Inner-Product Lossy Trapdoor Functions and Applications

Xiang Xie[1,2,⋆], Rui Xue[3], and Rui Zhang[3,⋆⋆]

[1] Institute of Software, Chinese Academy of Sciences
[2] Graduate University of Chinese Academy of Sciences
[3] The State Key Laboratory of Information Security
Institute of Information Engineering, Chinese Academy of Sciences
xiexiang@is.iscas.ac.cn, {rxue,r-zhang}@iie.ac.cn

Abstract. In this paper, we propose a new cryptographic primitive called inner-product lossy trapdoor function (IPLTDF). We give a formal definition, and a concrete construction from lattices. We then show this primitive is useful to obtain efficient chosen-plaintext secure inner-product encryption (IPE) schemes. The resulting IPE scheme has almost the same public key size for multi-bit encryption compared with a recent IPE scheme proposed by Agrawal, Freeman and Vaikuntanathan [2] for single-bit encryption. Unfortunately, our IPE scheme only supports attribute vectors with logarithmic length. On the positive side, our basic IPE scheme can be extended to achieve chosen-ciphertext (CCA) security. As far as we are aware, this is the first CCA-secure IPE scheme based on lattices.

Keywords: inner-product encryption, inner-product lossy trapdoor functions, lattices.

1 Introduction

In a traditional public key encryption system, data encrypted under a public key can only be decrypted by an receiver with the corresponding secret key. Inspired by a seminal work by Sahai and Waters [29], researchers have focused on more fine-grained encryption schemes, which led to the notion of *functional encryption* [13]. In a functional encryption, dedicated secret keys allow users to learn functions of encrypted data. Functional encryption is a broad framework and has many concrete expressions, among which, *predicate encryption* (PE) [19] is an important one. In a PE system, the secret key sk_f corresponding to a predicate f can be used to decrypt a ciphertext associated with attribute I if and only if $f(I) = 1$. A useful set of predicates for PE is called *inner-product*

⋆ Supported by the Fund of the National Natural Science Foundation of China under Grants No. 61170280.
⋆⋆ Supported by the One Hundred Person Project of the Chinese Academy of Sciences and the Fund of the National Natural Science Foundation of China under Grants No. 61100225.

F. Bao, P. Samarati, and J. Zhou (Eds.): ACNS 2012, LNCS 7341, pp. 188–205, 2012.

predicates. In an inner-product encryption (IPE) system, an attribute of inner-product predicates is expressed as vector $\mathbf{x}$ and predicate $f_{\mathbf{v}}$ is associated with vector $\mathbf{v}$, where $f_{\mathbf{v}}(\mathbf{x}) = 1$ iff $\langle \mathbf{x}, \mathbf{v} \rangle = 0$.

Katz, Sahai, and Waters defined the notion of predicate encryption and gave the first construction of IPE. However, their construction was based on complicated assumptions. Subsequently, Okamoto and Takashima [23] showed how to construct hierarchical IPE schemes. All the previous constructions are secure under selective adversaries until [20]. Lewko et al. [20] gave the first adaptively secure IPE scheme, which was further improved in [24,25]. All the above constructions made use of bilinear pairings. Very recently, Agrawal, Freeman and Vaikuntanathan [2] proposed the first IPE scheme under the *worst-case* lattice assumption.

In this work, we seek for a different way to construct IPE. We introduce a new primitive called inner-product lossy trapdoor functions (IPLTDFs). We define and construct IPLTDFs. Thanks to its lossiness, we can easily obtain IPE schemes from IPLTDFs via hardcore bits.

In a high level, for chosen public parameters pp and master secret key, an inner-product trapdoor function (IPTDF) F associated with any attribute vector $\mathbf{x}$ is an injective, deterministic map $F_{pp,\mathbf{x}}$ which can be inverted given a secret key derivable from a predicate vector $\mathbf{v}$ via the master secret key if and only if $\langle \mathbf{x}, \mathbf{v} \rangle = 0$. Suppose there is an another algorithm that generates "fake" public parameters pp^*, such that, for an adversary-specified challenge attribute vector $\mathbf{x}^*$, $F_{pp^*,\mathbf{x}^*}$ is no longer injective but has image much smaller than its domain. Moreover, given public parameters, one should not be able to tell whether it is real or fake. Importantly, as in inner-product encryption, this must hold even when the adversary may obtain, via a key-derivation oracle, an inversion key for predicates $\mathbf{v}$ with $\langle \mathbf{v}, \mathbf{x}^* \rangle \neq 0$. As with IPE, security may be selective (the adversary must specify $\mathbf{x}^*$ before seeing pp) or adaptive (no such restriction). In this paper, we only consider the selective security.

