Revocable Predicate Encryption from Lattices

San Ling, Khoa Nguyen, Huaxiong Wang, Juanyang Zhang

Division of Mathematical Sciences, School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore
{lingsan,hxwang,khoantt,zh0078ng}@ntu.edu.sg

Abstract. Predicate encryption, formalized by Katz, Sahai, and Waters (EUROCRYPT 2008), is an attractive branch of public-key encryption, which provides fine-grained and role-based access to encrypted data. As for many multi-user cryptosystems, an efficient revocation mechanism is necessary and imperative in the context of predicate encryption, in order to address scenarios when users misbehave or their private keys are compromised. The formal model of revocable predicate encryption was introduced by Nieto, Manulis and Sun (ACISP 2012), who suggest the strong, full-hiding security notion, demanding that the ciphertexts do not leak any information about the encrypted data, the attribute and the revocation information associated with it.

In this work, we introduce the first construction of lattice-based revocable predicate encryption. Our scheme satisfies the full-hiding security notion (in a selective manner) in the standard model, based on the hardness of the Learning With Errors (LWE) problem. In terms of asymptotic efficiency, the scheme is somewhat comparable to the pairing-based instantiation put forward by Nieto, Manulis and Sun. Furthermore, better efficiency could be easily achieved in the random oracle model.

1 Introduction

The notion of predicate encryption (PE), formalized by Katz, Sahai, and Waters [20], is an emerging paradigm of public-key encryption, which provides fine-grained and role-based access to encrypted data. In a PE scheme, the user's private key, issued by an authority, is associated with a predicate f, while a ciphertext is bound to an attribute a. The system then ensures that the user can decrypt the ciphertext if and only if $f(a) = 1$. PE can be viewed as a generalization of attribute-based encryption (ABE) [41,18]. Whereas the latter reveals the attribute bound to each ciphertext, the former preserves the privacy of not only the encrypted data but also the attribute. These powerful properties of PE yield numerous potential applications (see, e.g., [10,47,20]).

As for many multi-user cryptosystems, an efficient revocation mechanism is necessary and imperative in the PE setting. When some users misbehave or when their private keys are compromised, the users should be revoked from the system and should no longer be able to decrypt the ciphertext. In the ABE setting, Boldyreval et al. [8] suggested a revocation mechanism based on a time-based key update procedure. In their model, a ciphertext is not only bound to an attribute but also to a time period. The key authority, who possesses the up-to-date list of revoked users, have to publish an update key at each time period so that only non-revoked users can update their private keys to decrypt ciphertexts bound to the same time slot. This mechanism is known as indirect revocation, since the revocation information is not controlled by the message sender, but by the authority. A naïve solution for indirect revocation, first mentioned by Boneh and Franklin [9], consists of broadcasting user-specific update keys to all non-revoked users. However, this simple solution is inefficient, because the periodic workload of the authority is $O(N - r)$, where N is the number of users in the system and r is the number of revoked users at the given time period. Boldyreval et al. [8] adopted the classic subset-cover framework due to Naor et al. [29], which employs binary trees to handle user revocation, and reduced the size of update keys to $O\left(r \log \frac{N}{r}\right)$. Concrete pairing-based instantiations of revocable ABE following Boldyreval et al.'s approach were proposed in [5,40]. This approach, however, admits several limitations, since it requires the key authority to stay online regularly, and the non-revoked users to download updated system information periodically.

To eliminate the burden caused by the key update phase, Attrapadung and Imai [5] suggested the direct revocation mechanism for ABE, in which the revocation information can be controlled by the message sender. Each ciphertext is now bound to an attribute a as well as the current revocation list RL. Meanwhile, each private key associated with a predicate f is assigned a unique index I. The decryption procedure is successful if and only if $f(a) = 1$ and $I \notin \mathsf{RL}$. In this direct revocation model, the authority only can stay off-line after issuing private keys for users, and non-revoked users do not have to update their keys. Despite of the clear efficiency advantages for both the key authority and users, this approach requires that senders possess the current revocation list and perform encryptions based on it. The setting that the message sender should possess the revocation information might be inconvenient in certain scenarios, but it is well-suited in cases such information is naturally known to the sender. For instance, in Pay-TV systems [18], the TV program distributor should own the list of revoked users.

In [31,32], Nieto, Manulis and Sun (NMS) adapted the Attrapadung-Imai direct revocation mechanism into the context of PE, and formalized the notion of revocable predicated encryption (RPE). As discussed in [31,32], involved privacy challenges may rise when one plugs the revocation problem into the PE setting. In particular, Nieto, Manulis and Sun consider two security notions: attribute-hiding and full-hiding. The former means that the ciphertext only preserves privacy of attribute (and of the encrypted data) as in ordinary PE. The latter is a very strong notion which additionally guarantees that the revocation information is not leaked by the ciphertext. This requirement is suitable for applications where it is necessary for the sender to hide the list of revoked users. Nieto, Manulis and Sun pointed out that a generic construction of full-hiding RPE can be obtained by a combination of a PE scheme and an anonymous broadcasting scheme, but it is inefficient since the size of the ciphertexts is linearly dependent on the maximal number of users N. Then they proposed a more efficient paring-based instantiation of full-hiding RPE for inner-product predicates, which relies on the PE schemes by Okamoto and Takashima [34] and Lewko et al. [22], as well as the subset-cover framework [29].

In this work, inspired by the potentials of PE and the advantages of the direct revocation mechanism, we consider full-hiding RPE in the context of lattice-based cryptography, and aim to design the first such scheme from lattice assumptions. Lattice-based cryptography, pioneered by the seminal works by Regev [39] and Gentry et al. [15], has been one of the most exciting research areas in the last decade. Lattices provide several advantages over conventional number-theoretic cryptography, such as conjectured resistance against quantum adversaries and faster arithmetic operations. In the scope of lattice-based revocation schemes, there have been several proposals [11,12,30,48], but they only consider the setting of identity-based encryption (IBE). To the best of our knowledge, the problem of constructing lattice-based RPE schemes has not been addressed so far.

Our results and techniques. We introduce the first construction of RPE from lattices. Our scheme satisfies the full-hiding security notion [31,32] (in a selective manner) in the standard model, based on the hardness of the Learning With Errors (LWE) problem [39]. The scheme inherits the main advantage of the direct revocation mechanism: the authority does not have to be online after the key generation phase, and key updating is not needed. Let N be the maximum expected number of private keys in the system and let r be the number of revoked keys. Then, the efficiency of our scheme is comparable to that of the pairing-based RPE scheme from [31,32], in the following sense: the size of public parameters is $O(N)$; the size of the private key is $O(\log N)$, and the ciphertext has size $O\left(r \log \frac{N}{r}\right)$ which is ranged between $O(1)$ (when no key is revoked) and $O\left(\frac{N}{2}\right)$ (in the worst case when every second key is revoked).

At a high level, we adopt the approach suggested by Nieto, Manulis and Sun in their pairing-based instantiation [31,32], for which we introduce several modifications. Recall that, in [31,32], to obtain a full-hiding RPE, the authors apply the tree-based revocation technique from [29] to two layers of PE [34,22], in the following manner: the main PE layer deals with predicate vector $\overrightarrow{x}$ and attribute vector $\overrightarrow{y}$, while an additional layer is introduced to handle the index I of the private key (encoded as a "predicate") and the revocation list RL (encoded as an "attribute"). Thanks to the attribute-hiding property of the second PE layer, RL is kept hidden. It is worth noting that Nieto, Manulis and Sun managed to prove the full-hiding security by exploiting the dual system encryption techniques [50] underlying the PE blocks. Their security

proof fairly relies on the fact that the simulator is able to compute at least one private key for all predicates, including those for which the challenge attributes satisfy.

To adapt the approach from [31,32] into the lattice setting, we employ as the main PE layer the scheme for inner-product predicates put forward by Agrawal, Freeman and Vaikuntanathan [2] and subsequently improved by Xagawa [51]. However, we were not able to find a suitable lattice-based ingredient to be used as the second PE layer, so that it interacts smoothly and securely with the main layer (which might due to the fact that there has not been a lattice analogue of the dual system encryption techniques). Instead, we use a variant of Agrawal et al.'s anonymous IBE [1] to realize the second layer as follows. We first consider a binary tree with N leaves, where N is the maximum expected number of private keys. We then associate each node θ in the binary tree with an "identifier" $\mathbf{D}_\theta$. Then, for each $I \in [N]$, we equip the private key indexed by I with "decryption keys" corresponding to all identifiers in the tree path from I to the root. When generating a ciphertext with respect to revocation list RL, the sender aims to the identifiers $\mathbf{D}_{\theta'}$'s, for all θ' belonging to the cover set determined by RL. Thanks to the anonymity of the scheme, RL is kept hidden. Furthermore, the correctness of the tree-based revocation technique from [29] ensures that the ciphertext is decryptable using the private key indexed by I if and only if $I \notin \mathsf{RL}$.

To combine the AFV PE layer with the above anonymous IBE layer, we rely on a splitting technique that can be seen as a secret sharing mechanism and that was used in previous lattice-based revocation schemes [11,30,48]. To this end, for each $I \in [N]$, we split a public matrix $\mathbf{U}$ into two random parts: (i) $\mathbf{U}_I$ which is associated with the main PE layer; (ii) $\mathbf{U} - \mathbf{U}_I$ that is linked with the second layer.

The efficiency of our RPE can be improved in the random oracle model, where instead of storing all the matrices $\mathbf{D}_\theta$'s in the public parameters, we simply obtain them as outputs of a random oracle.

Other related works. The subset-cover framework, proposed by Naor et al. [29] in the context of broadcast encryption, is arguably the most well-known revocation technique for multi-user systems. It uses a binary tree, each leaf of which is designated to each user. Non-revoked users are partitioned into disjoint subsets, and are assigned keys according to the Complete Subtree (CS) method or the Subset Difference (SD) method. This framework was first considered in the IBE setting by Boldyreva et al. [8]. Subsequently, several identity-based instantiations from pairings [8,25] and from lattices [11,12,30,48] were proposed, providing various improvements. Seo and Emura [43] suggested a strong security notion for revocable IBE, that takes into account the threat of decryption key exposure attacks. There have been several constructions satisfying this strong notion, which operate in the subset-cover framework, e.g., [43,45,42,44,46,49]. The framework also found applications in the context of revocable group signatures [24,23], revocable ABE [8,5,40] and revocable PE [31,32,21].

Predicate encryption for inner-product predicates was introduced by Katz, Sahai, and Waters [20]. In such a scheme, attribute a and predicate f are expressed as vectors $\overrightarrow{x}$ and $\overrightarrow{y}$ respectively, and we say $f(a) = 1$ if and only if $\langle \overrightarrow{x}, \overrightarrow{y} \rangle = 0$ (hereafter, $\langle \overrightarrow{x}, \overrightarrow{y} \rangle$ denotes the inner product of vector $\overrightarrow{x}$ and vector $\overrightarrow{y}$). Katz, Sahai, and Waters also demonstrated the expressiveness of inner-product predicates: they can be used to encode several other predicates, such as equalities, hidden vector predicate, polynomial evaluation and CNF/DNF formulae. Following the work of [20], a number of pairing-based PE schemes [33,22,34,6,35,36] for inner products have been proposed. In the lattice-based world, Agrawal et al. [2] proposed the first such scheme, and Xagawa [51] suggested an improved variant.

Organization. The rest of this paper is organized as follows. In Section 2, we recall some background on lattice-based cryptography, revocable predicate encryption and the Complete Subtree method. Our main construction is described and analyzed in Section 3. Finally, we discuss possible extensions of our scheme and some open questions in Section 4.

