d)} \lambda^{\omega(1)} \chi_1$.

– $\mathsf{G}_{2.\delta}, \delta \in [0, D]$: Identical to G_1 except that we replace $\mathbf{c}_{i,\delta}$ with $\tilde{\mathbf{c}}_{i,\delta} \leftarrow \mathbb{Z}_q^{2m}$ for all $i \in \{0,1\}^D$. Clearly, we have $\mathsf{G}_1 = \mathsf{G}_{2.0}$. We claim that $\mathsf{G}_{2.\delta-1} \approx_c \mathsf{G}_{2.\delta}$ for all $\delta \in [D]$. This follows from Lemma 1 and Lemma 2. The former ensures that $(\mathsf{PRFGen}_0, \mathsf{PRFKey}_0, \mathsf{PRFEval}_0)$ is χ_1-mPRF of norm at most $(m_0\lambda\sigma_0)^D$ under $\mathsf{LWE}_{n,\mathsf{poly}(m_0,D,2^D),q,\sigma_0}$ assumption with conditions

$$m_0 = 6n \log q, \quad \sigma_0 = \Omega(\sqrt{n \log q}), \quad \chi_1 \ge \lambda^{D+\omega(1)} \cdot (m_0\sigma_0)^D.$$

Along with the fact that PRFKey_0 samples a random matrix, the latter implies that: Under $\mathsf{LWE}_{n,O(m),q,\chi_2}$ assumption, it holds that

$$\{-\mathsf{F}(\mathbf{K}_{\delta-1}, i) \cdot (\mathbf{B}_0 \mid \mathbf{B}_1) + \mathbf{e}_{i,\delta}^\top\}_{i \in \{0,1\}^D} \text{ are pseudorandom}$$

with condition $O(\lambda^{D+\omega(1)} (m_0\sigma_0)^D m\chi_2) \le \chi_1$.

– G_3: Identical to $\mathsf{G}_{2.D}$ except that we replace $c_{i,0}$, $\mathbf{c}_{i,D+1}$ and $\mathbf{c}_{i,D+2}$ with $\tilde{c}_{i,0} \leftarrow \mathbb{Z}_q$, $\tilde{\mathbf{c}}_{i,D+1} \leftarrow \mathbb{Z}_q^{m(\ell+1)}$ and $\tilde{\mathbf{c}}_{i,D+2} \leftarrow \mathbb{Z}_q^m$, respectively, for all $i \in \{0,1\}^D$. We prove $\mathsf{G}_{2.D} \approx_c \mathsf{G}_3$ later on.

This readily proves that the challenge ciphertext is pseudorandom. In the remaining of this section, we prove the last transition in the game sequence.

From $\mathsf{G}_{2.D}$ to G_3. Recall that challenge ciphertext $\mathsf{ct}_\mathbf{x}$ in $\mathsf{G}_{2.D}$ looks like:

$$\begin{aligned} c_{i,0} &= -\textstyle\sum_{j=1}^{D} \tilde{\mathbf{c}}_{i,j}^\top \mathbf{w}_{i,j} + \mathbf{c}_{i,D+1}^\top \mathbf{w}_{i,D+1} + f_i(\mathbf{x}) \cdot \mathsf{F}(\mathbf{K}_D, i) \cdot \mathbf{v} + \mathsf{F}(\mathbf{K}_{D+1}, i) \cdot \mathbf{u}_i + e_{i,0} \\ \mathbf{c}_{i,D+1}^\top &= -\mathsf{F}(\mathbf{K}_D, i) \cdot (\mathbf{F} - \mathbf{x}^\top \otimes \mathbf{G} \mid \mathbf{P}) + \mathsf{F}(\mathbf{K}_{D+1}, i) \cdot (\mathbf{0} \mid \mathbf{G}) + \mathbf{e}_{i,D+1}^\top \\ \mathbf{c}_{i,D+2}^\top &= -\mathsf{F}(\mathbf{K}_{D+1}, i) \cdot \mathbf{D} + \mathbf{e}_{i,D+2}^\top \end{aligned}$$

where we omit $\mathbf{c}_{i,j} = \tilde{\mathbf{c}}_{i,j} \leftarrow \mathbb{Z}_q^{2m}$ for all $i \in \{0,1\}^D$ and $j \in [D]$. We prove that they are pseudorandom as in G_3 via the following hybrid arguments:

$$\mathsf{G}_{2.D} \equiv \mathsf{G}_{2.D.1} \approx_c \mathsf{G}_{2.D.2} \approx_c \mathsf{G}_3$$

where

– $\mathsf{G}_{2.D.1}$: Identical to $\mathsf{G}_{2.D}$ except that we carry out the change of variable $\mathbf{F} \mapsto \mathbf{F} + \mathbf{x}^\top \otimes \mathbf{G}$. It is straight-forward to conclude $\mathsf{G}_{2.D} \equiv \mathsf{G}_{2.D.1}$. We remark that, with $\mathbf{F} + \mathbf{x}^\top \otimes \mathbf{G}$ in crs, we only achieve selective security.
– $\mathsf{G}_{2.D.2}$: Identical to $\mathsf{G}_{2.D.1}$ except that we
  - replace $\mathbf{c}_{i,D+1}$ for all $i \in \{0,1\}^D$ with $\tilde{\mathbf{c}}_{i,D+1} \leftarrow \mathbb{Z}_q^{m(\ell+1)}$;

- replace $c_{i,0}$ for all i such that $f_i(\mathbf{x}) = 1$ with $\tilde{c}_{i,0} \leftarrow \mathbb{Z}_q$.

We claim that $\mathsf{G}_{2.D.1} \approx_c \mathsf{G}_{2.D.2}$. This is analogous to $\mathsf{G}_{2.\delta-1} \approx_c \mathsf{G}_{2.\delta}$ by Lemma 1 and Lemma 2 which implies that

$$\{\mathsf{F}(\mathbf{K}_D, i) \cdot (\mathbf{v}|\mathbf{F}|\mathbf{P}) + (e_{i,0}|\mathbf{e}_{i,D+1}^{\top})\}_{i \in \{0,1\}^D} \text{ are pseudorandom.}$$

We note that when generating challenge ciphertext ct, we know f_i for all $i \in \{0,1\}^D$ and $\mathbf{x}$, therefore the game is well-defined.

It remains to prove that $\mathsf{G}_{2.D.2} \approx_c \mathsf{G}_3$.

From $\mathsf{G}_{2.D.2}$ to G_3. Recall that, in $\mathsf{G}_{2.D.2}$, we almost have a pseudorandom ciphertext except the following terms:

$$\begin{aligned} c_{i,0} &= -\textstyle\sum_{j=1}^{D} \tilde{\mathbf{c}}_{i,j}^{\top} \mathbf{w}_{i,j} + \mathbf{c}_{i,D+1}^{\top} \mathbf{w}_{i,D+1} + \mathsf{F}(\mathbf{K}_{D+1}, i) \cdot \mathbf{u}_i + e_{i,0} \quad \forall f_i(\mathbf{x}) = 0 \\ \mathbf{c}_{i,D+2}^{\top} &= -\mathsf{F}(\mathbf{K}_{D+1}, i) \cdot \mathbf{D} + \mathbf{e}_{i,D+2}^{\top} \quad \forall i \in \{0,1\}^D. \end{aligned}$$