In order to build secure IPTDFs, an intuitive idea is to apply the matrix-based framework of [27] and encrypt each matrix entry with an IPE scheme. For already complicated IPE schemes this method brings us more complicated analysis. Alternatively, we derive one-way IPTDFs by applying the ideas from [2], and then show how to make it lossy which is non-trivial. Our solution shows that secure inner-product lossy trapdoor function (IPLTDF) can be achieved in principle, which was not clear prior to our work.

1.1 Our Contributions

In this work, we define the notion of inner-product lossy trapdoor functions, and we give a concrete construction of IPTDF based on lattices. However, to make the scheme correct and lossy simultaneously, our IPTDF only supports attribute vectors with logarithmic length. we then use it to construct a chosen-plaintext secure IPE scheme for multi-bit encryption based on lattices with public key size almost the same as the scheme presented by Agrawal, Freeman, and Vaikuntanathan [2] for single-bit encryption. Unfortunately, in order to invert

correctly, we have to append the attribute vector after the function value of our IPTDFs, which limits our IPE scheme only achieves payload hiding security. We leave it as a future work to construct IPTDFs whose attribute information is hidden in the function value using our methodology.

As an interesting observation, we note that the information of the lossy attribute actually is hidden in the public parameters of the lossy IPTDFs. This property is also satisfied in identity-based trapdoor functions (IBTDFs). I.e., the public parameters of lossy IBTDFs hide the information of the lossy identity. Under the framework presented by Peikert and Waters [27], we obtain the first chosen-ciphertext secure IPE from lattices. As a by-product, we also observe that lossy IBTDFs are actually All-But-One (ABO) trapdoor functions [27]. Combine our IPTDF and the concrete construction of lossy IBTDF in [7] from lattices. We get a chosen-ciphertext secure IPE scheme with public size almost twice as our chosen-plaintext secure IPE scheme.

1.2 Related Works

Many encryption schemes of different types can be included in the framework of inner-product encryption. Identity-based encryption (IBE) [30,10,11] and hidden-vector encryption (HVE) [14] can be viewed as a special case of inner-product predicates encryption with equality-test predicates. Attribute-based encryption (ABE) [29,18,8] where policies are given by CNF or DNF can be implemented by inner-product encryption.

Inner-product encryption was introduced by Katz, Sahai, and Waters [19], however, their scheme only achieves selective security without delegatability (see [23]). Okamoto and Takashima [23] introduced a notion called dual pairing vector spaces (DPVS) and proposed a hierarchical IPE scheme based on DPVS, but again, only selective security is proven. To achieve adaptive security, Lewko et al. [20] adapted the dual system encryption methodology [31], and obtained the first adaptively secure IPE and hierarchical IPE schemes. Later, Okamoto and Takashima [24,25] proposed adaptively secure IPE and hierarchical IPE schemes under simpler assumptions. All these previous constructions use bilinear pairings except a recent scheme proposed by Agrawal, Freeman and Vaikuntanathan [2], which is the first IPE scheme under the *worst-case* lattice assumption. We note that the scheme in [2] seems difficult to be improved into a hierarchical IPE scheme.

The notion of lossy trapdoor functions (LTDFs) was first explicitly presented in [27]. A trapdoor function F specifies, for each public key pk, an injective, *deterministic* map F_{pk} that can be inverted given an associated trapdoor. There is an algorithm that generates a "fake" public key pk^* indistinguishable from the real one, such that F_{pk^*} has image much smaller than its domain. Peikert and Waters [27] call such a trapdoor function lossy. LTDFs was shown to be a powerful tool. Peikert and Waters [27] showed that LTDFs provided very natural constructions of many cryptographic primitives, including chosen-ciphertext secure public key encryptions, pseudo-random generators, collision-resistant hash functions, and oblivious transfer. Besides the original work of Peikert and Waters, Many other

applications of LTDFs were discovered, these include deterministic public key encryption [9], hedged public key encryption [5] and selective-opening secure public key encryption [6].

Another notion related to inner-product trapdoor function is identity-based trapdoor functions (IBTDFs) [7], which can be viewed as an identity-based version of LTDFs. In an IBTDF, each encryption function is associated with an identity and anyone has the secret key corresponding to the same identity can invert. Bellare et al. gave two constructions of IBTDFs and described two applications of IBTDFs in [7]: deterministic identity-based encryption schemes and hedged identity-based encryption schemes. Actually, IBTDFs can be viewed as a special case of our IPTDFs. More specifically, when we use the 2-dimensional attribute vector $\mathbf{x} = (id, -1)$ and predicate vector $\mathbf{v} = (1, id')$ in our IPTDFs, this is exactly the case of IBTDFs, since $\langle \mathbf{x}, \mathbf{v} \rangle = 0$ if and only if $id = id'$.