2 Preliminaries

Notations. The acronym PPT stands for "probabilistic polynomial-time". We often write $x \hookleftarrow \chi$ to indicate that we sample x from probability distribution χ. If Ω is a finite set, the notation $x \xleftarrow{\$} \Omega$ means that x

is chosen uniformly at random from Ω. Meanwhile, if x is an output of PPT algorithm $\mathcal{A}$, then we write $x \leftarrow \mathcal{A}$.

We use bold upper-case letters (e.g., $\mathbf{A}, \mathbf{B}$) to denote matrices and use bold lower-case letters (e.g., $\mathbf{x}, \mathbf{y}$) to denote column vectors. In addition, we user over-arrows to denote predicate and attribute vectors as $\overrightarrow{x}, \overrightarrow{y}$. For two matrices $\mathbf{A} \in \mathbb{R}^{n\times m}$ and $\mathbf{B} \in \mathbb{R}^{n\times k}$, we denote by $[\mathbf{A} \mid \mathbf{B}] \in \mathbb{R}^{n\times(m+k)}$ the column-concatenation of $\mathbf{A}$ and $\mathbf{B}$. For a vector $\mathbf{x} \in \mathbb{Z}^n$, $||\mathbf{x}||$ denotes the Euclidean norm of $\mathbf{x}$. We use $\widetilde{\mathbf{A}}$ to denote the Gram-Schmidt orthogonalization of matrix $\mathbf{A}$, and $||\mathbf{A}||$ to denote the Euclidean norm of the longest column in $\mathbf{A}$. If n is a positive integer, $[n]$ denotes the set $\{1, .., n\}$. For $c \in \mathbb{R}$, let $\lfloor c \rceil = \lceil c - 1/2 \rceil$ denote the integer closest to c.

2.1 Background on Lattices

Integer lattices. An *m-dimensional lattice* Λ is a discrete subgroup of $\mathbb{R}^m$. A full-rank matrix $\mathbf{B} \in \mathbb{R}^{m\times m}$ is a *basis* of Λ if

$$\Lambda = \{\mathbf{y} \in \mathbb{R}^m : \exists \mathbf{s} \in \mathbb{Z}^m, \mathbf{y} = \mathbf{B} \cdot \mathbf{s}\}.$$

We are interested in integer lattices, i.e., when $\Lambda \subseteq \mathbb{Z}^m$. For any integer $q \geq 2$ and any $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, define the q-ary lattice:

$$\Lambda_q^{\perp}(\mathbf{A}) = \left\{\mathbf{r} \in \mathbb{Z}^m : \ \mathbf{A} \cdot \mathbf{r} = \mathbf{0} \bmod q\right\} \subseteq \mathbb{Z}^m.$$

For any $\mathbf{u}$ in the image of $\mathbf{A}$, define the coset $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \left\{\mathbf{r} \in \mathbb{Z}^m : \ \mathbf{A} \cdot \mathbf{r} = \mathbf{u} \bmod q\right\}$.

A fundamental tool in lattice-based cryptography is an algorithm that generates a matrix $\mathbf{A}$ statistically close to uniform over $\mathbb{Z}_q^{n\times m}$ together with a short basis $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$.

Lemma 1 ([3,4,27]). *Let $n \geq 1, q \geq 2$ and $m \geq 2n\log q$ be integers. There exists a PPT algorithm $\mathsf{TrapGen}(n, q, m)$ that outputs a pair $(\mathbf{A}, \mathbf{T_A})$ such that $\mathbf{A}$ is statistically close to uniform over $\mathbb{Z}_q^{n\times m}$ and $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ is a basis for $\Lambda_q^{\perp}(\mathbf{A})$, satisfying*

$$\|\widetilde{\mathbf{T_A}}\| \leq O(\sqrt{n\log q}) \text{ and } \|\mathbf{T_A}\| \leq O(n\log q)$$

with all but negligible probability in n.

Micciancio and Peikert [27] consider a structured matrix $\mathbf{G}$, called the *primitive matrix*, which admits a publicly known short basis.

Lemma 2 ([27]). *Let $n \geq 1, q \geq 2$ be integers and let $m \geq n\lceil\log q\rceil$. There exists a full-rank matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ such that the lattice $\Lambda_q^{\perp}(\mathbf{G})$ has a known basis $\mathbf{T_G} \in \mathbb{Z}^{m\times m}$ with $||\widetilde{\mathbf{T_G}}|| \leq \sqrt{5}$.*

Furthermore, there exists a deterministic polynomial-time algorithm $\mathbf{G}^{-1}$ which takes the input $\mathbf{U} \in \mathbb{Z}_q^{n\times m}$ and outputs $\mathbf{X} = \mathbf{G}^{-1}(\mathbf{U})$ such that $\mathbf{X} \in \{0,1\}^{m\times m}$ and $\mathbf{GX} = \mathbf{U}$.

Discrete Gaussians. Let $\Lambda \subseteq \mathbb{Z}^m$ be a lattice. For any vector $\mathbf{c} \in \mathbb{R}^m$ and any parameter $s > 0$, define $\rho_{s,\mathbf{c}}(\mathbf{r}) = \exp(-\pi\frac{\|\mathbf{r}-\mathbf{c}\|^2}{s^2})$ and $\rho_{s,\mathbf{c}}(\Lambda) = \sum_{\mathbf{r}\in\Lambda}\rho_{s,\mathbf{c}}(\mathbf{r})$ The discrete Gaussian distribution over Λ with center $\mathbf{c}$ and parameter s is

$$\forall \mathbf{r} \in \Lambda, \mathcal{D}_{\Lambda,s,\mathbf{c}}(\mathbf{r}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{r})}{\rho_{s,\mathbf{c}}(\Lambda)}.$$

If $\mathbf{c} = \mathbf{0}$, for simplicity, we often use the notations ρ_s and $\mathcal{D}_{\Lambda,s}$.
Gentry et al. [15] showed how to sample from discrete Gaussians over lattices that have sufficiently short bases.

Lemma 3 ([15]). *There exists an efficient PPT algorithm $\mathsf{SampleGaussian}(\mathbf{B}, s, \mathbf{c})$ that, given a basis $\mathbf{B}$ of an m-dimensional lattice Λ, a Gaussian parameter $s \geq ||\widetilde{\mathbf{B}}|| \cdot \omega(\sqrt{\log m})$, and a center $\mathbf{c} \in \mathbb{R}^m$, outputs a sample from a distribution that is statistically close to $\mathcal{D}_{\Lambda,s,\mathbf{c}}$.*

We also need the following lemma for proving the correctness and security of the construction in Section 3. The lemma is obtained based on known facts from [15, Lemma 5.2], [27] and [13, Lemma 5].

Lemma 4. *Let $n \geq 1, q \geq 2$, $m \geq 2n \log q$ and $k \geq 1$ be integers. Let $\mathbf{F}$ be a full-rank matrix in $\mathbb{Z}_q^{n \times m}$ and $\mathbf{T_F}$ be a basis of $\Lambda_q^{\perp}(\mathbf{F})$. Assume that $s \geq ||\widetilde{\mathbf{T_F}}|| \cdot \omega(\sqrt{\log n})$. Then, for $\mathbf{Z} \hookleftarrow (\mathcal{D}_{\mathbb{Z}^m, s})^k$, the distribution of $\mathbf{FZ} \bmod q$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{n \times k}$.*

In particular, Lemma 4 holds when $\mathbf{F}$ is a uniformly random matrix in $\mathbb{Z}_q^{n \times m}$ (see [15, Lemma 5.1] or when $\mathbf{F}$ is the primitive matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ specified in Lemma 2.

Sampling algorithms. It was shown in [1,27] how to efficiently sample short vectors from specific lattices. Looking ahead, we will employ algorithm SampleLeft to sample keys in the RPE scheme of Section 3, while algorithm SampleRight will be used to generate keys in the security proof.

SampleLeft $(\mathbf{A}, \mathbf{M}, \mathbf{T_A}, \mathbf{u}, s)$: On input a rank n matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a matrix $\mathbf{M} \in \mathbb{Z}_q^{n \times m_1}$, a trapdoor $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter $s \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(m + m_1)})$, it outputs a vector $\mathbf{z} \in \mathbb{Z}^{(m+m_1)}$, which is sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}), s}$. Here we define $\mathbf{F} = [\mathbf{A} \,|\, \mathbf{M}] \in \mathbb{Z}_q^{n \times (m+m_1)}$.

SampleRight $(\mathbf{A}, \mathbf{R}, t, \mathbf{G}, \mathbf{T_G}, \mathbf{u}, s)$: On input matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{R} \in \mathbb{Z}^{m \times m}$, a scalar $t \in \mathbb{Z}_q \backslash \{0\}$, the primitive matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ together with trapdoor $\mathbf{T_G}$ of $\Lambda_q^{\perp}(\mathbf{G})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter $s \geq \|\widetilde{\mathbf{T_B}}\| \cdot ||\mathbf{R}|| \cdot \omega(\sqrt{\log m})$, it outputs a vector $\mathbf{z} \in \mathbb{Z}^{2m}$, which is sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}), s}$. Here we define $\mathbf{F} = [\mathbf{A} \,|\, \mathbf{AR} + t\mathbf{G}] \in \mathbb{Z}_q^{n \times 2m}$.

The above sampling algorithms are easily extended to the case where instead of taking a vector $\mathbf{u} \in \mathbb{Z}_q^n$ as input, one takes a matrix $\mathbf{U} \in \mathbb{Z}_q^{n \times k}$, for some $k \geq 1$. In this case, the output is a matrix $\mathbf{Z} \in \mathbb{Z}^{2m \times k}$.

We will also need a variant of left over hash lemma from [1].

Lemma 5. *Suppose that $m > (n+1) \log q + \omega(\log n)$ and $q > 2$ is a prime. Choose $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n \times \kappa}$ and $\mathbf{R} \xleftarrow{\$} \{-1, 1\}^{m \times \kappa}$ where $\kappa = \kappa(n)$ is polynomial in n. Then for any vector $\mathbf{v} \in \mathbb{Z}_q^m$, the distribution of $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^{\top}\mathbf{v})$ is statistically close to the distribution of $(\mathbf{A}, \mathbf{B}, \mathbf{R}^{\top}\mathbf{v})$.*

Learning With Errors. We now recall the Learning With Errors (LWE) problem [39], as well as its hardness.

Definition 1 (LWE). *Let $n, m \geq 1, q \geq 2$, and let χ be a probability distribution on $\mathbb{Z}$. For $\mathbf{s} \in \mathbb{Z}_q^n$, let $\mathbf{A}_{\mathbf{s}, \chi}$ be the distribution obtained by sampling $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ and $e \hookleftarrow \chi$, and outputting the pair $\left(\mathbf{a}, \mathbf{a}^{\top}\mathbf{s} + e\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The (n, q, χ)-LWE problem asks to distinguish m samples chosen according to $\mathbf{A}_{\mathbf{s}, \chi}$ (for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$) and m samples chosen according to the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

If q is a prime power and $B \geq \sqrt{n} \cdot \omega(\log n)$, then there exists an efficient sampleable B-bounded distribution χ (i.e., χ outputs samples with norm at most B with overwhelming probability) such that (n, q, χ)-LWE is as least as hard as worst-case lattice problem SIVP with approximate factor $O(nq/B)$ (see [39,37,26,27]).