We prove that they are pseudorandom via the following hybrid arguments:

$$\mathsf{G}_{2.D.2} \equiv \mathsf{G}_{2.D.2.1} \approx_s \mathsf{G}_{2.D.2.2} \approx_c \mathsf{G}_{2.D.2.3} \approx_s \mathsf{G}_3$$

where

- $\mathsf{G}_{2.D.2.1}$: Identical to $\mathsf{G}_{2.D.2}$ except that we replace $\mathbf{u}_i$ in $c_{i,0}$ such that $f_i(\mathbf{x}) = 0$ with $\mathbf{D}\mathbf{k}_i$ where $\mathbf{k}_i = \mathcal{L}[\mathbf{u}_i]$; namely, we have

  $$c_{i,0} = -\sum_{j=1}^{D} \tilde{\mathbf{c}}_{i,j}^{\top} \mathbf{w}_{i,j} + \mathbf{c}_{i,D+1}^{\top} \mathbf{w}_{i,D+1} + \mathsf{F}(\mathbf{K}_{D+1}, i) \cdot \boxed{\mathbf{D} \cdot \mathbf{k}_i} + e_{i,0} \quad \forall f_i(\mathbf{x}) = 0$$

  It is straight-forward to see that $\mathsf{G}_{2.D.2} \equiv \mathsf{G}_{2.D.2.1}$.Here we use the fact that

  $$f_i(\mathbf{x}) = 0 \implies \mathbf{u}_i \in \mathcal{L}_{\text{hon}} \implies \mathbf{k}_i = \mathcal{L}[\mathbf{u}_i] \neq \perp;$$

  the specification of security game ensures that $\mathbf{u}_i = \mathbf{D}\mathbf{k}_i$.
- $\mathsf{G}_{2.D.2.2}$: Identical to $\mathsf{G}_{2.D.2.1}$ except that we replace $c_{i,0}$ such that $f_i(\mathbf{x}) = 0$ with

  $$c_{i,0} = -\sum_{j=1}^{D} \tilde{\mathbf{c}}_{i,j}^{\top} \mathbf{w}_{i,j} + \mathbf{c}_{i,D+1}^{\top} \mathbf{w}_{i,D+1} + \boxed{\mathbf{c}_{i,D+2}^{\top}} \cdot \mathbf{k}_i + e_{i,0} \quad \forall f_i(\mathbf{x}) = 0$$

  We claim that $\mathsf{G}_{2.D.2.1} \approx_s \mathsf{G}_{2.D.2.2}$. This is analogous to $\mathsf{G}_0 \approx_s \mathsf{G}_1$; we use the fact that $\|\mathbf{k}_i\| = 1$ for all i and rely on smudging lemma with condition $\chi_0 \geq m\lambda^{\omega(1)}\chi_1$.
- $\mathsf{G}_{2.D.2.3}$: Identical to $\mathsf{G}_{2.D.2.2}$ except that we replace $\mathbf{c}_{i,D+2}$ for all $i \in \{0,1\}^D$ with $\tilde{\mathbf{c}}_{i,D+2}^{\top} \leftarrow \mathbb{Z}_q^m$. This is analogous to $\mathsf{G}_{2.\delta-1} \approx_c \mathsf{G}_{2.\delta}$ by Lemma 1 and Lemma 2 which implies that

  $$\{\mathsf{F}(\mathbf{K}_{D+1}, i) \cdot \mathbf{D} + \mathbf{e}_{i,D+2}^{\top}\}_{i \in \{0,1\}^D} \text{ are pseudorandom.}$$

Finally, we claim that $\mathsf{G}_{2.D.2.4} \approx_s \mathsf{G}_3$. This uses the fact that

$$f_i(\mathbf{x}) = 0 \implies \mathbf{u}_i \in \mathcal{L}_{\text{hon}} \implies \mathbf{u}_i \notin \mathcal{C} \wedge \mathbf{k}_i = \mathcal{L}[\mathsf{pk}_i] \neq \perp$$

which means $\mathbf{k}_i$ is sampled honestly and keeps secret from $\mathcal{A}$. Applying the leftover hash lemma with condition $m \geq O(\log q)$ gives us

$$\begin{bmatrix} \mathbf{u}_i \\ \mathbf{c}_{i,D+2}^{\top} \cdot \mathbf{k}_i \end{bmatrix} = \begin{bmatrix} \mathbf{D} \\ \tilde{\mathbf{c}}_{i,D+2}^{\top} \end{bmatrix} \cdot \mathbf{k}_i \approx_s \begin{bmatrix} \tilde{\mathbf{u}}_i \\ \tilde{u}_i \end{bmatrix} \quad \forall f_i(\mathbf{x}) = 0$$

This suffices to hide $c_{i,0}$ such that $f_i(\mathbf{x}) = 0$ and completes the proof.

4 Our Slotted Registered ABE Scheme

This section presents our final slotted Reg-ABE scheme from the slotted Reg-ABE with large ciphertext in Section 3.1 and algebraic obfuscator for mPRF promised in Lemma 1. This yields a slotted Reg-ABE from LWE and evasive LWE. Applying the "power-of-two" transformation (c.f. Section B in **Appendix**) yields our final Reg-ABE from the same set of assumptions. We begin with an overview in the language of mPRF and obfuscation; see Section 1.4 for a brief overview using GGH encoding from scratch.

4.1 Overview

Let $\mathsf{mpk} = (\mathbf{B}_0, \mathbf{B}_1, \mathbf{F}, \mathbf{P}, \mathbf{D}, \mathbf{v}, \mathbf{h}_\epsilon, L)$. Plugging the PRF scheme $\mathsf{PRF}_0 = (\mathsf{PRFGen}_0, \mathsf{PRFKey}_0, \mathsf{PRFEval}_0)$ in Lemma 1 into the slotted Reg-ABE scheme in Section 3.1, a ciphertext for $\mathbf{x} \in \{0,1\}^\ell$ of $\mathsf{m} \in \{0,1\}$ is in the following form

$$\mathsf{ct}_\mathbf{x} = \{\overbrace{c_{i,0}, \mathbf{c}_{i,1}, \ldots, \mathbf{c}_{i,D}, \mathbf{c}_{i,D+1}, \mathbf{c}_{i,D+2}}^{\mathsf{ct}_{\mathbf{x},i}}\}_{i\in\{0,1\}^D}$$

where

$$\begin{aligned}
c_{i,0} &= \mathbf{t}^\top \mathbf{M}_{i_1} \cdots \mathbf{M}_{i_D} \mathbf{K}_0 \mathbf{h}_\epsilon + e_{i,0} + \lfloor q/2 \rfloor \cdot \mathsf{m} \\
\mathbf{c}_{i,j}^\top &= -\mathbf{t}^\top \mathbf{M}_{i_1} \cdots \mathbf{M}_{i_D} \mathbf{K}_{j-1} (\mathbf{B}_0 \mid \mathbf{B}_1) + \mathbf{t}^\top \mathbf{M}_{i_1} \cdots \mathbf{M}_{i_D} \mathbf{K}_j (\bar{i}_j \cdot \mathbf{G} \mid i_j \cdot \mathbf{G}) + \mathbf{e}_{i,j}^\top \\
\mathbf{c}_{i,D+1}^\top &= -\mathbf{t}^\top \mathbf{M}_{i_1} \cdots \mathbf{M}_{i_D} \mathbf{K}_D (\mathbf{F} - \mathbf{x} \otimes \mathbf{G} \mid \mathbf{P}) + \mathbf{t}^\top \mathbf{M}_{i_1} \cdots \mathbf{M}_{i_D} \mathbf{K}_{D+1} (\mathbf{0} \mid \mathbf{G}) + \mathbf{e}_{i,D+1}^\top \\
\mathbf{c}_{i,D+2}^\top &= -\mathbf{t}^\top \mathbf{M}_{i_1} \cdots \mathbf{M}_{i_D} \mathbf{K}_{D+1} \mathbf{D} + \mathbf{e}_{i,D+2}^\top
\end{aligned}$$