2 Notations

If x is a string, $|x|$ denotes its length. If S is a set, $|S|$ denotes its size. If S is a set then $s \leftarrow S$ denotes the operation of picking an element s of S uniformly at random. We write $z \leftarrow \mathcal{A}^{\mathcal{O}}(x)$ to indicate that $\mathcal{A}$ is an algorithm with input x and access to oracle $\mathcal{O}$ and output z. We say a function $f(n)$ is *negligible* if $f(n) < 1/n^c$ for any $c > 0$ and all sufficiently large n, denoted as $\mathrm{negl}(n)$. Let X and Y be two random variables over set S. The *statistic distance* between X and Y is defined as $\Delta(X, Y) = \frac{1}{2} \sum_{s \in S} \left| \Pr[X = s] - \Pr[Y = s] \right|$. We say X and Y are statistically indistinguishable if $\Delta(X, Y)$ is negligible.

We use bold capital letters (e.g. $\mathbf{A}$) to denote matrices, and use $\mathbf{I}_n$ to denote the identity matrix with dimension n. We use bold lowercase letters (e.g. $\mathbf{x}$) to denote vectors. $\mathbf{A}^t$ denotes the transpose of the matrix $\mathbf{A}$. When we say a matrix defined over $\mathbb{Z}_q$ has full rank, we mean that it has full rank modulo q. If $\mathbf{A}_1$ is an $n \times m$ matrix and $\mathbf{A}_2$ is an $n \times m'$ matrix, then $[\mathbf{A}_1 | \mathbf{A}_2]$ denotes the $n \times (m + m')$ matrix formed by concatenating $\mathbf{A}_1$ and $\mathbf{A}_2$. If $\mathbf{x}_1$ is a length m vector and $\mathbf{x}_2$ is a length m' vector, then we let $[\mathbf{x}_1 | \mathbf{x}_2]$ denote the length $m + m'$ vector formed by concatenating $\mathbf{x}_1$ and $\mathbf{x}_2$. When doing matrix-vector multiplication, we always view vectors as column vectors.

3 Inner-Product Lossy Trapdoor Functions

In this section, we define the notion of *inner-product trapdoor functions*. In inner-product trapdoor functions, each function value is associated with an attribute $\mathbf{x}$ and each secret key sk_f corresponds to an inner-product predicate f. A user with sk_f can invert the function value if and only if $f(\mathbf{x}) = 1$. An inner-product trapdoor function consists of four algorithms (IPTDF.Pg, IPTDF.Kg, IPTDF.Ev, IPTDF.Inv) associated with input space InSp, output space OutSp, a class of inner-product predicate functions F, and a set of attributes Σ.

IPTDF.Pg(λ) takes as input a security parameter λ. It returns public parameters PP and a master secret key msk.

$\texttt{IPTDF.Kg}(PP, f, msk)$ takes as input public parameters PP, a predicate $f \in \texttt{F}$, and a master secret key msk. It returns an inversion key sk_f for f.

$\texttt{IPTDF.Ev}(PP, \mathbf{x}, \cdot)$ which is an injective function, takes as input public parameters PP, an attribute $\mathbf{x} \in \Sigma$, and a value in $\texttt{InSp}$. It returns a function value in $\texttt{OutSp}$.

$\texttt{IPTDF.Inv}(PP, sk_f, \cdot)$ takes as input public parameters PP, a secret key sk_f for f, and a function value in $\texttt{OutSp}$. It returns a value in $\texttt{InSp}$.

For correctness, we require that $\forall(PP, msk) \leftarrow \texttt{IPTDF.Pg}(\lambda)$, $\forall f \in \texttt{F}$, $\forall sk_f \leftarrow \texttt{IPTDF.Kg}(PP, f, msk)$ and $\forall \mathbf{x} \in \Sigma$, if $C_{\mathbf{x}} \leftarrow \texttt{IPTDF.Ev}(PP, \mathbf{x}, m)$, where $m \in \texttt{InSp}$,

- If $f(\mathbf{x}) = 1$ then $\texttt{IPTDF.Inv}(PP, sk_f, C_{\mathbf{x}}) = m$.
- If $f(\mathbf{x}) = 0$ then $\texttt{IPTDF.Inv}(PP, sk_f, C_{\mathbf{x}}) = \perp$ with all but negligible probability.