2.2 The Agrawal-Freeman-Vaikuntanathan Predicate Encryption Scheme

Next, we recall the LWE-based predicate encryption, proposed by Agrawal, Freeman and Vaikuntanathan (AFV) [2]. The scheme is for inner-product predicates, where an attribute is expressed as a vector $\overrightarrow{y} \in \mathbb{Z}_q^{\ell}$ (for some integers q and ℓ) and a predicate $f_{\overrightarrow{x}}$ is associated with a vector $\overrightarrow{x} \in \mathbb{Z}_q^{\ell}$. We say that $f_{\overrightarrow{x}}(\overrightarrow{y}) = 1$ if $\langle \overrightarrow{x}, \overrightarrow{y} \rangle = 0$, and $f_{\overrightarrow{x}}(\overrightarrow{y}) = 0$ otherwise. The set $\mathbb{A} = \mathbb{Z}_q^{\ell}$ is called the attribute space, while the set $\mathbb{P} = \{f_{\overrightarrow{x}} \mid \overrightarrow{x} \in \mathbb{Z}_q^{\ell}\}$ is called the predicate space.

In the AFV scheme, the key authority possesses a short basis $\mathbf{T_A}$ for a public lattice $\Lambda_q^{\perp}(\mathbf{A})$, outputted by the TrapGen algorithm. Each predicate $f_{\overrightarrow{x}} \in \mathbb{P}$ is associated with a super-lattice of $\Lambda_q^{\perp}(\mathbf{A})$, a short vector of which can be efficiently computed using the trapdoor $\mathbf{T_A}$. Such a short vector allows to decrypt a

Dual-Regev ciphertext [15] bound to an attribute vector $\overrightarrow{y} \in \mathbb{A}$ satisfying $f_{\overrightarrow{x}}(\overrightarrow{y}) = 1$. In order to improve efficiency, Xagawa [51] suggested an enhanced variant that employs the primitive matrix $\mathbf{G}$. In the below, we will describe the AFV scheme with Xagawa's improvement. The scheme works with appropriately chosen parameters n, q, m, s and LWE error distribution χ.

Setup: Generate $(\mathbf{A}, \mathbf{T}_{\mathbf{A}}) \leftarrow \mathsf{TrapGen}(n, q, m)$. Pick $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$ and for each $i \in [\ell]$, sample $\mathbf{A}_i \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$. Output

$$\mathsf{pp} = (\mathbf{A}, \{\mathbf{A}_i\}_{i \in [\ell]}, \mathbf{u}); \quad \mathsf{msk} = \mathbf{T}_{\mathbf{A}}.$$

KeyGen: For a predicate vector $\overrightarrow{x} = (x_1, \ldots, x_\ell) \in \mathbb{Z}_q^\ell$, compute $\mathbf{A}_{\overrightarrow{x}} = \sum_{i=1}^{\ell} \mathbf{A}_i \mathbf{G}^{-1}(x_i \cdot \mathbf{G}) \in \mathbb{Z}_q^{n \times m}$ and output the private key $\mathsf{sk}_{\overrightarrow{x}} = \mathbf{r} \in \mathbb{Z}^{2m}$ by running

$$\mathbf{r} \leftarrow \mathsf{SampleLeft}\left(\mathbf{A}, \mathbf{A}_{\overrightarrow{x}}, \mathbf{T}_{\mathbf{A}}, \mathbf{u}, s\right).$$

Enc: To encrypt a message $M \in \{0,1\}$ under an attribute vector $\overrightarrow{y} = (\overrightarrow{y}_1, \ldots, \overrightarrow{y}_\ell) \in \mathbb{Z}_q^\ell$, choose $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \hookleftarrow \chi^m$, $e \hookleftarrow \chi$, and $\mathbf{R}_i \xleftarrow{\$} \{-1, 1\}^{m \times m}$ for each $i \in [\ell]$, then output $\mathsf{ct} = (c', \mathbf{c}_0, \{\mathbf{c}_i\}_{i \in [\ell]})$, where:

$$\begin{aligned} c' &= \mathbf{u}^\top \mathbf{s} + e + M \cdot \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q, \\ \mathbf{c}_0 &= \mathbf{A}^\top \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m, \\ \forall i \in [\ell]: \quad \mathbf{c}_i &= (\mathbf{A}_i + y_i \cdot \mathbf{G})^\top \mathbf{s} + \mathbf{R}_i^\top \mathbf{e} \in \mathbb{Z}_q^m. \end{aligned}$$

Dec: Set $\mathbf{c}_{\overrightarrow{x}} = \sum_{i=1}^{\ell} \left(\mathbf{G}^{-1}(x_i \cdot \mathbf{G})\right)^\top \mathbf{c}_i \in \mathbb{Z}_q^m$. Then compute $z = c' - \mathbf{r}^\top [\mathbf{c}_0 \mid \mathbf{c}_{\overrightarrow{x}}] \in \mathbb{Z}_q$ and output $\lfloor \frac{2}{q} \cdot z \rceil \in \{0, 1\}$.

Agrawal, Freeman and Vaikuntanathan showed that, under the (n, q, χ)-LWE assumption, their PE scheme satisfies the weak attribute-hiding security notion defined by Katz, Sahai and Waters [20], in a selective attribute setting. Xagawa [51] proved that the same assertion holds for his improved scheme variant. In Section 3, the scheme will be used as a building block for our lattice-based instantiation of revocable predicate encryption.

2.3 Revocable Predicate Encryption

Now, we recall the definition of RPE from [5,31,32], and its full-hiding security notion suggested by Nieto, Manulis and Sun [31,32].

Definition 2. *A revocable predicate encryption scheme consists of four algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *and has an associated predicate space* $\mathbb{P}$*, an attribute space* $\mathbb{A}$*, an index space* $\mathcal{I}$ *and a message space* $\mathcal{M}$*.*

$\mathsf{Setup}\left(1^\lambda\right)$ *takes as input a security parameter* λ*. It outputs a state information* ST*, a set of public parameters* pp *and a master secret key* msk*. We assume* pp *to be an implicit input of all other algorithms.*

$\mathsf{KeyGen}\left(\mathsf{msk}, \mathsf{ST}, \overrightarrow{x}, I\right)$ *takes as input the master secret key* msk*, the state* ST*, a predicate vector* $\overrightarrow{x}$ *corresponding to a predicate* $f_{\overrightarrow{x}} \in \mathbb{P}$ *and an index* $I \in \mathcal{I}$*. It outputs an updated state* ST *and a private key* $\mathsf{sk}_{\overrightarrow{x},I}$*.*

$\mathsf{Enc}\left(\overrightarrow{y}, \mathsf{RL}, M\right)$ *takes as input an attribute vector* $\overrightarrow{y} \in \mathbb{A}$*, a revocation list* $\mathsf{RL} \subseteq \mathcal{I}$*, and a message* $M \in \mathcal{M}$*. It outputs a ciphertext* ct*.*

$\mathsf{Dec}\left(\mathsf{ct}, \mathsf{sk}_{\overrightarrow{x},I}\right)$ *takes as input a ciphertext* ct *and a private key* $\mathsf{sk}_{\overrightarrow{x},I}$*. It outputs a message* M *or the distinguished symbol* $\perp$*.*

Correctness. The correctness requirement demands that, for all pp and msk generated by $\mathsf{Setup}\left(1^\lambda\right)$, all $f_{\overrightarrow{x}} \in \mathbb{P}$, $\overrightarrow{y} \in \mathbb{A}$, $I \in \mathcal{I}$, all possible valid state information ST, all $\mathsf{sk}_{\overrightarrow{x},I} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{ST}, \overrightarrow{x}, I)$ and $\mathsf{ct} \leftarrow \mathsf{Enc}(\overrightarrow{y}, \mathsf{RL}, M)$, if $I \notin \mathsf{RL}$ then:

1. If $f_{\overrightarrow{x}}(\overrightarrow{y}) = 1$ then $\mathsf{Dec}\left(\mathsf{ct}, \mathsf{sk}_{\overrightarrow{x},I}\right) = M$.
2. If $f_{\overrightarrow{x}}(\overrightarrow{y}) = 0$ then $\mathsf{Dec}\left(\mathsf{ct}, \mathsf{sk}_{\overrightarrow{x},I}\right) = \perp$ with all but negligible probability.

Full-Hiding Security. In [31,32], Nieto, Manulis and Sun introduced the notion of full-hiding security against chosen plaintext attacks for RPE, which demands that ciphertexts do not leak any information about the plaintexts, the attributes, nor the revoked indexes. This notion can be defined in the strong, adaptive manner, or in the relaxed, selective sense where the adversary is required to announce the challenge attribute vectors $\overrightarrow{y}^{(0)}$, $\overrightarrow{y}^{(1)}$ and revocation lists $\mathsf{RL}^{(0)}, \mathsf{RL}^{(1)}$ before seeing public parameters. In this work, we consider the latter.

Definition 3. *An RPE scheme is selectively full hiding against chosen plaintext attacks if any PPT adversary $\mathcal{A}$ has negligible advantage in the following game:*

1. *$\mathcal{A}$ first announces the challenge attribute vectors $\overrightarrow{y}^{(0)}$, $\overrightarrow{y}^{(1)}$, revocation lists $\mathsf{RL}^{(0)}, \mathsf{RL}^{(1)}$.*
2. *$\mathsf{Setup}\left(1^\lambda\right)$ is run to generate a state information ST, a set of public parameters pp and a master secret key msk. Then $\mathcal{A}$ is given pp.*
3. *$\mathcal{A}$ may make queries for private keys. For a query of a predicate vector and an index in the form $(\overrightarrow{x}, I)$, $\mathcal{A}$ is given $\mathsf{sk}_{\overrightarrow{x},I} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{ST}, \overrightarrow{x}, I)$, subject to one of the following restrictions:*
   - *$f_{\overrightarrow{x}}(\overrightarrow{y}^{(0)}) = f_{\overrightarrow{x}}(\overrightarrow{y}^{(1)}) = 0$;*
   - *$f_{\overrightarrow{x}}(\overrightarrow{y}^{(0)}) = f_{\overrightarrow{x}}(\overrightarrow{y}^{(1)}) = 1$ and $I \in \mathsf{RL}^{(0)} \cap \mathsf{RL}^{(1)}$;*
   - *$f_{\overrightarrow{x}}(\overrightarrow{y}^{(0)}) = 1 \wedge f_{\overrightarrow{x}}(\overrightarrow{y}^{(1)}) = 0$ and $I \in \mathsf{RL}^{(0)}$;*
   - *$f_{\overrightarrow{x}}(\overrightarrow{y}^{(0)}) = 0 \wedge f_{\overrightarrow{x}}(\overrightarrow{y}^{(1)}) = 1$ and $I \in \mathsf{RL}^{(1)}$.*
4. *$\mathcal{A}$ outputs two challenge plaintexts $M^{(0)}, M^{(1)}$. A uniformly random bit b is chosen, and $\mathcal{A}$ is given the ciphertext $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\overrightarrow{y}^{(b)}, \mathsf{RL}^{(b)}, M^{(b)})$.*
5. *The adversary may continue to make additional queries for private keys, subject to the same restrictions as before.*
6. *$\mathcal{A}$ outputs a bit b' and succeeds if $b' = b$. The advantage of $\mathcal{A}$ in the game is defined as:*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{sFH}}(\lambda) = \left| \Pr\left[b' = b\right] - \frac{1}{2} \right|.$$

Remark 1. In the above game, the restrictions for private-key queries are to prevent the adversary to trivially win the game by decrypting the challenge ciphertext ct^*. For the same reason, it is necessary to assume that the two ciphertexts $\mathsf{Enc}(\overrightarrow{y}^{(0)}, \mathsf{RL}^{(0)}, M^{(0)})$ and $\mathsf{Enc}(\overrightarrow{y}^{(1)}, \mathsf{RL}^{(1)}, M^{(1)})$ have the same size.