Recall that we sample $\mathbf{M}_0, \mathbf{M}_1 \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_0}^{m_0}$, $\mathbf{K}_j \leftarrow \mathbb{Z}_q^{m_0 \times n}$ for all $j \in [0, D+1]$ and

$$\mathbf{e}_i = (e_{i,0}, \mathbf{e}_{i,1}, \ldots, \mathbf{e}_{i,D}, \mathbf{e}_{i,D+1}, \mathbf{e}_{i,D+2}) \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_1}^{1+(2D+\ell+2)m}$$

Motivated by decryption procedure, our idea is to build a ciphertext generator CTGen which (1) returns $\mathsf{ct}_{\mathbf{x},i}$ on input $i \in \{0,1\}^D$ and (2) has much smaller size than enumerating $\mathsf{ct}_{\mathbf{x},i}$ for all $i \in \{0,1\}^D$ (as in Section 3.1). With this, we can simply publish CTGen as the ciphertext ideally.

Rewriting Each Terms. We begin with studying the algebraic structure of $\mathsf{ct}_{\mathbf{x},i}$. First, by linear algebra, we can write all terms without index i as follows:

$$c_{i,0} = (\mathbf{t}^\top, 1) \overbrace{\begin{bmatrix} \mathbf{M}_{i_1} & \\ & 1 \end{bmatrix}}^{\mathbf{N}_{i_1}} \cdots \overbrace{\begin{bmatrix} \mathbf{M}_{i_D} & \\ & 1 \end{bmatrix}}^{\mathbf{N}_{i_D}} \overbrace{\begin{bmatrix} \mathbf{K}_0 \mathbf{h}_\epsilon \\ \lfloor q/2 \rfloor \cdot \mathsf{m} \end{bmatrix}}^{\mathbf{K}_{\text{root,msg}}} + e_{i,0}$$

$$\mathbf{c}_{i,D+1}^\top = (-\mathbf{t}^\top, \mathbf{t}^\top) \overbrace{\begin{bmatrix} \mathbf{M}_{i_1} & \\ & \mathbf{M}_{i_1} \end{bmatrix}}^{\mathbf{Q}_{i_1}} \cdots \overbrace{\begin{bmatrix} \mathbf{M}_{i_D} & \\ & \mathbf{M}_{i_D} \end{bmatrix}}^{\mathbf{Q}_{i_D}} \overbrace{\begin{bmatrix} \mathbf{K}_D (\mathbf{F} - \mathbf{x}^\top \otimes \mathbf{G} | \mathbf{P}) \\ \mathbf{K}_{D+1} (\mathbf{0} | \mathbf{G}) \end{bmatrix}}^{\mathbf{K}_{\text{func}}} + \mathbf{e}_{i,D+1}^\top$$

$$\mathbf{c}_{i,D+2}^\top = -\mathbf{t}^\top \mathbf{M}_{i_1} \cdots \mathbf{M}_{i_D} \overbrace{\mathbf{K}_{D+1} \mathbf{D}}^{\mathbf{K}_{\text{user}}} + \mathbf{e}_{i,D+2}^\top$$

Then, for all $j \in [D]$, we can rewrite terms $\mathbf{c}_{i,j}^\top$ involving index i as follows:

$$(-\mathbf{t}^\top, \mathbf{t}^\top, \mathbf{t}^\top) \overbrace{\begin{bmatrix} \mathbf{M}_{i_1} & & \\ & \mathbf{M}_{i_1} & \\ & & \mathbf{M}_{i_1} \end{bmatrix}}^{\mathbf{T}_{i_1}} \cdots \overbrace{\begin{bmatrix} \mathbf{M}_{i_{j-1}} & & \\ & \mathbf{M}_{i_{j-1}} & \\ & & \mathbf{M}_{i_{j-1}} \end{bmatrix}}^{\mathbf{T}_{i_{j-1}}} \cdot \overbrace{\begin{bmatrix} \mathbf{M}_{i_j} & & \\ & \bar{i}_j \cdot \mathbf{M}_{i_j} & \\ & & i_j \cdot \mathbf{M}_{i_j} \end{bmatrix}}^{\mathbf{V}_{i_j}}$$

$$\cdot \overbrace{\begin{bmatrix} \mathbf{M}_{i_{j+1}} & & \\ & \mathbf{M}_{i_{j+1}} & \\ & & \mathbf{M}_{i_{j+1}} \end{bmatrix}}^{\mathbf{T}_{i_{j+1}}} \cdots \overbrace{\begin{bmatrix} \mathbf{M}_{i_D} & & \\ & \mathbf{M}_{i_D} & \\ & & \mathbf{M}_{i_D} \end{bmatrix}}^{\mathbf{T}_{i_D}} \overbrace{\begin{bmatrix} \mathbf{K}_{j-1}\mathbf{B}_0 & \mathbf{K}_{j-1}\mathbf{B}_1 \\ \mathbf{K}_j\mathbf{G} & \\ & \mathbf{K}_j\mathbf{G} \end{bmatrix}}^{\mathbf{K}_{\mathrm{tree},j}} + \mathbf{e}_{i,j}^{\top}.$$

Note that it is crucial to embed i_j into the j-th matrix $\mathbf{V}_{i_j}$; this makes $\mathbf{V}_b$ quite different from other matrices, i.e., $\mathbf{T}_b$ with $b \in \{0,1\}$.

Rewriting Ciphertexts. Putting them together and defining

$$\begin{array}{rcccccccc} \mathbf{s}^{\top} = & (& \overbrace{\mathbf{t}^{\top}, 1}^{c_{i,0}}, & \overbrace{-\mathbf{t}^{\top}, \mathbf{t}^{\top}, \mathbf{t}^{\top}}^{\mathbf{c}_{i,1}^{\top}}, & \ldots, & \overbrace{-\mathbf{t}^{\top}, \mathbf{t}^{\top}, \mathbf{t}^{\top}}^{\mathbf{c}_{i,D}^{\top}}, & \overbrace{-\mathbf{t}^{\top}, \mathbf{t}^{\top}}^{\mathbf{c}_{i,D+1}^{\top}}, & \overbrace{-\mathbf{t}^{\top}}^{\mathbf{c}_{i,D+2}^{\top}} &) \\ \mathbf{S}_{\iota,b} = & \mathsf{diag}(& \mathbf{N}_b, & \mathbf{T}_b, & \ldots, \mathbf{V}_b, \ldots, & \mathbf{T}_b, & \mathbf{Q}_b, & \mathbf{M}_b &) \\ \mathbf{K} = & \mathsf{diag}(& \mathbf{K}_{\mathrm{root,msg}}, & \mathbf{K}_{\mathrm{tree},1}, & \ldots, & \mathbf{K}_{\mathrm{tree},D}, & \mathbf{K}_{\mathrm{func}}, & \mathbf{K}_{\mathrm{user}} &) \end{array}$$

where $\mathbf{V}_b$ appears at the $(\iota+1)$-th block of $\mathbf{S}_{\iota,b}$ for all $\iota \in [D]$, we have

$$\mathsf{ct}_{\mathbf{x},i} = \mathbf{s}^{\top} \cdot \mathbf{S}_{1,i_1} \cdots \mathbf{S}_{D,i_D} \cdot \mathbf{K} + \mathbf{e}_i^{\top} \quad \forall i \in \{0,1\}^D.$$

Surprisingly, this already gives us a ciphertext generator, denoted by CTGen^*. It is defined by $\mathbf{s}$, $\mathbf{S} = \{\mathbf{S}_{\iota,b}\}_{\iota\in[D],b\in\{0,1\}}$ and $\mathbf{K}$ and works as follows:

$$\mathsf{CTGen}^*(i) = \mathbf{s}^{\top} \cdot \mathbf{S}_{1,i_1} \cdots \mathbf{S}_{D,i_D} \cdot \mathbf{K} + \mathbf{e}_i^{\top}.$$

However, we *can not* simply publish $(\mathbf{s}, \mathbf{S}, \mathbf{K})$ as the ciphertext because this will leak m. Furthermore, there are two more issues: (1) we can not ask the decryptor to sample $\mathbf{e}_i$; (2) the decryptor might manipulate those vectors/matrices in an unexpected way.