An inner-product trapdoor function is associated with a sibling. An ℓ-lossy sibling $\texttt{L-IPTDF}$=($\texttt{L-IPTDF.Pg}$, $\texttt{L-IPTDF.Kg}$, $\texttt{L-IPTDF.Ev}$, $\texttt{L-IPTDF.Inv}$) differs from $\texttt{IPTDF}$ in the following sense:

1. $\texttt{L-IPTDF.Pg}(\lambda, \mathbf{x}^*)$ takes as input a security parameter λ and an attribute $\mathbf{x}^*$. It returns public parameters PP and a master secret key msk. We call $\mathbf{x}^*$ a lossy attribute.
2. $\texttt{L-IPTDF.Kg}(PP, f, msk)$ takes as input public parameters PP, a predicate f, and a master secret key msk. It returns an inversion key sk_f for all f with the requirement that $f(\mathbf{x}^*) = 0$.
3. For any $\mathbf{x} \neq \mathbf{x}^*$, $\texttt{L-IPTDF.Ev}(PP, \mathbf{x}, \cdot)$ computes an injective function over $\texttt{InSp}$, and $\texttt{L-IPTDF.Inv}(PP, sk_f, \cdot)$ computes its inversion if $f(\mathbf{x}) = 1$. Additionally, $\texttt{L-IPTDF.Ev}(PP, \mathbf{x}^*, \cdot)$ computes a function such that $\frac{|\texttt{OutSp}|}{|\texttt{InSp}|} \leq 2^{\ell}$.

We say $\texttt{IPTDF}$ is ℓ-lossy with sibling $\texttt{L-IPTDF}$, if for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, the advantage of the following game is negligible.

Experiment $\mathbf{Exp}^{sAtt\text{-}lossy}_{\texttt{IPTDF},\texttt{L-IPTDF},\mathcal{A}}(\lambda)$
$\mathbf{x}^* \leftarrow \mathcal{A}(\lambda)$;
$b \leftarrow \{0, 1\}$, if $b = 0$, $(PP_0, msk_0) \leftarrow \texttt{IPTDF.Pg}(\lambda)$;
if $b = 1$, $(PP_1, msk_1) \leftarrow \texttt{L-IPTDF.Pg}(\lambda, \mathbf{x}^*)$;
$b' \leftarrow \mathcal{A}^{\mathcal{O}(b,\cdot)}(PP_b)$;
if $b = b'$ return 1, otherwise 0.

Where oracle $\mathcal{O}(b, f)$ returns $sk_f \leftarrow \texttt{L-IPTDF.Kg}(PP_1, f, msk_1)$ when $b = 1$, and returns $sk_f \leftarrow \texttt{IPTDF.Kg}(PP_0, f, msk_0)$ when $b = 0$ with the restriction that $\mathcal{A}$ is not allowed to query f such that $f(\mathbf{x}^*) = 1$. We define the advantage of $\mathcal{A}$ in the above experiment as

$$\mathbf{Adv}^{sAtt\text{-}lossy}_{\texttt{IPTDF},\texttt{L-IPTDF},\mathcal{A}}(\lambda) = \left| \Pr[\mathbf{Exp}^{sAtt\text{-}lossy}_{\texttt{IPTDF},\texttt{L-IPTDF},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right|.$$

3.1 Lossy Attribute Hiding

We observe that the sibling L-IPTDF enjoys an interesting property. We call it *lossy attribute hiding* property. Informally, the public parameters of the L-IPTDF generated from any distinct lossy attributes are indistinguishable, even given access to obtain the inversion key of predicates that the lossy attributes do not satisfy. For any PPT adversary $\mathcal{A}$ associated with the following game:

$$\begin{aligned}&\textbf{Experiment } \mathbf{Exp}^{sAtt\text{-}lah}_{\texttt{L-IPTDF},\mathcal{A}}(\lambda)\\ &\quad \mathbf{x}_0, \mathbf{x}_1 \leftarrow \mathcal{A}(\lambda);\\ &\quad b \leftarrow \{0,1\},\ (PP_b, msk_b) \leftarrow \texttt{L-IPTDF.Pg}(\lambda, \mathbf{x}_b);\\ &\quad b' \leftarrow \mathcal{A}^{\mathcal{O}(b,\cdot)}(PP_b);\\ &\quad \text{if } b = b' \text{ return 1, otherwise 0.}\end{aligned}$$

Where oracle $\mathcal{O}(b, f)$ returns $sk_f \leftarrow \texttt{L-IPTDF.Kg}(PP_b, f, msk_b)$ with the restriction that $\mathcal{A}$ is not allowed to query f such that $f(\mathbf{x}_0) = 1$ or $f(\mathbf{x}_1) = 1$. We define the advantage of $\mathcal{A}$ in the above experiment as

$$\mathbf{Adv}^{sAtt\text{-}lah}_{\texttt{L-IPTDF},\mathcal{A}}(\lambda) = \left| \Pr[\mathbf{Exp}^{sAtt\text{-}lah}_{\texttt{L-IPTDF},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right|.$$

We say L-IPTDF is lossy attribute hiding, if for any PPT adversary $\mathcal{A}$, the above advantage is negligible.