2.4 The Complete Subtree Method

The complete subtree (CS) method, introduced by Naor, Naor and Lotspiech [29], has been widely used in revocation systems. It makes use of a node selection algorithm (called $\mathsf{KUNodes}$). In the algorithm, we build a complete binary tree BT and use the following notation: If θ is a non-leaf node, then θ_ℓ and θ_r denote the left and right child of θ, respectively. Whenever ν is a leaf node, the set $\mathsf{Path}(\nu)$ stands for the collection of nodes on the path from ν to the root (including ν and the root). The $\mathsf{KUNodes}$ algorithm takes as input the binary tree BT and a revocation list RL, and outputs a set of nodes Y which is the smallest subset of nodes that contains an ancestor of all the leaf nodes corresponding to non-revoked indexes. It is known [29] that the set Y generated by $\mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL})$ has a size at most $r \log \frac{N}{r}$, where r is the number of indexes in RL. The detailed description of algorithm $\mathsf{KUNodes}$ is given below.

$\mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL})$

$X, Y \leftarrow \emptyset$

$\forall \nu \in \mathsf{RL}:$ add $\mathsf{Path}(\nu)$ to X

$\forall \theta \in X:$

if $\theta_\ell \notin X,$ then add θ_ℓ to Y

if $\theta_r \notin X,$ then add θ_r to Y

If $Y = \emptyset,$ then add root to Y

Return Y

In the scheme of Section 3, we will employ the CS method to realize user revocation.

3 Our Lattice-Based RPE Scheme

This section presents our construction of lattice-based RPE scheme for inner-product predicates. As we briefly discussed in Section 1, the scheme employs two encryption layers: the AFV PE scheme [2,51] and a variant of Agrawal et al.'s anonymous IBE scheme [1]. Revocation is realized using the CS method and a splitting technique that can be seen as a secret sharing mechanism and that was used in previous lattice-based revocation schemes [11,30,48].

Before describing our scheme in detail, let us discuss a small issue in existing PE schemes [2,51,13,17] from lattices. Recall that the correctness of PE requires in particular that if $f_{\overrightarrow{x}}(\overrightarrow{y}) = 0$ then the decryption algorithm with private key $\mathsf{sk}_{\overrightarrow{x}}$ must fail with all but negligible probability when applying to a ciphertext associated with $\overrightarrow{y}$. However, in the LWE-based public-key encryption schemes used in the above constructions, the decryption algorithm does not fail: it outputs a random element in the plaintext space $\mathcal{M}$. To overcome this issue and enforce correctness, the following idea was suggested and implemented in [2,51,13,17], assuming that the scheme can be modified to work with plaintext space $\mathcal{M}'$, such that $|\mathcal{M}|/|\mathcal{M}'| = \mathsf{negl}(\lambda)$, where λ is the security parameter. Then, to encrypt an element of $\mathcal{M}$, one encodes it to an element of $\mathcal{M}'$ and proceeds with the encoding. Since the probability that a random element in $\mathcal{M}'$ is a proper encoding is negligible, the correctness of the scheme is ensured with all but negligible probability.

Our scheme operates with plaintext space $\mathcal{M} = \{0,1\}$. Following the idea discussed above, let us define the encoding function $\mathsf{encode} : \mathcal{M} \to \{0,1\}^k$ for $k = \omega(\log \lambda)$, such that for each $b \in \mathcal{M}$, we have $\mathsf{encode}(b) = (b, 0, \ldots, 0) \in \{0,1\}^k$ - the binary vector that has b as the first coordinate and 0 elsewhere. This encoding technique satisfies the desirable property, since we have $2/2^k = 2^{-\omega(\log \lambda)} = \mathsf{negl}(\lambda)$.

3.1 Description of the Scheme

Our scheme works with security parameter λ and global parameters $N, \ell, n, q, m, k, \mathbf{G}, s, B, \chi$ specified below.

- $N = \mathsf{poly}(\lambda)$: the maximum expected number of users;
- $\ell = \mathsf{poly}(\lambda)$: the length of predicate and attribute vectors;
- Lattice parameter $n = O(\lambda)$, prime modulus $q = \widetilde{O}(\ell^2 n^4)$, dimensions $m = \lceil 2n \log q \rceil$ and $k = \omega(\log \lambda)$;
- The primitive matrix $\mathbf{G}$ with public trapdoor $\mathbf{T}_{\mathbf{G}}$ (see Lemma 2);
- Gaussian parameter $s = \widetilde{O}(\ell\sqrt{m})$; Norm bound $B = \widetilde{O}(\sqrt{m})$ and B-bounded distribution χ.

The attribute space is set as $\mathbb{A} = \mathbb{Z}_q^\ell$. Each $\overrightarrow{x} \in \mathbb{A}$ is associated with predicate $f_{\overrightarrow{x}} : \mathbb{A} \to \{0,1\}$, where for all $\overrightarrow{y} \in \mathbb{A}$, we have: $f_{\overrightarrow{x}}(\overrightarrow{y}) = 1$ if and only if $\langle \overrightarrow{x}, \overrightarrow{y} \rangle = 0$. The predicate space is then defined as $\mathbb{P} = \{f_{\overrightarrow{x}} \mid \overrightarrow{x} \in \mathbb{A}\}$. The scheme works with index space $\mathcal{I} = [N]$.

We now provide the detailed description of the scheme.

$\mathsf{Setup}(1^\lambda)$: On input security parameter λ, this algorithm performs the following steps:

1. Run the algorithm $\mathsf{TrapGen}(n, q, m)$ to generate a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a basis $\mathbf{T_A}$ for $\Lambda_q^\perp(\mathbf{A})$ such that $\|\widetilde{\mathbf{T_A}}\| \leq O(\sqrt{n \log q})$.
2. Pick $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{n \times k}$.
3. Sample $\mathbf{A}_i \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, for each $i \in [\ell]$.
4. Build a binary tree BT with N leaf nodes. For each node $\theta \in \mathsf{BT}$, choose $\mathbf{D}_\theta \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, which will be viewed as the "identifier" of the node.
5. Initialize the state $\mathsf{ST} = \emptyset$, which records the assigned indexes so far.
6. Output ST, $\mathsf{pp} = (\mathbf{A}, \{\mathbf{A}_i\}_{i \in [\ell]}, \mathbf{U}, \mathsf{BT})$ and $\mathsf{msk} = \mathbf{T_A}$.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{ST}, \overrightarrow{x}, I)$: On input the master key msk, state ST, a predicate vector $\overrightarrow{x} = (x_1, \ldots, x_\ell) \in \mathbb{Z}_q^\ell$ and an index $I \in [N]$, this algorithm performs the following steps:

1. If $I \in \mathsf{ST}$, then return $\perp$. Else, update the state $\mathsf{ST} \leftarrow \mathsf{ST} \cup \{I\}$.
2. Pick $\mathbf{U}_I \xleftarrow{\$} \mathbb{Z}_q^{n \times k}$.
3. Set $\mathbf{A}_{\overrightarrow{x}} = \sum_{i=1}^{\ell} \mathbf{A}_i \mathbf{G}^{-1}(x_i \cdot \mathbf{G})$ and sample $\mathbf{Z} \leftarrow \mathsf{SampleLeft}\left(\mathbf{A}, \mathbf{A}_{\overrightarrow{x}}, \mathbf{T_A}, \mathbf{U}_I, s\right)$. We note that $\mathbf{Z}$ is a matrix in $\mathbb{Z}^{2m \times k}$ satisfying $[\mathbf{A} \mid \mathbf{A}_{\overrightarrow{x}}] \cdot \mathbf{Z} = \mathbf{U}_I$.
4. For each $\theta \in \mathsf{Path}(I)$, sample $\mathbf{Z}_\theta \leftarrow \mathsf{SampleLeft}\left(\mathbf{A}, \mathbf{D}_\theta, \mathbf{T_A}, \mathbf{U} - \mathbf{U}_I, s\right)$. We remark that each $\mathbf{Z}_\theta$ is a matrix in $\mathbb{Z}^{2m \times k}$ satisfying $[\mathbf{A} \mid \mathbf{D}_\theta] \cdot \mathbf{Z}_\theta = \mathbf{U} - \mathbf{U}_I$.
5. Output the updated state ST and $\mathsf{sk}_{\overrightarrow{x},I} = \big(I,\ \mathbf{Z},\ \{\mathbf{Z}_\theta\}_{\theta \in \mathsf{Path}(I)}\big)$.

$\mathsf{Enc}(\overrightarrow{y}, \mathsf{RL}, M)$: On input an attribute vector $\overrightarrow{y} = (y_1, \ldots, y_\ell) \in \mathbb{Z}_q^\ell$, a revocation list $\mathsf{RL} \subseteq [N]$ and a message $M \in \{0, 1\}$, this algorithm performs the following steps:

1. Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}' \hookleftarrow \chi^k$ and $\mathbf{e} \hookleftarrow \chi^m$.
2. Pick $\mathbf{R}_i, \mathbf{S}_\theta \xleftarrow{\$} \{-1, 1\}^{m \times m}$, for each $i \in [\ell]$ and each $\theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL})$.
3. Output $\mathsf{ct} = \big(\mathbf{c}',\ \mathbf{c}_0,\ \{\mathbf{c}_i\}_{i \in [\ell]},\ \{\widehat{\mathbf{c}}_\theta\}_{\theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL})}\big)$, where:

$$\begin{aligned}
\mathbf{c}' &= \mathbf{U}^\top \mathbf{s} + \mathbf{e}' + \lfloor \frac{q}{2} \rfloor \cdot \mathsf{encode}(M) \in \mathbb{Z}_q^k, \\
\mathbf{c}_0 &= \mathbf{A}^\top \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m, \\
\forall i \in [\ell]: \quad \mathbf{c}_i &= (\mathbf{A}_i + y_i \cdot \mathbf{G})^\top \mathbf{s} + \mathbf{R}_i^\top \mathbf{e} \in \mathbb{Z}_q^m, \\
\forall \theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}): \quad \widehat{\mathbf{c}}_\theta &= \mathbf{D}_\theta^\top \mathbf{s} + \mathbf{S}_\theta^\top \mathbf{e} \in \mathbb{Z}_q^m.
\end{aligned}$$

$\mathsf{Dec}\left(\mathsf{ct}, \mathsf{sk}_{\overrightarrow{x},I}\right)$: On input a ciphertext $\mathsf{ct} = \big(\mathbf{c}',\ \mathbf{c}_0,\ \{\mathbf{c}_i\}_{i \in [\ell]},\ \{\widehat{\mathbf{c}}_{\theta'}\}_{\theta'}\big)$, where $\{\widehat{\mathbf{c}}_{\theta'}\}_{\theta'}$ denotes a collection of vectors in $\mathbb{Z}_q^m$, and a private key $\mathsf{sk}_{\overrightarrow{x},I} = \big(I,\ \mathbf{Z},\ \{\mathbf{Z}_\theta\}_{\theta \in \mathsf{Path}(I)}\big)$, this algorithm proceeds as follows:

1. Compute $\mathbf{c}_{\overrightarrow{x}} = \sum_{i=1}^{\ell} \left(\mathbf{G}^{-1}(x_i \cdot \mathbf{G})\right)^\top \mathbf{c}_i \in \mathbb{Z}_q^m$.
2. For all pairs (θ, θ'), compute $\mathbf{d}_{\theta,\theta'} = \mathbf{c}' - \mathbf{Z}^\top [\mathbf{c}_0 \mid \mathbf{c}_{\overrightarrow{x}}] - \mathbf{Z}_\theta^\top [\mathbf{c}_0 \mid \widehat{\mathbf{c}}_{\theta'}] \in \mathbb{Z}_q^k$.
3. If there exists a pair (θ, θ') such that $\lfloor \frac{2}{q} \cdot \mathbf{d}_{\theta,\theta'} \rceil = \mathsf{encode}(M')$, for some $M' \in \{0, 1\}$, then output M'. Otherwise, output $\perp$.

3.2 Correctness, Efficiency and Potential Implementation

Correctness. We will demonstrate that the scheme satisfies the correctness requirement with all but negligible probability. We proceed as in [2,51,13].