Final Step. To fix the above issues, we may choose to obfuscating CTGen^*. However a general $i\mathcal{O}$ is highly impractical [27,28] and will make a non-black-box use of underlying algebraic objects. Fortunately, the following observation rescues us: Without the noise $\mathbf{e}_i^{\top}$,

$$\mathsf{CTGen}^{**}(i) = \mathbf{s}^{\top} \cdot \mathbf{S}_{1,i_1} \cdots \mathbf{S}_{D,i_D}\mathbf{K}$$

looks like the evaluation algorithm of a mPRF. In fact, Theorem 2 basically says that CTGen^{**} is actually a σ_1-mPRF against some aux depending on security game with $\mathcal{A}$. This allows us to employ the *algebraic* obfuscator Obf recently proposed in [30] (c.f. Theorem 1), the ciphertext will be

$$\mathsf{ct}_{\mathbf{x}} = \mathsf{CTGen} \leftarrow \mathsf{Obf}(\mathsf{CTGen}^{**})$$

Decryption works as follows: run $\mathsf{ct}_{\mathbf{x},i} = \mathsf{CTGen}(i)$ for some i and invoke IndDec to recover the message as in Section 3.1. We note that the resulting scheme is a *black-box* construction and avoid the use of Barrington Theorem.

4.2 Matrix PRF Induced by Slotted Reg-ABE in Section 3.1

Motivated by Section 4.1, we describe a mPRF that generates $\mathsf{ct}_{\mathbf{x},i}$ as follows:

– $\mathsf{PRFGen}(w, \sigma_0)$: Output $\mathbf{S} = (\mathbf{M}_0, \mathbf{M}_1) \leftarrow \mathsf{PRFGen}_0(w, \sigma_0)$.

– $\mathsf{PRFKey}(w, \mathsf{mpk}, \mathbf{x}, \mathsf{m})$: Let $\mathsf{mpk} = (\mathbf{B}_0, \mathbf{B}_1, \mathbf{F}, \mathbf{P}, \mathbf{D}, \mathbf{v}, \mathbf{h}_\epsilon, L = 2^D)$. Sample

$$\mathbf{K}_j \leftarrow \mathsf{PRFKey}_0(w, n) \quad \forall j \in [0, D+1]$$

and define

$$\mathbf{K}_{\mathrm{root,msg}} = \begin{bmatrix} \mathbf{K}_0\mathbf{h}_\epsilon \\ \lfloor q/2 \rfloor \cdot \mathsf{m} \end{bmatrix} \qquad \mathbf{K}_{\mathrm{tree},j} = \begin{bmatrix} \mathbf{K}_{j-1}(\mathbf{B}_0|\mathbf{B}_1) \\ \mathbf{I}_2 \otimes \mathbf{K}_j\mathbf{G} \end{bmatrix} \quad \forall j \in [D]$$

$$\mathbf{K}_{\mathrm{func}} = \begin{bmatrix} \mathbf{K}_D(\mathbf{F} - \mathbf{x} \otimes \mathbf{G}|\mathbf{P}) \\ \mathbf{K}_{D+1}(\mathbf{0}|\mathbf{G}) \end{bmatrix} \qquad \mathbf{K}_{\mathrm{user}} = \mathbf{K}_{D+1}\mathbf{D}$$

Output

$$\mathbf{K} = \mathsf{diag}(\mathbf{K}_{\mathrm{root,msg}}, \mathbf{K}_{\mathrm{tree},1}, \ldots, \mathbf{K}_{\mathrm{tree},D}, \mathbf{K}_{\mathrm{func}}, \mathbf{K}_{D+1}\mathbf{D}).$$

We assume that one can efficiently extract D from $\mathbf{K}$.

– $\mathsf{PRFEval}(\mathbf{S}, \mathbf{K}, \mathbf{x})$: Parse $\mathbf{S} = (\mathbf{M}_0, \mathbf{M}_1)$. For all $b \in \{0, 1\}$, define

$$\mathbf{N}_b = \mathsf{diag}(\mathbf{M}_b, 1),\ \ \mathbf{Q}_b = \mathbf{I}_2 \otimes \mathbf{M}_b,\ \ \mathbf{T}_b = \mathbf{I}_3 \otimes \mathbf{M}_b,\ \ \mathbf{V}_b = \mathsf{diag}(1, \bar{b}, b) \otimes \mathbf{M}_b$$

and for all $\iota \in [D]$ and $b \in \{0, 1\}$, define

$$\mathbf{s}^\top = (\mathbf{t}^\top, 1, \mathbf{1}_\ell^\top \otimes (-\mathbf{t}^\top, \mathbf{t}^\top, \mathbf{t}^\top), -\mathbf{t}^\top, \mathbf{t}^\top, -\mathbf{t}^\top)$$
$$\mathbf{S}_{\iota,b} = \mathsf{diag}(\mathbf{N}_b, \mathbf{I}_{\iota-1} \otimes \mathbf{T}_b, \mathbf{V}_b, \mathbf{I}_{\ell-\iota} \otimes \mathbf{T}_b, \mathbf{Q}_b, \mathbf{M}_b)$$

Output

$$\mathbf{s}^\top\mathbf{S}_{1,i_1} \cdots \mathbf{S}_{D,i_D}\mathbf{K}.$$

For security, we define the following algorithm that generates $\mathbf{K}$ along with aux:

– $\mathsf{PRFKey}^*(w, \mathsf{par}^* = \mathcal{A})$: Run $\mathcal{A}$ with random coin r as follows:
  1. Given $\mathbf{x} \leftarrow \mathcal{A}$, sample $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^d, 1^\ell)$;
  2. Send crs to $\mathcal{A}$ and maintain OGen and OCor with responses $\mathsf{rsp} \in \{0,1\}^*$;
  3. Receiving m and $\{\mathsf{pk}_i, f_i\}_{i \in \{0,1\}^D}$, run $\mathsf{mpk} \leftarrow \mathsf{Agg}(\mathsf{crs}, \{\mathsf{pk}_i, f_i\}_{i \in \{0,1\}^D})$.

  Output

$$\mathsf{aux}_\mathcal{A} = (r, \mathbf{x}, \mathsf{crs}, \mathsf{rsp}, \mathsf{mpk}) \quad \text{and} \quad \mathbf{K}_\mathcal{A} \leftarrow \mathsf{PRFKey}(m_0, \mathsf{mpk}, \mathbf{x}, \mathsf{m}).$$

Security. Before we proceed to describe the security of (PRFGen, PRFKey, PRFEval) defined above, it is useful to rewrite Enc of the slotted Reg-ABE in Section 3.1 with it; other algorithms are not relevant.