Next, we show that the lossiness of inner-product trapdoor functions implies the lossy attribute hiding property of the corresponding sibling functions.

Lemma 1. *Let* IPTDF *be an inner-product trapdoor function, and* L-IPTDF *be its sibling. If* IPTDF *is ℓ-lossy, then* L-IPTDF *is lossy attribute hiding.*

Proof. Considering the lossy attribute hiding game that the challenger generates the public parameters and master key under $b = 0$, we denote this game as Game_0. Since IPTDF is ℓ-lossy and L-IPTDF is its sibling, no PPT adversary can tell differences with non-negligible probability if the public parameters and master key are changed from $\texttt{IPTDF.Pg}(\lambda)$, as long as the adversary do not query f with $f(\mathbf{x}_0) = 1$. Analogously, we denote the lossy attribute hiding game as Game_1 when the challenger generates the public parameters and master key under $b = 1$. No PPT adversary can tell differences with non-negligible probability if the public parameters and master key are changed from $\texttt{IPTDF.Pg}(\lambda)$, as long as the adversary do not query f with $f(\mathbf{x}_1) = 1$. We then conclude that no PPT adversary can distinguish the public parameters between Game_0 and Game_1 with non-negligible probability, with restriction that the adversary do not query f such that $f(\mathbf{x}_0) = 1$ or $f(\mathbf{x}_1) = 1$. This completes the proof. □

Remark. We can similarly define the *lossy identity hiding* property of lossy IBTDFs. This means that the information of the identity will be hidden in the public parameters of lossy IBTDFs. Analogously, the lossiness of IBTDF implies the lossy identity hiding property.

4 Inner-Product Trapdoor Functions from Lattices

Background. A full-rank m-dimensional integer lattice $\Lambda \subseteq \mathbb{Z}^m$ is a discrete additive subgroup whose linear span is $\mathbb{R}^m$. Every lattice is generated as the $\mathbb{Z}$-linear combination of some basis of linearly independent vectors i.e.,$\Lambda = \{\sum_{i=1}^{m} z_i \mathbf{b}_i : z_i \in \mathbb{Z}\}$. In this work we deal exclusively with "q-ary" lattices, where for simplicity we always take q =poly(n) to be prime. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, define the integer lattice

$$\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = 0 \mod q\}.$$

Let $\mathbf{S} = \{\mathbf{s}_1, ..., \mathbf{s}_k\}$ be a set of vectors in $\mathbb{R}^m$. We use $\widetilde{\mathbf{S}} = \{\widetilde{\mathbf{s}_1}, ..., \widetilde{\mathbf{s}_k}\}$ to denote the Gram-Schmidt orthogonalization of the vectors $\mathbf{s}_1, ..., \mathbf{s}_k$. We use $\|\mathbf{S}\|$ to denote the length of the longest vector in $\mathbf{S}$. For a real-valued matrix $\mathbf{R}$, we let $s_1(\mathbf{R})$ denote the largest singular value of $\mathbf{R}$, i.e. $s_1(\mathbf{R})$=$\max_{\|\mathbf{u}\|=1}\|\mathbf{Ru}\|$.

Let Λ be a discrete subset of $\mathbb{Z}^m$. For any vector $\mathbf{c} \in \mathbb{R}^m$ and any positive parameter $\sigma \in \mathbb{R}_{>0}$, let $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2)$ be the Gaussian function on $\mathbb{R}^m$ with center $\mathbf{c}$ and parameter σ. Let $\rho_{\sigma,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x}\in\Lambda}\rho_{\sigma,\mathbf{c}}(\mathbf{x})$ be the discrete integral of $\rho_{\sigma,\mathbf{c}}$ over Λ, and let $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$ be the discrete Gaussian distribution over Λ with center $\mathbf{c}$ and parameter σ. Specifically, for all $\mathbf{y} \in \Lambda$, we have $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}(\mathbf{y}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{y})}{\rho_{\sigma,\mathbf{c}}(\Lambda)}$. For notional convenience, $\rho_{\sigma,\mathbf{0}}$ and $\mathcal{D}_{\Lambda,\sigma,\mathbf{0}}$ are abbreviated as ρ_σ and $\mathcal{D}_{\Lambda,\sigma}$, respectively.