Suppose that $\mathsf{ct} = \left(\mathbf{c}',\ \mathbf{c}_0,\ \{\mathbf{c}_i\}_{i\in[\ell]},\ \{\widehat{\mathbf{c}}_\theta\}_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{RL})}\right)$ is an honestly computed ciphertext of message $M \in \{0,1\}$, with respect to some $\overrightarrow{y} \in \mathcal{A}$ and some $\mathsf{RL} \subseteq [N]$. Let $\mathsf{sk}_{\overrightarrow{x},I} = \left(I,\ \mathbf{Z},\ \{\mathbf{Z}_\theta\}_{\theta\in\mathsf{Path}(I)}\right)$ be a correctly generated private key, where $I \notin \mathsf{RL}$. We first observe that the following holds:

$$\mathbf{c}_{\overrightarrow{x}} = \sum_{i=1}^{\ell} \left(\mathbf{G}^{-1}(x_i \cdot \mathbf{G})\right)^\top \mathbf{c}_i = \left(\mathbf{A}_{\overrightarrow{x}} + \langle \overrightarrow{x}, \overrightarrow{y}\rangle \cdot \mathbf{G}\right)^\top \mathbf{s} + \sum_{i=1}^{\ell} \left(\mathbf{R}_i \mathbf{G}^{-1}(x_i \cdot \mathbf{G})\right)^\top \mathbf{e}. \tag{1}$$

By construction, since $I \notin \mathsf{RL}$, there exists a pair (θ, θ') corresponding to the same node in BT satisfying

$$[\mathbf{A} \mid \mathbf{A}_{\overrightarrow{x}}] \cdot \mathbf{Z} + [\mathbf{A} \mid \mathbf{D}_{\theta'}] \cdot \mathbf{Z}_\theta = \mathbf{U}.$$

We now consider two cases:

1. Suppose that $\langle \overrightarrow{x}, \overrightarrow{y}\rangle = 0$. In this case, we have: $\mathbf{c}_{\overrightarrow{x}} = (\mathbf{A}_{\overrightarrow{x}})^\top \mathbf{s} + \sum_{i=1}^{\ell} \left(\mathbf{R}_i \mathbf{G}^{-1}(x_i \cdot \mathbf{G})\right)^\top \mathbf{e}$. Then for the pair (θ, θ') specified above, the following holds:

$$\begin{aligned}
\mathbf{d}_{\theta,\theta'} &= \mathbf{c}' - \mathbf{Z}^\top [\mathbf{c}_0 \mid \mathbf{c}_{\overrightarrow{x}}] - \mathbf{Z}_\theta^\top [\mathbf{c}_0 \mid \widehat{\mathbf{c}}_{\theta'}] \\
&= \mathbf{U}^\top \mathbf{s} + \mathbf{e}' + \lfloor \frac{q}{2} \rfloor \cdot \mathsf{encode}(M) - \mathbf{Z}^\top \left([\mathbf{A} \mid \mathbf{A}_{\overrightarrow{x}}]^\top \mathbf{s} + \begin{bmatrix} \mathbf{e} \\ \sum_{i=1}^{\ell} \left(\mathbf{R}_i \mathbf{G}^{-1}(x_i \cdot \mathbf{G})\right)^\top \mathbf{e} \end{bmatrix} \right) \\
&\qquad - \mathbf{Z}_\theta^\top \left([\mathbf{A} \mid \mathbf{D}_{\theta'}]^\top \mathbf{s} + \begin{bmatrix} \mathbf{e} \\ \mathbf{S}_{\theta'}^\top \mathbf{e} \end{bmatrix} \right) \\
&= \lfloor \frac{q}{2} \rfloor \cdot \mathsf{encode}(M) + \underbrace{\mathbf{e}' - \mathbf{Z}^\top \begin{bmatrix} \mathbf{e} \\ \sum_{i=1}^{\ell} \left(\mathbf{R}_i \mathbf{G}^{-1}(x_i \cdot \mathbf{G})\right)^\top \mathbf{e} \end{bmatrix} - \mathbf{Z}_\theta^\top \begin{bmatrix} \mathbf{e} \\ \mathbf{S}_{\theta'}^\top \mathbf{e} \end{bmatrix}}_{\mathsf{error}}.
\end{aligned}$$

   As in [1,2,51,13], the above error term can be showed to be bounded by $s\ell m^2 B \cdot \omega(\log n) = \widetilde{O}(\ell^2 n^3)$, with all but negligible probability. In order for the decryption algorithm to recover $\mathsf{encode}(M)$, and subsequently the plaintext M, it is required that the error term is bounded by $q/5$, i.e., $||\mathsf{error}||_\infty < q/5$. This is guaranteed by our setting of modulus q, i.e., $q = \widetilde{O}\left(\ell^2 n^4\right)$.
2. Suppose that $\langle \overrightarrow{x}, \overrightarrow{y}\rangle \neq 0$. In this case, we have:

$$\mathbf{c}_{\overrightarrow{x}} = \left(\mathbf{A}_{\overrightarrow{x}} + \underbrace{\langle \overrightarrow{x}, \overrightarrow{y}\rangle}_{\neq 0} \cdot \mathbf{G}\right)^\top \mathbf{s} + \sum_{i=1}^{\ell} \left(\mathbf{R}_i \mathbf{G}^{-1}(x_i \cdot \mathbf{G})\right)^\top \mathbf{e}. \tag{2}$$

   Then for each pair (θ, θ'), the following holds:

$$\begin{aligned}
\mathbf{d}_{\theta,\theta'} = \mathbf{U}^\top \mathbf{s} + \mathbf{e}' + \lfloor \frac{q}{2} \rfloor \cdot \mathsf{encode}(M) &- \mathbf{Z}^\top \left([\mathbf{A} \mid \mathbf{A}_{\overrightarrow{x}} + \langle \overrightarrow{x}, \overrightarrow{y}\rangle \cdot \mathbf{G}]^\top \mathbf{s} + \begin{bmatrix} \mathbf{e} \\ \sum_{i=1}^{\ell} \left(\mathbf{R}_i \mathbf{G}^{-1}(x_i \cdot \mathbf{G})\right)^\top \mathbf{e} \end{bmatrix} \right) \\
&- \mathbf{Z}_\theta^\top \left([\mathbf{A} \mid \mathbf{D}_{\theta'}]^\top \mathbf{s} + \begin{bmatrix} \mathbf{e} \\ \mathbf{S}_{\theta'}^\top \mathbf{e} \end{bmatrix} \right) \\
= \lfloor \frac{q}{2} \rfloor \cdot \mathsf{encode}(M) + \left(\mathbf{U} - [\mathbf{A} \mid \mathbf{A}_{\overrightarrow{x}}] \cdot \mathbf{Z} - [\mathbf{A} \mid \mathbf{D}_{\theta'}] \cdot \mathbf{Z}_\theta\right)^\top \mathbf{s} &- \mathbf{Z}^\top [\mathbf{0} \mid \langle \overrightarrow{x}, \overrightarrow{y}\rangle \cdot \mathbf{G}]^\top \mathbf{s} + \mathsf{error}
\end{aligned}$$

Observe that the term $\mathbf{Z}^\top[\mathbf{0} \mid \langle \overrightarrow{x}, \overrightarrow{y} \rangle \cdot \mathbf{G}]^\top \mathbf{s}$ can be written as $\langle \overrightarrow{x}, \overrightarrow{y} \rangle \cdot (\mathbf{G}\mathbf{Z}_2)^\top \mathbf{s} \in \mathbb{Z}_q^k$, where $\mathbf{Z}_2 \in \mathbb{Z}^{m \times k}$ is the bottom part of matrix $\mathbf{Z}$. By Lemma 4, we have that the distribution of $\mathbf{G}\mathbf{Z}_2 \in \mathbb{Z}_q^{n \times k}$ is statistically close to uniform. This implies that, vector $\mathbf{d}_{\theta,\theta'} \in \mathbb{Z}_q^k$, for each pair (θ, θ'), is indistinguishable from uniform. As a result, the probability that the last $k-1$ coordinates of vector $\lfloor \frac{2}{q} \cdot \mathbf{d}_{\theta,\theta'} \rceil$ are all 0 is at most $2^{-(k-1)} = 2^{-\omega(\log \lambda)}$, which is negligible in λ. In other words, except for negligible probability, the decryption algorithm outputs $\perp$ since it does not obtain a proper encoding $\mathsf{encode}(M) \in \{0,1\}^k$, for $M \in \{0,1\}$.

Efficiency. The efficiency aspect of our RPE scheme is as follows:

- The bit-size of public parameters pp is $((\ell + 2N)nm + nk) \log q = \big(\widetilde{O}(\ell) + O(N)\big) \cdot \widetilde{O}\left(\lambda^2\right)$.
- The private key $\mathsf{sk}_{\overrightarrow{x},I}$ has bit-size $O(\log N) \cdot \widetilde{O}\left(\lambda\right)$.
- The bit-size of ciphertext ct is $\big(\widetilde{O}(\ell) + O(r \log \frac{N}{r})\big) \cdot \widetilde{O}\left(\lambda\right)$.

The efficiency of our scheme is comparable to that of the pairing-based RPE scheme from [31,32], in the following sense: the size of public parameters is $O(N)$; the size of the private key is $O(\log N)$, and the ciphertext has size $O\left(r \log \frac{N}{r}\right)$ which is ranged between $O(1)$ (when no key is revoked) and $O\left(\frac{N}{2}\right)$ (in the worst case when every second key is revoked).

In Section 4, we will discuss a variant of our scheme in the random oracle model, which has shorter public parameters.

Potential Implementation. While the focus of this work is to provide the first provably secure construction of RPE from lattice assumptions, it would be desirable to back it up with practical implementations and to compare the implementation details with those of pairing-based counterparts. However, this would be a highly challenging task, due to two main reasons:

1. We are not aware of any concrete implementation of the two building blocks of our scheme, i.e., the AFV PE scheme [2,51] and Agrawal et al.'s IBE scheme [1].
2. In [31,32], Nieto, Manulis and Sun did not provide implementation details of their pairing-based RPE scheme.

Given these circumstances, we leave the implementation aspect of our scheme as a future investigation. Nevertheless, in the following, we will discuss the potential of such implementation, by analyzing the main cryptographic operations needed for implementing the scheme. Apart from simple operations such as samplings of uniformly random matrices and vectors whose entries are in $\mathbb{Z}_q$ or $\{-1, 1\}$, as well as multiplication and addition operations over $\mathbb{Z}_q$, the algorithms of the scheme requires the following time-consuming tasks:

- $\diamond$ Generation of a lattice trapdoor;
- $\diamond$ Samplings of discrete Gaussian vectors over lattices;
- $\diamond$ Samplings of LWE noise vectors.

We note that it is feasible to implement the listed above cryptographic tasks using the algorithms provided in [15,27], which were recently improved in [28,14]. Some implementation results of those cryptographic tasks were recently reported in [19], which may serve as a stepping stone of the potential implementation of our scheme.

3.3 Security

In the following theorem, we prove that our scheme in Section 3 is selectively full hiding in the standard model, under the LWE assumption.

Theorem 1. *Our RPE scheme satisfies the selective full-hiding security defined in Definition 3, assuming hardness of the* (n, q, χ)-LWE *problem.*

Proof. We proceed via a series of games, similar to those in [2,13,16]. First, we define the auxiliary algorithms for generating simulated public parameters, private keys and ciphertexts, and then, we describe the games.

Auxiliary algorithms. We consider the following auxiliary algorithms.