– $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}, \mathsf{m})$: Let $L = 2^D$ (read from mpk). Sample

$$\mathbf{S} \leftarrow \mathsf{PRFGen}(m_0, \sigma_0) \quad \text{and} \quad \mathbf{K} \leftarrow \mathsf{PRFKey}(m_0, \mathsf{mpk}, \mathbf{x}, \mathsf{m}).$$

Output

$$\mathsf{ct}_\mathbf{x} = \{\mathsf{PRFEval}(\mathbf{K}, i) + \mathbf{e}_i^\top\}_{i \in \{0,1\}^D}$$

where $\mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_1}^{1+(2D+\ell+2)m}$ for all $i \in \{0,1\}^D$.

This suggests Lemma 3 which is a straight-forward implication of Theorem 2.

Lemma 3. *Under the same assumptions mentioned in Theorem 2 and conditions in Section 3.2,* (PRFGen, PRFKey, PRFEval) *is a σ_1-mPRF against* PRFKey^*. *In particular, for all P.P.T. algorithms $\mathcal{A}, \mathcal{B}$,*

$$\Pr[\mathcal{B}(1^\lambda, \boxed{\{\mathsf{PRFEval}(\mathbf{S}, \mathbf{K}_\mathcal{A}, i)\}_{i \in \{0,1\}^D}}, \mathsf{aux}_\mathcal{A}) = 1] - \Pr[\mathcal{B}(1^\lambda, \{\tilde{\mathbf{c}}_i\}_{i \in \{0,1\}^D}, \mathsf{aux}_\mathcal{A}) = 1] = \varepsilon(\lambda)$$

where $\mathbf{S} \leftarrow \mathsf{PRFGen}(m_0, \sigma_0)$, $(\mathbf{K}_\mathcal{A}, \mathsf{aux}_\mathcal{A}) \leftarrow \mathsf{PRFKey}^*(m_0, \mathcal{A})$ *and* $\tilde{\mathbf{c}}_i \leftarrow \mathbb{Z}_q^{1+(2D+\ell+2)m}$ *for all* $i \in \{0,1\}^D$.

4.3 Scheme

Our final slotted Reg-ABE is identical to the scheme presented in Section 3.1 except that we replace Enc and Dec with Enc^* and Dec^*, respectively.

- $\mathsf{Enc}^*(\mathsf{mpk}, \mathbf{x}, \mathsf{m})$: Let $L = 2^D$ (read from mpk). Sample

$$\mathbf{S} \leftarrow \mathsf{PRFGen}(m_0, \sigma_0) \quad \text{and} \quad \mathbf{K} \leftarrow \mathsf{PRFKey}(m_0, \mathsf{mpk}, \mathbf{x}, \mathsf{m}).$$

  Output

$$\mathsf{ct}_{\mathbf{x}} = \boxed{\mathsf{CTGen} \leftarrow \mathsf{Obf}(1^\lambda, 1^D, \mathbf{S}, \mathbf{K}).}$$

- $\mathsf{Dec}^*(\mathsf{sk}_{i^*}, \mathsf{hsk}_{i^*}, \mathsf{ct}_{\mathbf{x}})$: Let $\mathsf{ct} = \mathsf{CTGen}$ and run

$$\mathsf{ct}_{\mathbf{x},i^*} \leftarrow \boxed{\mathsf{CTGen}(i^*)}.$$

  Return $\mathsf{IndDec}(\mathsf{sk}_{i^*}, \mathsf{hsk}_{i^*}, \mathsf{ct}_{\mathbf{x},i^*})$.

We highlight the differences with Enc and Dec with boxes.

Parameter Selection. We set

$$n = \mathsf{poly}(\lambda), \qquad m = O(n^{1+c}), \qquad m_0 = 6n \log q,$$
$$q = \mathsf{poly}(\ell) \cdot n^{\mathsf{poly}(D,d)}, \quad \sigma_0 = \Omega(\sqrt{n \log q}), \quad \sigma_1 = O(\ell) \cdot (n \log q)^{\mathsf{poly}(D,d)}$$

so that they satisfy conditions in Section 3.2 and the following additional ones:

$$q/4 \geq \ell m^{O(d)} (D m_0 \sqrt{\lambda} \sigma_0)^D \qquad \text{// correctness}$$
$$\sigma_1 = 2^{D^3} (n^2 \sqrt{2n})^{D+1} \qquad \text{// obfuscation}$$

Correctness & Compactness. Observe that PRF has width $3Dm_0 + 4m_0 + 1$ and length D; furthermore, with overwhelming probability, each entry of $\mathbf{s}$ and $\mathbf{S}$ is bounded by $B = \sqrt{\lambda}\sigma_0$. By Theorem 1, for all $i^* \in \{0,1\}^D$, we have that

$$\mathsf{ct}_{\mathbf{x},i^*} = (c_{i^*,0}, \mathbf{c}_{i^*,1}, \ldots, \mathbf{c}_{i^*,D}, \mathbf{c}_{i^*,D+1}, \mathbf{c}_{i^*,D+2}) \leftarrow \mathsf{CTGen}(i^*)$$

where $c_{i^*,0}, \mathbf{c}_{i^*,1}, \ldots, \mathbf{c}_{i^*,D+2}$ are defined as in (5) except that we have

$$\mathbf{e}_i = (e_{i,0}, \mathbf{e}_{i,1}, \ldots, \mathbf{e}_{i,D}, \mathbf{e}_{i,D+1}, \mathbf{e}_{i,D+2}) \in [-B', B']^{1+(2D+\ell+2)m}$$

with $B' = (Dm_0\sqrt{\lambda}\sigma_0)^D$. Analogous to (13), correctness holds under the following condition:

$$(1 + 2mD + m(\ell+1)m^{O(d)} + m)B \leq \ell m^{O(d)} (Dm_0 \sqrt{\lambda}\sigma_0)^D \leq q/4.$$

Furthermore we have

$$|\mathsf{CTGen}| = O(D^4 m_0^2 m \ell \log q) = \mathsf{poly}(D, m_0, m, \ell, \log q).$$

Security. We prove the following theorem.

Theorem 3. *Under $\mathsf{LWE}_{n,\mathsf{poly}(n),q,\sqrt{2n}}$ and $\mathsf{evLWE}_{\sigma_1,\sigma_1}$ where $\sigma_1 = 2^{D^3}(n^2\sqrt{2n})^{D+1}$ with conditions in Section 3.2 and Section 4.3, our slotted Reg-ABE scheme presented in Section 4.3 is selectively secure (c.f. Section 2.4).*

Before we proceed, we present the following lemma which immediately follows from Lemma 3 and Theorem 1. We omit the proof.

Lemma 4. *Under the same assumptions mentioned in Theorem 1 and Theorem 2 and conditions in Section 3.2, for all P.P.T.* $\mathcal{A}, \mathcal{B}$*,*

$$|\Pr[\mathcal{B}(1^\lambda, \boxed{\mathsf{CTGen}}, \mathsf{aux}_\mathcal{A}) = 1] - \Pr[\mathcal{B}(1^\lambda, \mathsf{CTGen}_\$, \mathsf{aux}_\mathcal{A}) = 1]| = \varepsilon(\lambda)$$

where $\mathbf{S} \leftarrow \mathsf{PRFGen}(m_0, \sigma_0)$*,* $(\mathbf{K}_\mathcal{A}, \mathsf{aux}_\mathcal{A}) \leftarrow \mathsf{PRFKey}^*(m_0, \mathcal{A})$*,* $\mathsf{CTGen} \leftarrow \mathsf{Obf}(1^\lambda, 1^D, \mathbf{S}, \mathbf{K}_\mathcal{A})$ *and* $\mathsf{CTGen}_\$ \leftarrow \mathcal{D}$*.*

Proof (of Theorem 3). We prove via hybrid arguments:

- G_0: The real game.
- G_1: Identical to G_0 except that we replace ct with $\mathsf{CTGen}_\$ \leftarrow \mathcal{D}$.