Security of our construction reduces to the learning with errors (LWE) problem, a classic hard problem on lattices defined by Regev [28]. The (decisional) learning with errors problem in dimension n with error rate $\alpha \in (0,1)$, stated in matrix form, is: given an input $(\mathbf{A}, \mathbf{b})$ where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ for any m=poly(n) is uniformly random and $\mathbf{b} \in \mathbb{Z}_q^m$ is either of the form $\mathbf{b} = [\mathbf{I}_m|\mathbf{A}^t]\mathbf{x} \mod q$ for $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}^{m+n}$, or is uniformly random (and independent of $\mathbf{A}$), distinguish which is the case, with non-negligible advantage.[1] By standard hybrid argument, replacing $\mathbf{x}$ with a matrix $\mathbf{X} \in \mathbb{Z}_q^{(m+n)\times\omega}$ (ω = poly(n)) whose each column sampled independently from $\mathcal{D}_{\mathbb{Z},\alpha q}^{m+n}$, and replacing $\mathbf{b}$ with either $\mathbf{B} = [\mathbf{I}_m|\mathbf{A}^t]\mathbf{X} \mod q$ or uniformly random $\mathbf{B}$ of the same dimension, yields an equivalent problem (up to a ω factor in the adversary's advantage). It is known that when $\alpha q \geq 2\sqrt{n}$, this decisional problem is at least as hard as approximating several problems on n-dimensional lattices in the *worst-case* to within $\widetilde{O}(n/\alpha)$ factors with a quantum computer [28] or on a classical computer for a subset of these problems [26]. We give some useful facts for our construction.

Lemma 2 ([22]). *Let Λ be an n-dimensional lattice, let $\mathbf{T}$ be a basis for Λ, and suppose $\sigma \geq \|\widetilde{\mathbf{T}}\| \cdot \omega(\sqrt{\log n})$. Then for any $\mathbf{c} \in \mathbb{R}^n$ we have*

$$\Pr[\|\mathbf{x}-\mathbf{c}\| > \sigma\sqrt{n} : \mathbf{x} \leftarrow \mathcal{D}_{\Lambda,\sigma,\mathbf{c}}] \leq \text{negl}(n).$$

[1] This is actually the "normal form" of the LWE problem, which is equivalent to the one from [28] in which the portion of $\mathbf{x}$ that is multiplied by $\mathbf{A}^t$ is uniformly random in $\mathbb{Z}_q^n$. The equivalence is shown in [4].

Lemma 3 ([17,21]). *For prime q and integer $b \geq 2$. Let $m \geq n \log_b q + \omega(\log n)$. For $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{R} \leftarrow \mathcal{D}_{\mathbb{Z}, b \cdot \omega(\sqrt{\log n})}^{m \times m}$. Then $(\mathbf{A}, \mathbf{AR})$ is statistically close to uniform in $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{n \times m}$.*

Lemma 4 ([3,21]). *Let q, n, m, b be positive integers with $b \geq 2$ and $m \geq 6n \log_b q$. There is a probabilistic polynomial-time algorithm $\texttt{TrapGen}(q, n, m, b)$ that outputs a pair $(\mathbf{A}, \mathbf{T}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}^{m \times m}$ such that $\mathbf{A}$ is statistically close to uniform in $\mathbb{Z}_q^{n \times m}$ and $\mathbf{T}$ is a basis for $\Lambda^{\perp}(A)$, satisfying $\|\widetilde{\mathbf{T}}\| \leq O(b \cdot \sqrt{n \log_b q})$.*

Lemma 5 ([17,21]). *Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be full-rank. Given $\mathbf{A}$ and any basis $\mathbf{T} \in \mathbb{Z}^{m \times m}$ of $\Lambda^{\perp}(\mathbf{A})$, one can efficiently recover $\mathbf{x} = [\mathbf{x}_1 | \mathbf{x}_2] \in \mathbb{Z}_q^{m+n}$ from $[\mathbf{I}_m | \mathbf{A}^t] \cdot [\mathbf{x}_1 | \mathbf{x}_2] \mod q = \mathbf{A}^t \mathbf{x}_2 + \mathbf{x}_1 \mod q$, as long as $\|\mathbf{x}_1\| \leq q/(2\|\widetilde{\mathbf{T}}\|)$.*