$\mathsf{Sim.Setup}\big(1^\lambda, \mathbf{A}, \mathbf{U}, \overrightarrow{y}^*, \mathsf{RL}^*\big)$: On input a security parameter λ, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a matrix $\mathbf{U} \in \mathbb{Z}_q^{n\times k}$, the challenge attribute vector $\overrightarrow{y}^* = (y_1^*, \ldots, y_\ell^*) \in \mathbb{Z}_q^\ell$ and revocation list $\mathsf{RL}^* \subseteq [N]$, this algorithm performs the following steps:

1. For each $i \in [\ell]$, choose $\mathbf{R}_i \xleftarrow{\$} \{-1,1\}^{m\times m}$ and set $\mathbf{A}_i = \mathbf{A}\mathbf{R}_i - y_i^* \cdot \mathbf{G}$.
2. Build a binary tree BT and choose $\mathbf{S}_\theta \xleftarrow{\$} \{-1,1\}^{m\times m}$ for each $\theta \in \mathsf{BT}$. Set the identifier:
$$\mathbf{D}_\theta = \begin{cases} \mathbf{A}\mathbf{S}_\theta, & \text{if } \theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}^*), \\ \mathbf{A}\mathbf{S}_\theta + \mathbf{G}, & \text{otherwise.} \end{cases}$$
3. Initialize the state ST.
4. Output $\mathsf{ST}, \mathsf{pp} = \big(\mathbf{A}, \{\mathbf{A}_i\}_{i\in[\ell]}, \mathbf{U}, \mathsf{BT}\big)$ and $\mathsf{msk}^* = \big(\{\mathbf{R}_i\}_{i\in[\ell]}, \{\mathbf{S}_\theta\}_{\theta\in\mathsf{BT}}\big)$.

$\mathsf{Sim.KeyGen}\big(\mathsf{msk}^*, \mathsf{ST}, \overrightarrow{x}, I, \overrightarrow{y}^*, \mathsf{RL}^*\big)$: This algorithm takes as input msk^*, state ST, a predicate vector $\overrightarrow{x} \in \mathbb{Z}_q^\ell$, an index $I \in [N]$, the challenge attribute vector $\overrightarrow{y}^* \in \mathbb{Z}_q^\ell$ and revocation list $\mathsf{RL}^* \subseteq [N]$, such that the following condition holds: If $\langle \overrightarrow{x}, \overrightarrow{y}^* \rangle = 0$ then $I \in \mathsf{RL}^*$.
The algorithm returns $\perp$ if $I \in \mathsf{ST}$. Otherwise, it outputs the updated state $\mathsf{ST} \leftarrow \mathsf{ST} \cup \{I\}$ and private key $\mathsf{sk}_{\overrightarrow{x},I} = \big(I, \mathbf{Z}, \{\mathbf{Z}_\theta\}_{\theta\in\mathsf{Path}(I)}\big)$ computed based on the value $\langle \overrightarrow{x}, \overrightarrow{y}^* \rangle$ as follows.

1. **Case 1:** $\langle \overrightarrow{x}, \overrightarrow{y}^* \rangle \neq 0$.
   (a) If $I \notin \mathsf{RL}^*$, then there is exactly one node θ^* in the intersection $\mathsf{Path}(I) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}^*)$. Using Lemma 3, sample $\mathbf{Z}_{\theta^*} \hookleftarrow \big(\mathcal{D}_{\mathbb{Z}^{2m},s}\big)^k$ and set $\mathbf{U}_I = \mathbf{U} - [\mathbf{A} \mid \mathbf{D}_{\theta^*}] \cdot \mathbf{Z}_{\theta^*}$. For each node $\theta \in \mathsf{Path}(I)\backslash\{\theta^*\}$, sample $\mathbf{Z}_\theta \leftarrow \mathsf{SampleRight}(\mathbf{A}, \mathbf{S}_\theta, 1, \mathbf{G}, \mathbf{T}_\mathbf{G}, \mathbf{U} - \mathbf{U}_I, s)$. (See Section 2.1 for the description of algorithm $\mathsf{SampleRight}$.)
   (b) If $I \in \mathsf{RL}^*$, choose $\mathbf{U}_I \xleftarrow{\$} \mathbb{Z}_q^{n\times k}$. Then for each $\theta \in \mathsf{Path}(I)$, sample
   $$\mathbf{Z}_\theta \leftarrow \mathsf{SampleRight}(\mathbf{A}, \mathbf{S}_\theta, 1, \mathbf{G}, \mathbf{T}_\mathbf{G}, \mathbf{U} - \mathbf{U}_I, s).$$
   After determining $\mathbf{U}_I$, as we have: $\mathbf{A}_{\overrightarrow{x}} = \sum_{i=1}^{\ell} \mathbf{A}_i \mathbf{G}^{-1}(x_i \cdot \mathbf{G}) = \mathbf{A}\big(\sum_{i=1}^{\ell} \mathbf{R}_i \mathbf{G}^{-1}(x_i \cdot \mathbf{G})\big) - \underbrace{\langle \overrightarrow{x}, \overrightarrow{y}^* \rangle}_{\neq 0} \cdot \mathbf{G}$,
   we can sample matrix
   $$\mathbf{Z} \leftarrow \mathsf{SampleRight}(\mathbf{A}, \sum_{i=1}^{\ell} \mathbf{R}_i \mathbf{G}^{-1}(x_i \cdot \mathbf{G}), -\langle \overrightarrow{x}, \overrightarrow{y}^* \rangle, \mathbf{G}, \mathbf{T}_\mathbf{G}, \mathbf{U}_I, s)$$
   satisfying $[\mathbf{A} \mid \mathbf{A}_{\overrightarrow{x}}] \cdot \mathbf{Z} = \mathbf{U}_I$.
2. **Case 2:** $\langle \overrightarrow{x}, \overrightarrow{y}^* \rangle = 0$.
   In this case, the condition $I \in \mathsf{RL}^*$ implies that $\mathsf{Path}(I) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}^*) = \emptyset$. Note that, here we do not have a trapdoor for the matrix $[\mathbf{A} \mid \mathbf{A}_{\overrightarrow{x}}]$, but we can instead compute $\mathbf{Z}$ and $\{\mathbf{Z}_\theta\}_{\theta\in\mathsf{Path}(I)}$ as follows. First, we sample $\mathbf{Z} \hookleftarrow \big(\mathcal{D}_{\mathbb{Z}^{2m},s}\big)^k$ and set $\mathbf{U}_I = [\mathbf{A} \mid \mathbf{A}_{\overrightarrow{x}}] \cdot \mathbf{Z}$. Then, for each $\theta \in \mathsf{Path}(I)$, we sample
   $$\mathbf{Z}_\theta \leftarrow \mathsf{SampleRight}(\mathbf{A}, \mathbf{S}_\theta, 1, \mathbf{G}, \mathbf{T}_\mathbf{G}, \mathbf{U} - \mathbf{U}_I, s).$$

$\mathsf{Sim.Enc}(\mathsf{msk}^*, M, \mathbf{d}_0, \mathbf{d}')$: On input msk^*, a message $M \in \{0,1\}$, and the extra inputs $\mathbf{d}_0 \in \mathbb{Z}_q^m$, $\mathbf{d}' \in \mathbb{Z}_q^k$, this algorithm outputs $\mathsf{ct} = \big(\mathbf{c}', \mathbf{c}_0, \{\mathbf{c}_i\}_{i\in[\ell]}, \{\mathbf{c}_\theta\}_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{RL})}\big)$, where:

$$\begin{aligned}
\mathbf{c}' &= \mathbf{d}' + \lfloor \frac{q}{2} \rfloor \cdot \mathsf{encode}(M) \in \mathbb{Z}_q^k, \\
\mathbf{c}_0 &= \mathbf{d}_0 \in \mathbb{Z}_q^m, \\
\forall i \in [\ell]: \quad \mathbf{c}_i &= \mathbf{R}_i^\top \mathbf{d}_0 \in \mathbb{Z}_q^m, \\
\forall \theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}^*): \quad \widehat{\mathbf{c}}_\theta &= \mathbf{S}_\theta^\top \mathbf{d}_0 \in \mathbb{Z}_q^m.
\end{aligned}$$

The series of games. Let $\mathcal{A}$ be the adversary in the selective full-hiding game of Definition 3. We consider the following series of games.

- $\mathrm{Game}_0^{(b)}$: This game is the real security game in Definition 3, where the chosen bit is $b \in \{0,1\}$.
- $\mathrm{Game}_1^{(b)}$: This game is the same as $\mathrm{Game}_0^{(b)}$, except that algorithms $\mathsf{Setup}(1^\lambda)$, $\mathsf{Enc}(\overrightarrow{y}^{(b)}, \mathsf{RL}^{(b)}, M^{(b)})$ are replaced by

$$\mathsf{Sim.Setup}\big(1^\lambda, \mathbf{A}, \mathbf{U}, \overrightarrow{y}^{(b)}, \mathsf{RL}^{(b)}\big), \quad \mathsf{Sim.Enc}\big(\mathsf{msk}^*, M^{(b)}, \mathbf{A}^\top\mathbf{s}+\mathbf{e}, \mathbf{U}^\top\mathbf{s}+\mathbf{e}'\big),$$

  respectively, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \hookleftarrow \chi^m$, and $\mathbf{e}' \hookleftarrow \chi^k$.
- $\mathrm{Game}_2^{(b)}$: This game is the same as $\mathrm{Game}_1^{(b)}$, except that algorithm $\mathsf{KeyGen}\,(\mathsf{msk}, \mathsf{ST}, \overrightarrow{x}, I)$ is replaced by algorithm $\mathsf{Sim.KeyGen}\big(\mathsf{msk}^*, \mathsf{ST}, \overrightarrow{x}, I, \overrightarrow{y}^{(b)}, \mathsf{RL}^{(b)}\big)$.
- $\mathrm{Game}_3^{(b)}$: This game is the same as $\mathrm{Game}_2^{(b)}$, except that $\mathsf{Sim.Enc}\big(\mathsf{msk}^*, M^{(b)}, \mathbf{d}_0, \mathbf{d}'\big)$ takes as inputs $\mathbf{d}_0 \xleftarrow{\$} \mathbb{Z}_q^m$ and $\mathbf{d}' \xleftarrow{\$} \mathbb{Z}_q^k$.
- Game_4: In this final game, we make the following changes:
  - $\mathsf{Sim.Setup}\big(1^\lambda, \mathbf{A}, \mathbf{U}, \overrightarrow{y}^{(b)}, \mathsf{RL}^{(b)}\big)$ is replaced by $\mathsf{Setup}(1^\lambda)$.
  - $\mathsf{Sim.KeyGen}\big(\mathsf{msk}^*, \mathsf{ST}, \overrightarrow{x}, I, \overrightarrow{y}^{(b)}, \mathsf{RL}^{(b)}\big)$ is replaced by $\mathsf{KeyGen}\big(\mathsf{msk}, \mathsf{ST}, \overrightarrow{x}, I\big)$.
  - Instead of computing $\mathbf{c}' = \mathbf{d}' + \lfloor \frac{q}{2} \rfloor \cdot \mathsf{encode}(M^{(b)}) \in \mathbb{Z}_q^k$, we sample $\mathbf{c}' \xleftarrow{\$} \mathbb{Z}_q^k$.

To prove Theorem 1, we will first demonstrate in the following lemmas that any two consecutive games in the above series are either statistically indistinguishable or computationally indistinguishable under the LWE assumption.

Lemma 6. *The adversary $\mathcal{A}$'s view in* $\mathrm{Game}_0^{(b)}$ *is statistically close to the view in* $\mathrm{Game}_1^{(b)}$.