Clearly, in G_1, ct is independent of challenge message pair. It remains to prove that $\mathsf{G}_0 \approx_c \mathsf{G}_1$. It suffices to prove that if there exists $\mathcal{A}$ who can distinguish G_0 and G_1, then we have $\mathcal{B}$ against Lemma 4 with respect to $\mathcal{A}$. First, $\mathcal{B}$ runs $\mathcal{A}$ as in $\mathsf{PRFKey}^*(\mathcal{A})$ with the random coin r in $\mathsf{aux}_\mathcal{A}$. Then, $\mathcal{B}$ sends the challenge to $\mathcal{A}$, which is either $\mathsf{CTGen} \leftarrow \mathsf{Obf}(1^\lambda, 1^D, \mathbf{S}, \mathbf{K}_\mathcal{A})$ or $\mathsf{CTGen}_\$ \leftarrow \mathcal{D}$, and returns the output of $\mathcal{A}$. Observe that the advantage of $\mathcal{B}$ in distinguishing CTGen and $\mathsf{CTGen}_\$$ is exactly the advantage of $\mathcal{A}$ in distinguishing G_0 and G_0. This proves the theorem. ◻

References

1. Nuttapong Attrapadung and Junichi Tomida. A modular approach to registered ABE for unbounded predicates. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 280–316. Springer, Cham, August 2024. 1
2. Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 719–737. Springer, Berlin, Heidelberg, April 2012. 9
3. Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 533–556. Springer, 2014. 8
4. Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic PRFs and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 410–428. Springer, Berlin, Heidelberg, August 2013. 6, 9
5. Pedro Branco, Russell W. F. Lai, Monosij Maitra, Giulio Malavolta, Ahmadreza Rahimi, and Ivy K. Y. Woo. Traitor tracing without trusted authority from registered functional encryption. Cryptology ePrint Archive, Report 2024/179, 2024. 2
6. Jeffrey Champion and David J. Wu. Distributed broadcast encryption from lattices. TCC, 2024. 2
7. Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 577–607. Springer, Cham, August 2018. 1, 2, 6
8. Valerio Cini and Hoeteck Wee. ABE for circuits with poly (λ) -sized keys from LWE. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 435–446. IEEE, 2023. 4
9. Kelong Cong, Karim Eldefrawy, and Nigel P. Smart. Optimizing registration based encryption. In Maura B. Paterson, editor, *Cryptography and Coding - 18th IMA International Conference, IMACC 2021, Virtual Event, December 14-15, 2021, Proceedings*, volume 13129 of *Lecture Notes in Computer Science*, pages 129–157. Springer, 2021. 1
10. Pratish Datta, Tapas Pal, and Shota Yamada. Registered FE beyond predicates: (attribute-based) linear functions and more. Cryptology ePrint Archive, Report 2023/457, 2023. 2
11. Leo de Castro and Chris Peikert. Functional commitments for all functions, with transparent setup and from SIS. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 287–320. Springer, Cham, April 2023. 3
12. Nico Döttling, Dimitris Kolonelos, Russell W. F. Lai, Chuanwei Lin, Giulio Malavolta, and Ahmadreza Rahimi. Efficient laconic cryptography from learning with errors. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 417–446. Springer, Cham, April 2023. 1, 2, 3, 5

13. Dario Fiore, Dimitris Kolonelos, and Paola de Perthuis. Cuckoo commitments: Registration-based encryption and key-value map commitments for large spaces. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part V*, volume 14442 of *LNCS*, pages 166–200. Springer, Singapore, December 2023. 1, 2
14. Danilo Francati, Daniele Friolo, Monosij Maitra, Giulio Malavolta, Ahmadreza Rahimi, and Daniele Venturi. Registered (inner-product) functional encryption. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part V*, volume 14442 of *LNCS*, pages 98–133. Springer, Singapore, December 2023. 1
15. Cody Freitag, Brent Waters, and David J. Wu. How to use (plain) witness encryption: Registered ABE, flexible broadcast, and more. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 498–531. Springer, Cham, August 2023. 1, 2, 27
16. Rachit Garg, George Lu, Brent Waters, and David J. Wu. Realizing flexible broadcast encryption: How to broadcast to a public-key directory. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023*, pages 1093–1107. ACM Press, November 2023. 1
17. Rachit Garg, George Lu, Brent Waters, and David J. Wu. Reducing the CRS size in registered ABE systems. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 143–177. Springer, Cham, August 2024. 1
18. Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013. 1
19. Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ahmadreza Rahimi. Registration-based encryption: Removing private-key generator from IBE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 689–718. Springer, Cham, November 2018. 1, 2
20. Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, Ahmadreza Rahimi, and Sruthi Sekar. Registration-based encryption from standard assumptions. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 63–93. Springer, Cham, April 2019. 1
21. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008. 3
22. Noemi Glaeser, Dimitris Kolonelos, Giulio Malavolta, and Ahmadreza Rahimi. Efficient registration-based encryption. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023*, pages 1065–1079. ACM Press, November 2023. 1
23. Rishab Goyal and Satyanarayana Vusirikala. Verifiable registration-based encryption. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 621–651. Springer, Cham, August 2020. 1
24. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309. 1
25. Susan Hohenberger, George Lu, Brent Waters, and David J. Wu. Registered attribute-based encryption. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 511–542. Springer, Cham, April 2023. 1, 2, 11, 27
26. Pavel Hubacek and Daniel Wichs. On the communication complexity of secure function evaluation with long output. In Tim Roughgarden, editor, *ITCS 2015*, pages 163–172. ACM, January 2015. 2
27. Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd ACM STOC*, pages 60–73. ACM Press, June 2021. 20
28. Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part I*, volume 13275 of *LNCS*, pages 670–699. Springer, Cham, May / June 2022. 20
29. Hanjun Li, Huijia Lin, and Ji Luo. ABE for circuits with constant-size secret keys and adaptive security. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 680–710. Springer, Cham, November 2022. 4
30. Surya Mathialagan, Spencer Peters, and Vinod Vaikuntanathan. Adaptively sound zero-knowledge SNARKs for UP. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part X*, volume 14929 of *LNCS*, pages 38–71. Springer, Cham, August 2024. 7, 8, 10, 20, 26
31. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Berlin, Heidelberg, April 2012. 7, 10
32. Willy Quach, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and applications. In Mikkel Thorup, editor, *59th FOCS*, pages 859–870. IEEE Computer Society Press, October 2018. 3

33. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. 8
34. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Berlin, Heidelberg, May 2005. 1
35. Rotem Tsabary. Fully secure attribute-based encryption for t-CNF from LWE. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 62–85. Springer, Cham, August 2019. 8
36. Rotem Tsabary. Candidate witness encryption from lattice techniques. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 535–559. Springer, Cham, August 2022. 1, 2, 10
37. Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-IO from evasive LWE. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part I*, volume 13791 of *LNCS*, pages 195–221. Springer, Cham, December 2022. 1, 2, 7
38. Brent Waters, Hoeteck Wee, and David J. Wu. Multi-authority ABE from lattices without random oracles. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 651–679. Springer, Cham, November 2022. 9
39. Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 217–241. Springer, Cham, May / June 2022. 1, 2, 10
40. Hoeteck Wee. Circuit ABE with poly(depth, λ)-sized ciphertexts and keys from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 178–209. Springer, Cham, August 2024. 2
41. Ziqi Zhu, Jiangtao Li, Kai Zhang, Junqing Gong, and Haifeng Qian. Registered functional encryptions from pairings. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 373–402. Springer, Cham, May 2024. 1, 2
42. Ziqi Zhu, Kai Zhang, Junqing Gong, and Haifeng Qian. Registered ABE via predicate encodings. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part V*, volume 14442 of *LNCS*, pages 66–97. Springer, Singapore, December 2023. 1

Appendix

A Concrete Obfuscator from Evasive LWE

The obfuscator described in [30] works as follows:

– $\mathsf{Obf}(1^\lambda, 1^\ell, \mathbf{S}, \mathbf{K})$: Compute

$$\mathbf{m} \in \mathbb{Z}^w \quad \text{and} \quad \mathbf{M}_{i,b} \in \mathbb{Z}^{w \times w} \quad \forall i \in [\ell], b \in \{0,1\}.$$

This is ensured by the definition of mPRF, c.f., Section 2.2. For all $i \in [\ell]$ and $b \in \{0,1\}$, sample

$$\widehat{\mathbf{S}}_{i,b} = \mathsf{diag}(\mathbf{M}_{i,b}, \widetilde{\mathbf{S}}_{i,b}) \in \mathbb{Z}^{(w+n)\times(w+n)} \quad \text{where} \quad \widetilde{\mathbf{S}}_{i,b} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{n\times n}$$

Define

$$\mathbf{u}^\top = (\mathbf{m}^\top | \mathbf{1}_n^\top) \in \mathbb{Z}^{1\times(w+n)} \quad \text{and} \quad \mathbf{V} = \begin{bmatrix} \mathbf{K} \\ \mathbf{0}_{n\times m} \end{bmatrix} \in \mathbb{Z}_q^{(w+n)\times m}$$

Sample $(\mathbf{A}_i, \mathbf{A}_i^{-1}) \leftarrow \mathbb{Z}_q^{(w+n)\times W} \times \mathbb{Z}^{W\times W}$ for all $i \in [\ell-1]$ and

$$\mathbf{e}_{1,b}^\top \leftarrow \mathcal{D}_{Z,\chi}^{1\times W}, \quad \mathbf{E}_{2,b}, \ldots, \mathbf{E}_{\ell-1,b} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{(w+n)\times W}, \quad \mathbf{E}_{\ell,b} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{(w+n)\times m} \quad \forall b \in \{0,1\}.$$

Output

$$\{ \overbrace{\mathbf{u}^\top \widehat{\mathbf{S}}_{1,b}\mathbf{A}_1 + \mathbf{E}_{1,b}}^{\mathbf{D}_{1,b}}, \overbrace{\mathbf{A}_{i-1}^{-1}(\widehat{\mathbf{S}}_{i,b}\mathbf{A}_i + \mathbf{E}_{i,b})}^{\mathbf{B}_{i,b}}, \overbrace{\mathbf{A}_{\ell-1}^{-1}(\widehat{\mathbf{S}}_{\ell,b}\mathbf{V} + \mathbf{E}_{\ell,b})}^{\mathbf{D}_{\ell,b}} \}_{b\in\{0,1\}}.$$

B Registered Attribute-Based Encryption

Algorithms. Let $d, \ell \in \mathbb{N}$. A *registered attribute-based encryption* (Reg-ABE) for circuits is a tuple of algorithms with the following syntax:

	λ : security parameter
	d : depth of circuits
	ℓ : input length of circuits
$\mathsf{Setup}(1^\lambda, 1^d, 1^\ell) \to \mathsf{crs}, \mathsf{mpk}, \mathsf{aux}$	crs : common reference string
$\mathsf{Gen}(\mathsf{crs}) \to (\mathsf{pk}, \mathsf{sk})$	$\mathsf{mpk}, \mathsf{mpk}'$: master public key
$\mathsf{Reg}^{\mathsf{aux}}(\mathsf{crs}, \mathsf{mpk}, f, \mathsf{pk}) \to \mathsf{mpk}'$	aux : auxiliary information
$\mathsf{Upd}^{\mathsf{aux}}(\mathsf{crs}, \mathsf{mpk}, f, \mathsf{pk}) \to \mathsf{hsk}$	pk : user's public key
$\mathsf{Enc}(\mathsf{mpk}, x, \mathsf{m}) \to \mathsf{ct}$	sk : user's secret key
$\mathsf{Dec}(\mathsf{sk}, \mathsf{hsk}, \mathsf{ct}) \to \mathsf{m}/\bot/\mathsf{getupd}$	hsk : helper secret key
	$f \in \mathcal{C}_{d,\ell}$: function
	m : message
	$x \in \{0,1\}^\ell$: input to function

We require that Reg and Upd are deterministic.

Correctness. For all λ, d, ℓ and all (unbounded) adversary $\mathcal{A}$, it holds that

$$\Pr\left[\mathsf{m} = \mathsf{m}^* \left| \begin{array}{r} (\mathsf{crs}, \mathsf{mpk}, \mathsf{aux}) \leftarrow \mathsf{Setup}(1^\lambda, 1^d, 1^\ell) \\ (x^*, \mathsf{m}^*, f^*) \leftarrow \mathcal{A}^{\mathsf{mpk},\mathsf{aux},\mathsf{ORegH},\mathsf{ORegM}}(\mathsf{crs}) \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, x^*, \mathsf{m}^*) \\ (\mathsf{sk}^*, \mathsf{hsk}^*, f^*) \leftarrow \mathsf{hon}[\mathsf{pk}^*] \\ \mathsf{m} \ = \ \mathsf{Dec}(\mathsf{sk}^*, \mathsf{hsk}^*, \mathsf{ct}^*) \end{array} \right.\right] \geq 1 - \varepsilon(\lambda)$$

where we initialize $\mathsf{mpk} = \bot$, $\mathsf{aux} = \bot$ and oracles work as follows:

– $\mathsf{ORegH}(f)$: Run $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{crs})$ and update $\mathsf{mpk} \leftarrow \mathsf{Reg}^{\mathsf{aux}}(\mathsf{crs}, \mathsf{mpk}, f, \mathsf{pk})$, $\mathsf{hsk} \leftarrow \mathsf{Upd}^{\mathsf{aux}}(\mathsf{crs}, \mathsf{mpk}, f, \mathsf{pk})$, return pk and record $\mathsf{hon}[\mathsf{pk}] = (\mathsf{sk}, \mathsf{hsk}, f)$.
– $\mathsf{ORegM}(f, \mathsf{pk})$: If $\mathsf{hon}[\mathsf{pk}] = \bot$, update $\mathsf{mpk} \leftarrow \mathsf{Reg}^{\mathsf{aux}}(\mathsf{crs}, \mathsf{mpk}, f, \mathsf{pk})$ and $\mathsf{mal} = \mathsf{mal} \cup \{f\}$, return $\mathsf{hsk} \leftarrow \mathsf{Upd}^{\mathsf{aux}}(\mathsf{crs}, \mathsf{mpk}, f, \mathsf{pk})$.

and we require that $\mathsf{hon}[\mathsf{pk}^*] \neq \bot$ and $f^*(x^*) = 0$ in the fourth line.