Lemma 6 ([1,16]). *Let $q > 2, m > n$, $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{T}_{\mathbf{A}}$ be a basis of $\Lambda^{\perp}(A)$, and $\sigma \geq \|\widetilde{\mathbf{T}_{\mathbf{A}}}\| \cdot \omega(\sqrt{\log m})$. There exists an efficient randomized algorithm $\texttt{SampleLeft}$ that, takes as inputs $\mathbf{A}, \mathbf{B}, \mathbf{T}_{\mathbf{A}}, \sigma$, and outputs a basis $\mathbf{S}$ of $\Lambda^{\perp}(\mathbf{U})$ for $\mathbf{U} = [\mathbf{A}|\mathbf{B}]$ with $\|\widetilde{\mathbf{S}}\| \leq \sigma \cdot \sqrt{2m}$ whose distribution depends on $\mathbf{U}, \sigma$.*

Lemma 7 ([1]). *Let $q > 2, m > n$, $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{B}$ be full-rank, $\mathbf{R} \in \mathbb{Z}^{m \times m}$, $\mathbf{T}_{\mathbf{B}}$ be a basis of $\Lambda^{\perp}(\mathbf{B})$, and $\sigma \geq \|\widetilde{\mathbf{T}_{\mathbf{B}}}\| \cdot s_1(\mathbf{R}) \cdot \omega(\sqrt{\log m})$. There exists an efficient randomized algorithm $\texttt{SampleRight}$ that, takes as inputs $\mathbf{A}, \mathbf{B}, \mathbf{T}_{\mathbf{B}}, \mathbf{R}, \sigma$, and outputs a basis $\mathbf{S}$ of $\Lambda^{\perp}(\mathbf{U})$ for $\mathbf{U} = [\mathbf{A}|\mathbf{AR} + \mathbf{B}]$ with $\|\widetilde{\mathbf{S}}\| \leq \sigma \cdot \sqrt{2m}$ whose distribution depends on $\mathbf{U}, \sigma$.*

4.1 An Inner-Product Trapdoor Function from Lattices

In this subsection, we present a concrete inner-product trapdoor function from lattices. In our construction, each inversion key will be associated with a predicate vector $\mathbf{a} = (a_1, ..., a_\ell) \in \mathbb{Z}_q^\ell$ for some fixed $\ell \geq 2$ and each function value will be associated with an attribute vector $\mathbf{b} = (b_1, ..., b_\ell) \in \mathbb{Z}_q^\ell$. Inversion should succeed if and only if $\langle \mathbf{a}, \mathbf{b} \rangle = 0 \mod q$. Hence the predicate associated with the inversion key is defined as $f_{\mathbf{a}}(\mathbf{b}) = 1$ if $\langle \mathbf{a}, \mathbf{b} \rangle = 0 \mod q$, and $f_{\mathbf{a}}(\mathbf{b}) = 0$ otherwise. We note that we have to append the attribute vector $\mathbf{b}$ after the function value in order to invert it. This restriction limits our $\texttt{IPTDF}$ only enables *payload hiding* $\texttt{IPE}$ scheme (see Sec. 5), however, this will not harm the lossy attribute hiding property of lossy $\texttt{IPTDF}$s, since the public parameters will still hide the information of the lossy attribute.

Let $c > 1$ be an positive integer to be determined later. Let $n = \lambda$ be a security parameter and ℓ be the length of predicate and attribute vectors. Let $q = \text{poly}(n)$ be a prime, $b = b(n, \ell) \geq 2$ be an integer, $\hat{n} = cn$, and $m = \Theta(\hat{n} \log_b q)$. Let $r = r(n, \ell) \geq 2$ be an integer and define $k = \lfloor \log_r q \rfloor$. Define $D_\beta = \{0, 1, ..., \beta - 1\}$ and D_γ similarly for some positive integers $\beta \geq \gamma$ to be determined later. Let σ, α be positive real Gaussian parameters. Our inner-product trapdoor function has domain $\texttt{InSp}$=$D_\beta^{(\ell(k+1)+1)m+n} \times D_\gamma^{\hat{n}-n}$.

$\texttt{IPTDF.Pg}(n, \ell)$ takes as input a security parameter n and a parameter ℓ, denoting the length of predicate and attribute vectors,

1. Use the algorithm of Lemma 4 to generate a (nearly uniform) $\mathbf{A} \in \mathbb{Z}_q^{\hat{n} \times m}$, together with a basis $\mathbf{T_A}$ for lattice $\Lambda^{\perp}(\mathbf{A})$ such that $\|\widetilde{\mathbf{T_A}}\| = O(b \cdot \sqrt{\hat{n} \log_b q})$.
2. Choose $\ell \cdot (k+1)$ uniformly random matrices $\mathbf{A}_{i,j} \in \mathbb{Z}_q^{\hat{n} \times m}$ for $1 \le i \le \ell$ and $0 \le j \le k$. Choose a uniformly random matrix $\mathbf{B} \in \mathbb{Z}_q^{\hat{n} \times m}$.