Proof. We will show that $\mathsf{pp} = \big(\mathbf{A}, \{\mathbf{A}_i\}_{i\in[\ell]}, \mathbf{U}, \mathsf{BT}\big)$ and $\mathsf{ct} = \big(\mathbf{c}', \mathbf{c}_0, \{\mathbf{c}_i\}_{i\in[\ell]}, \{\widehat{\mathbf{c}}_\theta\}_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{RL})}\big)$ produced by algorithms $\mathsf{Sim.Setup}\big(1^\lambda, \mathbf{A}, \mathbf{U}, \overrightarrow{y}^{(b)}, \mathsf{RL}^{(b)}\big)$ and $\mathsf{Sim.Enc}(\mathsf{msk}^*, M^{(b)}, \mathbf{A}^\top\mathbf{s}+\mathbf{e}, \mathbf{U}^\top\mathbf{s}+\mathbf{e}')$ in $\mathrm{Game}_1^{(b)}$ are statistically close to those produced by algorithms Setup and Enc, respectively, in $\mathrm{Game}_0^{(b)}$.

First of all, we observe that matrix $\mathbf{A}$ is truly uniform in $\mathrm{Game}_1^{(b)}$. In $\mathrm{Game}_0^{(b)}$, it is generated via algorithm $\mathsf{TrapGen}$, and hence, is statistically close to uniform over $\mathbb{Z}_q^{n\times m}$ by Lemma 1. Furthermore, we note that matrix $\mathbf{U} \in \mathbb{Z}_q^{n\times k}$ is truly uniform in both games.

Let $\overrightarrow{y}^{(b)} = (y_1^{(b)}, \ldots, y_\ell^{(b)})$. For each $i \in [\ell]$ and each $\theta \in \mathsf{BT}$, the matrices $\mathbf{A}_i, \mathbf{D}_\theta \in \mathbb{Z}_q^{n\times m}$ are truly uniform in $\mathrm{Game}_0^{(b)}$, while in $\mathrm{Game}_1^{(b)}$, they are generated as:

$$\mathbf{A}_i = \mathbf{A}\mathbf{R}_i - y_i^{(b)} \cdot \mathbf{G}; \quad \mathbf{D}_\theta = \mathbf{A}\mathbf{S}_\theta + \rho_\theta \cdot \mathbf{G},$$

where $\mathbf{R}_i, \mathbf{S}_\theta \xleftarrow{\$} \{-1,1\}^{m\times m}$ and $\rho_\theta \in \{0,1\}$. Then, the ciphertext components $\mathbf{c}'$, $\mathbf{c}_0$, $\{\mathbf{c}_i\}_{i\in[\ell]}$ and $\{\widehat{\mathbf{c}}_\theta\}_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{RL})}$ in both games can be expressed as:

$$\begin{cases} \mathbf{c}' = \mathbf{U}^\top\mathbf{s} + \mathbf{e}' + \lfloor \frac{q}{2} \rfloor \cdot \mathsf{encode}(M^{(b)}) \in \mathbb{Z}_q^k, \\ \mathbf{c}_0 = \mathbf{A}^\top\mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m, \\ \mathbf{c}_i = (\mathbf{A}_i + y_i^{(b)} \cdot \mathbf{G})^\top\mathbf{s} + \mathbf{R}_i^\top\mathbf{e} = \mathbf{R}_i^\top(\mathbf{A}^\top\mathbf{s}+\mathbf{e}) \in \mathbb{Z}_q^m, \quad \forall i \in [\ell], \\ \widehat{\mathbf{c}}_\theta = \mathbf{D}_\theta^\top\mathbf{s} + \mathbf{S}_\theta^\top\mathbf{e} = \mathbf{S}_\theta^\top(\mathbf{A}^\top\mathbf{s}+\mathbf{e}) \in \mathbb{Z}_q^m, \quad \forall \theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}^{(b)}), \end{cases}$$

where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}' \hookleftarrow \chi^k$ and $\mathbf{e} \hookleftarrow \chi^m$. By Lemma 5, the joint distributions of:

$$\left(\mathbf{A},\ \mathbf{A}\mathbf{R}_i - \overrightarrow{y}_i^{(b)} \cdot \mathbf{G},\ \mathbf{R}_i^\top\mathbf{e}\right) \quad \text{and} \quad \left(\mathbf{A},\ \mathbf{A}_i,\ \mathbf{R}_i^\top\mathbf{e}\right),$$

as well as

$$\left(\mathbf{A},\ \mathbf{A}\mathbf{S}_\theta + \rho_\theta \cdot \mathbf{G},\ \mathbf{S}_\theta^\top\mathbf{e}\right) \quad \text{and} \quad \left(\mathbf{A},\ \mathbf{D}_\theta,\ \mathbf{S}_\theta^\top\mathbf{e}\right)$$

as statistically indistinguishable.

The above discussions imply that the distributions of

$$\left(\mathbf{A}, \{\mathbf{A}_i\}_{i\in[\ell]},\ \mathbf{U}, \{\mathbf{D}_\theta\}_{\theta\in\mathsf{BT}},\ \mathbf{c}',\ \mathbf{c}_0,\ \{\mathbf{c}_i\}_{i\in[\ell]},\ \{\widehat{\mathbf{c}}_\theta\}_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{RL})}\right)$$

in $\mathrm{Game}_0^{(b)}$ and $\mathrm{Game}_1^{(b)}$ are statistically indistinguishable. This concludes the lemma. □

Lemma 7. *The adversary $\mathcal{A}$'s view in $\mathrm{Game}_1^{(b)}$ is statistically close to the view in $\mathrm{Game}_2^{(b)}$.*

Proof. Recall that, from $\mathrm{Game}_1^{(b)}$ to $\mathrm{Game}_2^{(b)}$, we replace the real key generation algorithm KeyGen by algorithm $\mathsf{Sim.KeyGen}$. Thus, we need to demonstrate that for all queries of the form $(\overrightarrow{x}, I)$ from the adversary $\mathcal{A}$, the private keys $\mathsf{sk}_{\overrightarrow{x},I} = \left(I, \mathbf{Z}, \{\mathbf{Z}_\theta\}_{\theta\in\mathsf{Path}(I)}\right)$ outputted by $\mathsf{Sim.KeyGen}$ and KeyGen are statistically indistinguishable.

We first note that, in both cases, matrices $\mathbf{Z} \in \mathbb{Z}^{2m\times k}, \{\mathbf{Z}_\theta \in \mathbb{Z}^{2m\times k}\}_{\theta\in\mathsf{Path}(I)}$ satisfy the condition:

$$[\mathbf{A} \mid \mathbf{A}_{\overrightarrow{x}}] \cdot \mathbf{Z} + [\mathbf{A} \mid \mathbf{D}_\theta] \cdot \mathbf{Z}_\theta = \mathbf{U}, \quad \forall \theta \in \mathsf{Path}(I).$$

Next, we observe that, in KeyGen, the columns of these matrices are sampled via algorithm $\mathsf{SampleLeft}$, while in $\mathsf{Sim.KeyGen}$, they are either sampled via algorithm $\mathsf{SampleRight}$ or sampled from $\mathcal{D}_{\mathbb{Z}^m,s}$. The properties of these sampling algorithms (see Section 2) then guarantee that the two distributions are statistically indistinguishable. □

Lemma 8. *Under the (n,q,χ)-LWE assumption, the adversary $\mathcal{A}$'s view in $\mathrm{Game}_2^{(b)}$ is computationally indistinguishable from the view in $\mathrm{Game}_3^{(b)}$.*

Proof. From $\mathrm{Game}_2^{(b)}$ to $\mathrm{Game}_3^{(b)}$, we change the inputs $\mathbf{d}_0, \mathbf{d}'$ to algorithm $\mathsf{Sim.Enc}$ from LWE instances to uniformly random vectors in $\mathbb{Z}_q^m$ and $\mathbb{Z}_q^k$, respectively. Suppose that the adversary $\mathcal{A}$ has non-negligible advantage in distinguishing $\mathrm{Game}_2^{(b)}$ from $\mathrm{Game}_3^{(b)}$. We use $\mathcal{A}$ to construct an LWE solver $\mathcal{B}$ that proceeds as follows:

- $\mathcal{B}$ requests for $m+k$ instances $\{(\mathbf{a}_j, v_j) \in \mathbb{Z}_q^n \times \mathbb{Z}_q\}_{j\in[m+k]}$ from the LWE challenger.
- $\mathcal{B}$ forms the following matrices and vectors

$$\mathbf{A} := [\mathbf{a}_1, \ldots, \mathbf{a}_m] \in \mathbb{Z}_q^{n\times m}, \qquad \mathbf{U} := [\mathbf{a}_{m+1}, \ldots, \mathbf{a}_{m+k}] \in \mathbb{Z}_q^{n\times k},$$
$$\mathbf{d}_0 := [v_1, \cdots, v_m]^\top \in \mathbb{Z}_q^m, \qquad \mathbf{d}' := [v_{m+1}, \cdots, v_{m+k}]^\top \in \mathbb{Z}_q^k,$$

and runs $\mathsf{Sim.Setup}\left(1^\lambda, \mathbf{A}, \mathbf{U}, \overrightarrow{y}^{(b)}, \mathsf{RL}^{(b)}\right)$ as in $\mathrm{Game}_2^{(b)}$.

- $\mathcal{B}$ answers the private key queries by running the $\mathsf{Sim.KeyGen}\big(\mathsf{msk}^*, \mathsf{ST}, \overrightarrow{x}, I, \overrightarrow{y}^{(b)}, \mathsf{RL}^{(b)}\big)$ algorithm as in $\mathrm{Game}_2^{(b)}$.
- When receiving from $\mathcal{A}$ two messages $M^{(0)}, M^{(1)} \in \{0,1\}$, $\mathcal{B}$ prepares a challenge ciphertext ct^* by running $\mathsf{Sim.Enc}\left(\mathsf{msk}^*, M^{(b)}, \mathbf{d}_0, \mathbf{d}'\right)$.
- Finally, after being allowed to make additional queries, $\mathcal{A}$ guesses whether it is interacting with $\mathrm{Game}_2^{(b)}$ or $\mathrm{Game}_3^{(b)}$. Then, $\mathcal{B}$ outputs $\mathcal{A}$'s guess as the answer to the LWE challenger.

Recall that by Definition 1, for each $j \in [m+k]$, either $v_j = \langle \mathbf{a}_j, \mathbf{s} \rangle + e_j$ for secret $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and noise $e_j \hookleftarrow \chi$; or v_j is uniformly random in $\mathbb{Z}_q$. On the one hand, if $v_j = \langle \mathbf{a}_j, \mathbf{s} \rangle + e_j$, then the adversary $\mathcal{A}$'s view is as in $\mathrm{Game}_2^{(b)}$. On the other hand, if v_j is uniformly random in $\mathbb{Z}_q$, then $\mathcal{A}$'s view is as in $\mathrm{Game}_2^{(b)}$. Hence, algorithm $\mathcal{B}$ can solve the (n, q, χ)-LWE problem with non-negligible probability, assuming that the adversary $\mathcal{A}$ can distinguish $\mathrm{Game}_2^{(b)}$ from $\mathrm{Game}_3^{(b)}$ with non-negligible advantage. This concludes the lemma. □

Lemma 9. *The adversary $\mathcal{A}$'s view in $\mathrm{Game}_3^{(b)}$ is statistically close to the view in Game_4.*

Proof. Firstly, based on the same argument as in Lemma 6, we can deduce that the output of algorithm $\mathsf{Sim.Setup}\big(1^\lambda, \mathbf{A}, \mathbf{U}, \overrightarrow{y}^{(b)}, \mathsf{RL}^{(b)}\big)$ in $\mathrm{Game}_3^{(b)}$ is statistically close that of $\mathsf{Setup}(1^\lambda)$ in Game_4.