Security. For all P.P.T. stateful adversary $\mathcal{A}$,

$$\Pr\left[\beta = \beta' \left| \begin{array}{l} x^* \leftarrow \mathcal{A};\ (\mathsf{crs}, \mathsf{mpk}, \mathsf{aux}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}) \\ (\mathsf{m}_0^*, \mathsf{m}_1^*) \leftarrow \mathcal{A}^{\mathsf{mpk},\mathsf{aux},\mathsf{ORegH},\mathsf{ORegM},\mathsf{OCorHK}}(\mathsf{crs}) \\ \beta \leftarrow \{0,1\},\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, x^*, \mathsf{m}_\beta^*) \\ \beta' \leftarrow \mathcal{A}^{\mathsf{mpk},\mathsf{aux},\mathsf{ORegH},\mathsf{ORegM},\mathsf{OCorHK}}(\mathsf{ct}^*) \end{array} \right.\right] - \frac{1}{2} \leq \varepsilon(\lambda)$$

where oracles ORegH, ORegM work as above, and oracle OCorHK works as follows:

– $\mathsf{OCorHK}(\mathsf{pk})$: Let $\mathsf{hon}[\mathsf{pk}] = (\mathsf{sk}, \mathsf{hsk}, f)$, set $\mathsf{cor} = \mathsf{cor} \cup \{f\}$ and return sk.

and we require that for all $f \in \mathsf{mal} \cup \mathsf{cor}$, $f(x^*) = 1$.

Transformation. In [25], it is proved that slotted Reg-ABE generically implies Reg-ABE preserving efficiency and security via so-called "power-of-two" transformation. We restate their theorem. The details of transformation can be found in **Construction 6.1** in Section 6 in [25].

Theorem 4 ([25,15]). *Assume a L-slotted Reg-ABE scheme for $C_{d,\ell}$ achieving adaptive (resp. selective) security with* $|\mathsf{crs}|$, $|\mathsf{mpk}|$, $|\mathsf{ct}|$, *and* $|\mathsf{hsk}|$ *bounded by* $\mathsf{polylog}(\lambda, L)$. *There exists a Reg-ABE scheme for $C_{d,\ell}$ achieving adaptive (resp. selective) security with the same efficiency profile, asymptotically, and the number of updates is bounded by* $\mathsf{polylog}(\lambda, L)$.

C Details of Correctness in Section 3.3

For completeness, we verify (9), (10), (11), (12) as follows. First, we have

$$z_1 = \mathbf{c}_1^\top \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{1,0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{1,1}) \end{bmatrix} = (-\mathbf{s}_0^\top(\mathbf{B}_0 \mid \mathbf{B}_1) + \mathbf{s}_1^\top(\bar{i}_1^* \cdot \mathbf{G} \mid i_1^* \cdot \mathbf{G}) + \mathbf{e}_1^\top) \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{1,0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{1,1}) \end{bmatrix}$$

$$\stackrel{(6)}{=} (-\mathbf{s}_0^\top \mathbf{h}_\epsilon + \mathbf{s}_1^\top(\bar{i}_1^* \cdot \mathbf{h}_{1,0} + i_1^* \cdot \mathbf{h}_{1,1})) + \mathbf{e}_1^\top \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{1,0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{1,1}) \end{bmatrix}$$

$$= (-\mathbf{s}_0^\top \mathbf{h}_\epsilon + \mathbf{s}_1^\top \mathbf{h}_{1,i_1^*}) + \mathbf{e}_1^\top \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{1,0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{1,1}) \end{bmatrix}$$

The last equality follows from the fact $\bar{x} \cdot \mathbf{h}_0 + x \cdot \mathbf{h}_1 = \mathbf{h}_x$ for all $x \in \{0,1\}$. Analogously, we have

$$\begin{aligned} z_j &= \mathbf{c}_j^\top \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{j,0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{j,1}) \end{bmatrix} \\ &= (-\mathbf{s}_{j-1}^\top(\mathbf{B}_0 \mid \mathbf{B}_1) + \mathbf{s}_j^\top(\bar{i}_j^* \cdot \mathbf{G} \mid i_j^* \cdot \mathbf{G}) + \mathbf{e}_j^\top) \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{j,0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{j,1}) \end{bmatrix} \\ &\overset{(7)}{=} (-\mathbf{s}_{j-1}^\top \mathbf{h}_{j-1,i_{j-1}^*} + \mathbf{s}_j^\top(\bar{i}_j^* \cdot \mathbf{h}_{j,0} + i_j^* \cdot \mathbf{h}_{j,1})) + \mathbf{e}_j^\top \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{j,0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{j,1}) \end{bmatrix} \\ &= (-\mathbf{s}_{j-1}^\top \mathbf{h}_{j-1,i_{j-1}^*} + \mathbf{s}_j^\top \mathbf{h}_{j,i_j^*}) + \mathbf{e}_j^\top \begin{bmatrix} \mathbf{G}^{-1}(\mathbf{h}_{j,0}) \\ \mathbf{G}^{-1}(\mathbf{h}_{j,1}) \end{bmatrix} \end{aligned}$$

and also

$$\begin{aligned} z_{D+1} &= \mathbf{c}_{D+1}^\top \begin{bmatrix} \mathbf{H}_{f,\mathbf{x}} \mathbf{G}^{-1}(\mathbf{v}) \\ \mathbf{G}^{-1}(\mathbf{u}) \end{bmatrix} \\ &= (-\mathbf{s}_D^\top(\mathbf{F} - \mathbf{x}^\top \otimes \mathbf{G} \mid \mathbf{P}) + \mathbf{s}_{D+1}^\top(\mathbf{0} \mid \mathbf{G}) + \mathbf{e}_{D+1}^\top) \begin{bmatrix} \mathbf{H}_{f,\mathbf{x}} \mathbf{G}^{-1}(\mathbf{v}) \\ \mathbf{G}^{-1}(\mathbf{u}) \end{bmatrix} \\ &\overset{(4)}{=} -\mathbf{s}_D^\top(\mathbf{F}\mathbf{H}_f \mathbf{G}^{-1}(\mathbf{v}) - f(\mathbf{x}) \cdot \mathbf{v} + \mathbf{P}\mathbf{G}^{-1}(\mathbf{u})) + \mathbf{s}_{D+1}^\top \mathbf{u} + \mathbf{e}_{D+1}^\top \begin{bmatrix} \mathbf{H}_{f,\mathbf{x}} \mathbf{G}^{-1}(\mathbf{v}) \\ \mathbf{G}^{-1}(\mathbf{u}) \end{bmatrix} \\ &\overset{(8)}{=} -\mathbf{s}_D^\top \mathbf{h}_{D,i_D^*}^* + f(\mathbf{x}) \cdot \mathbf{s}_D^\top \mathbf{v} + \mathbf{s}_{D+1}^\top \mathbf{u} + \mathbf{e}_{D+1}^\top \begin{bmatrix} \mathbf{H}_{f,\mathbf{x}} \mathbf{G}^{-1}(\mathbf{v}) \\ \mathbf{G}^{-1}(\mathbf{u}) \end{bmatrix} \end{aligned}$$

Finally, it is straight-forward to see that

$$z_{D+2} = (-\mathbf{s}_{D+1}^\top \mathbf{D} + \mathbf{e}_{D+2}^\top)\mathbf{k} = -\mathbf{s}_{D+1}^\top \mathbf{u} + \mathbf{e}_{D+2}^\top \mathbf{k}.$$

Table of Contents