Secondly, based on the same argument as in Lemma 7, we can deduce that the output of algorithm $\mathsf{Sim.KeyGen}\big(\mathsf{msk}^*, \mathsf{ST}, \overrightarrow{x}, I, \overrightarrow{y}^{(b)}, \mathsf{RL}^{(b)}\big)$ in $\mathrm{Game}_3^{(b)}$ is statistically close to that of $\mathsf{KeyGen}\big(\mathsf{msk}, \mathsf{ST}, \overrightarrow{x}, I\big)$ in Game_4.

Finally, the shift from $\mathbf{c}' = \mathbf{d}' + \lfloor \frac{q}{2} \rfloor \cdot \mathsf{encode}(M^{(b)}) \in \mathbb{Z}_q^k$ to a uniformly random $\mathbf{c}' \in \mathbb{Z}_q^k$ is only a conceptual change, because vector $\mathbf{d}'$ in $\mathrm{Game}_3^{(b)}$ is uniformly random over $\mathbb{Z}_q^k$. □

The theorem now follows from the fact that the advantage of $\mathcal{A}$ in Game_4 is 0, since Game_4 no longer depends on the bit b. □

4 Extensions and Open Questions

In this section, we discuss several possible extensions of our lattice-based RPE scheme, as well as some questions that we left open.

4.1 Extensions

Multi-bit version. The scheme presented in Section 3 only allows to encrypt 1-bit messages. Using standard techniques for multi-bit LWE-based encryption, e.g., [38,15,1], we can achieve a τ-bit variant with small overhead, for any $\tau = \mathsf{poly}(\lambda)$. A notable change in this case is that we will employ a revised encoding function $\mathsf{encode}' : \{0,1\}^\tau \to \{0,1\}^{\tau+k}$, where for any $\mu \in \{0,1\}^\tau$, vector $\mathsf{encode}'(\mu)$ is obtained by appending $k = \omega(\log \lambda)$ entries 0 to vector μ.

Better efficiency in the random oracle model. The RPE scheme from Section 3 has relatively large public parameters pp, i.e., of bit-size $\big(\widetilde{O}(\ell) + O(N)\big) \cdot \widetilde{O}(\lambda^2)$, for which the dependence on N is due to the fact that we have to associate each node θ in the binary tree with a uniformly random matrix in $\mathbf{D}_\theta \in \mathbb{Z}_q^{n \times m}$, in order to obtain full-hiding security in the standard model. Fortunately, the size of pp can be reduced to $\widetilde{O}(\ell) \cdot \widetilde{O}(\lambda^2)$ (which is comparable to that of the underlying PE scheme [2,51]), if we work in the random oracle model [7]. The idea is as follows.

Let $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q^{n \times m}$ be a random oracle. Then, in the scheme, for each node θ, we obtain uniformly random matrix $\mathbf{D}_\theta$ as $\mathbf{D}_\theta := \mathcal{H}(\mathbf{A}, \{\mathbf{A}_i\}_{i \in [\ell]}, \mathbf{U}, \theta)$. The rest of the scheme remains the same. In the security proof, we first simulate the generation of $\mathbf{D}_\theta$ as in the proof of Theorem 1. Then, it remains to program the random oracle such that $\mathcal{H}(\mathbf{A}, \{\mathbf{A}_i\}_{i \in [\ell]}, \mathbf{U}, \theta) := \mathbf{D}_\theta$. This modification allows us to make the size of pp independent of N.

4.2 Open Questions

We introduced the first revocable predicate encryption scheme based on the LWE assumption. While the pairing-based scheme from [31,32] achieved adaptive full-hiding security, our construction is only proven secure in the selective setting. Achieving the stronger notion of [31,32] seems to require that the underlying PE be adaptively secure. However, to the best of our knowledge, existing lattice-based PE schemes [2,51,13,17] only achieved selective security. We therefore view the problem of constructing adaptively secure lattice-based RPE as an interesting open question.

Finally, as shown in [20,51], some applications of PE for inner-product predicate over R^ℓ (in our scheme, $R = \mathbb{Z}_q$) require that R has exponentially large cardinality. Those include implementations of PE for CNF formulae [20] and hidden vector encryption [10]. However, for our scheme, this requires to set the modulus q to be exponential in λ. Hence, it would be desirable to achieve a lattice-based PE scheme supporting both revocation and exponentially large R, that demands only polynomial moduli. One possible approach towards tackling this question is to adapt the techniques introduced by Xagawa [51], where one works with $R = \mathrm{GF}(q^n)$ instead of $\mathbb{Z}_q$.

Acknowledgements. We thank the reviewers for helpful discussions and comments. The research was supported by the "Singapore Ministry of Education under Research Grant MOE2016-T2-2-014(S)". Huaxiong Wang was also supported by NTU under Tier 1 grant RG143/14.

References

1. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, 2010.
2. Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In *ASIACRYPT 2011*, volume 7073 of *LNCS*. Springer, 2011.
3. Miklós Ajtai. Generating hard instances of the short basis problem. In *ICALP 1999*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.
4. Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. *Theory of Computing Systems*, 48(3):535–553, 2011.
5. Nuttapong Attrapadung and Hideki Imai. Attribute-based encryption supporting direct/indirect revocation modes. In *Cryptography and Coding 2009*, pages 278–300. Springer, 2009.
6. Nuttapong Attrapadung and Benoît Libert. Functional encryption for inner product: achieving constant-size ciphertexts with adaptive security or support for negation. In *PKC 2010*, volume 6056 of *LNCS*, pages 384–402. Springer, 2010.
7. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS 1993*, pages 62–73. ACM, 1993.
8. Alexandra Boldyreva, Vipul Goyal, and Virendra Kumar. Identity-based encryption with efficient revocation. In *CCS 2008*, pages 417–426. ACM, 2008.
9. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.
10. Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC 2007*, volume 4392 of *LNCS*, pages 535–554. Springer, 2007.
11. Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Khoa Nguyen. Revocable identity-based encryption from lattices. In *ACISP 2012*, volume 7372 of *LNCS*, pages 390–403. Springer, 2012.
12. Shantian Cheng and Juanyang Zhang. Adaptive-ID secure revocable identity-based encryption from lattices via subset difference method. In *ISPEC 2015*, volume 9065 of *LNCS*, pages 283–297. Springer, 2015.
13. Romain Gay, Pierrick Méaux, and Hoeteck Wee. Predicate encryption for multi-dimensional range queries from lattices. In *PKC 2015*, volume 9020 of *LNCS*, pages 752–776. Springer, 2015.
14. Nicholas Genise and Daniele Micciancio. Faster gaussian sampling for trapdoor lattices with arbitrary modulus. *IACR Cryptology ePrint Archive*, 2017:308, 2017.
15. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC 2008*, pages 197–206. ACM, 2008.
16. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Predicate encryption for circuits from LWE. In *CRYPTO 2015*, number 9216 in LNCS, pages 503–523. Springer, 2015.

17. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In *CRYPTO 2015*, volume 9216 of *LNCS*, pages 503–523. Springer, 2015.
18. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *CCS 2006*, pages 89–98. ACM, 2006.
19. Kamil Doruk Gur, Yuriy Polyakov, Kurt Rohloff, Gerard W. Ryan, and Erkay Savas. Implementation and evaluation of improved gaussian sampling for lattice trapdoors. *IACR Cryptology ePrint Archive*, 2017:285, 2017.
20. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, 2008.
21. Kwangsu Lee, Intae Kim, and Seong Oun Hwang. Privacy preserving revocable predicate encryption revisited. *Security and Communication Networks*, 8(3):471–485, 2015.
22. Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 62–91. Springer, 2010.
23. Benoît Libert, Thomas Peters, and Moti Yung. Group signatures with almost-for-free revocation. In *CRYPTO 2012*, volume 7417 of *LNCS*, pages 571–589. Springer, 2012.
24. Benoît Libert, Thomas Peters, and Moti Yung. Scalable group signatures with revocation. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 609–627. Springer, 2012.
25. Benoît Libert and Damien Vergnaud. Adaptive-ID secure revocable identity-based encryption. In *CT-RSA 2009*, volume 5473 of *LNCS*, pages 1–15. Springer, 2009.
26. Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In *CRYPTO 2011*, volume 6841 of *LNCS*, pages 465–484. Springer, 2011.
27. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012.
28. Daniele Micciancio and Michael Walter. Gaussian sampling over the integers: Efficient, generic, constant-time. *IACR Cryptology ePrint Archive*, 2017:259, 2017.
29. Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, 2001.
30. Khoa Nguyen, Huaxiong Wang, and Juanyang Zhang. Server-aided revocable identity-based encryption from lattices. In *CANS 2016*, volume 10052 of *LNCS*, pages 107–123. Springer, 2016.
31. Juan Manuel González Nieto, Mark Manulis, and Dongdong Sun. Fully pivate revocable predicate encryption. In *ACISP 2012*, volume 7372 of *LNCS*, pages 350–363. Springer, 2012.
32. Juan Manuel González Nieto, Mark Manulis, and Dongdong Sun. Fully private revocable predicate encryption. *IACR Cryptology ePrint Archive*, 2012:403, 2012.
33. Tatsuaki Okamoto and Katsuyuki Takashima. Hierarchical predicate encryption for inner-products. In *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 214–231. Springer, 2009.
34. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, 2010.
35. Tatsuaki Okamoto and Katsuyuki Takashima. Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption. In *CANS 2011*, volume 7092 of *LNCS*, pages 138–159. Springer, 2011.
36. Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 591–608. Springer, 2012.
37. Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC 2009*, pages 333–342. ACM, 2009.
38. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO 2008*, volume 5157 of *LNCS*, pages 554–571. Springer, 2008.
39. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC 2005*, pages 84–93. ACM, 2005.
40. Amit Sahai, Hakan Seyalioglu, and Brent Waters. Dynamic credentials and ciphertext delegation for attribute-based encryption. In *CRYPTO 2012*, volume 7417 of *LNCS*, pages 199–217. Springer, 2012.
41. Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, 2005.
42. Jae Hong Seo and Keita Emura. Efficient delegation of key generation and revocation functionalities in identity-based encryption. In *CT-RSA 2013*, volume 7779 of *LNCS*, pages 343–358. Springer, 2013.
43. Jae Hong Seo and Keita Emura. Revocable identity-based encryption revisited: ecurity model and construction. In *PKC 2013*, volume 7778 of *LNCS*, pages 216–234. Springer, 2013.
44. Jae Hong Seo and Keita Emura. Revocable hierarchical identity-based encryption. *Theor. Comput. Sci.*, 542:44–62, 2014.

45. Jae Hong Seo and Keita Emura. Revocable identity-based cryptosystem revisited: security sodels and constructions. *IEEE Trans. Information Forensics and Security*, 9(7):1193–1205, 2014.
46. Jae Hong Seo and Keita Emura. Adaptive-ID secure revocable hierarchical identity-based encryption. In *IWSEC 2015*, pages 21–38, 2015.
47. Elaine Shi, John Bethencourt, Hubert T.-H. Chan, Dawn Xiaodong Song, and Adrian Perrig. Multi-dimensional range query over encrypted data. In *IEEE Symposium on Security and Privacy (S&P 2007)*, pages 350–364. IEEE Computer Society, 2007.
48. Atsushi Takayasu and Yohei Watanabe. Lattice-based revocable identity-based encryption with bounded decryption key exposure resistance. In *ACISP 2017*, volume 10342 of *LNCS*, pages 184–204. Springer, 2017.
49. Yohei Watanabe, Keita Emura, and Jae Hong Seo. New revocable IBE in prime-order groups: adaptively secure, decryption key exposure resistant, and with short public parameters. In *CT-RSA 2017*, volume 10159 of *LNCS*, pages 432–449. Springer, 2017.
50. Brent Waters. Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, 2009.
51. Keita Xagawa. Improved (hierarchical) inner-product encryption from lattices. In *PKC 2013*, volume 7778 of *LNCS*, pages 235–252. Springer, 2013.