Output $PP = (\mathbf{A}, \mathbf{B}, \{\mathbf{A}_{i,j}\}_{1 \le i \le \ell, 0 \le j \le k})$ and $msk = \mathbf{T_A}$.

`IPTDF.Kg`$(PP, \mathbf{a}, msk)$ takes as input public parameters PP, a predicate vector $\mathbf{a} = (a_1, ..., a_\ell) \in \mathbb{Z}_q^\ell$, and a master secret key msk,

1. For $i = 1, ..., \ell$, write the r-ary decomposition of $a_i \in \mathbb{Z}_q$ as

$$a_i = \sum_{j=0}^{k} a_{i,j} \cdot r^j, \quad \text{where} \quad a_{i,j} \in [0, ..., r-1].$$

2. Define the matrix $\mathbf{U_a} = [\mathbf{A} | \sum_{i=1}^{\ell} \sum_{j=0}^{k} a_{i,j} \mathbf{A}_{i,j}]$.
3. Use the `SampleLeft` algorithm in Lemma 6 to generate a basis $\mathbf{S_a}$ of $\Lambda^{\perp}(\mathbf{U_a})$ with $\|\widetilde{\mathbf{S_a}}\| \le \sigma\sqrt{2m}$.

Output the inversion key $sk_{\mathbf{a}} = \mathbf{S_a}$.

`IPTDF.Ev`$(PP, \mathbf{b}, \mathbf{m})$ takes as input a public parameters PP, an attribute vector $\mathbf{b} = (b_1, ..., b_\ell) \in \mathbb{Z}_q^\ell$, and a message $\mathbf{m} = [\mathbf{x}_0 | \mathbf{x}_{1,0} | ... | \mathbf{x}_{i,j} | ... | \mathbf{x}_{\ell,k} | \mathbf{x}] \in D_\beta^{(\ell(k+1)+1)m+n} \times D_\gamma^{\hat{n}-n}$, where $\mathbf{x}_0, \mathbf{x}_{i,j} \in D_\beta^m$ for $1 \le i \le \ell$, $0 \le j \le k$, and $\mathbf{x} \in D_\beta^n \times D_\gamma^{\hat{n}-n}$,

1. Define the matrix

$$\mathbf{F_b} = [\mathbf{A} | \mathbf{A}_{1,0} + r^0 b_1 \mathbf{B}| \cdots | \mathbf{A}_{i,j} + r^j b_i \mathbf{B}| \cdots | \mathbf{A}_{\ell,k} + r^k b_\ell \mathbf{B}].$$

2. Compute $C_{\mathbf{b}} = [\mathbf{I}_{(\ell(k+1)+1)m} | \mathbf{F}_{\mathbf{b}}^t] \cdot \mathbf{m} \mod q$.

Output $C_{\mathbf{b}}$ together with the attribute vector $\mathbf{b}$.

`IPTDF.Inv`$(PP, sk_{\mathbf{a}}, (C_{\mathbf{b}}, \mathbf{b}))$ takes as input public parameters PP, an inversion key $sk_{\mathbf{a}}$ for predicate $\mathbf{a}$, and a function value $(C_{\mathbf{b}}, \mathbf{b})$ for attribute vector $\mathbf{b}$,

1. Parse $C_{\mathbf{b}}$ into $\mathbf{c}_0$, $\mathbf{c}_{i,j}$ for $1 \le i \le \ell, 0 \le j \le k$, where $\mathbf{c}_0 = \mathbf{A}^t \mathbf{x} + \mathbf{x}_0$, $\mathbf{c}_{i,j} = (\mathbf{A}_{i,j} + r^j b_i \mathbf{B})^t \mathbf{x} + \mathbf{x}_{i,j}$.
2. Compute $\widetilde{\mathbf{c}} = \sum_{i=1}^{\ell} \sum_{j=0}^{k} a_{i,j} \mathbf{c}_{i,j} = \sum_{i=1}^{\ell} \sum_{j=0}^{k} a_{i,j} \mathbf{A}_{i,j}^t \mathbf{x} + \langle \mathbf{a}, \mathbf{b} \rangle \mathbf{B}^t \mathbf{x} + \sum_{i=1}^{\ell} \sum_{j=0}^{k} a_{i,j} \mathbf{x}_{i,j}$.
3. Note that $[\mathbf{c}_0 | \widetilde{\mathbf{c}}] = [\mathbf{A} | \sum_{i=1}^{\ell} \sum_{j=0}^{k} a_{i,j} \mathbf{A}_{i,j} + \langle \mathbf{a}, \mathbf{b} \rangle \mathbf{B}]^t \mathbf{x} + [\mathbf{x}_0 | \sum_{i=1}^{\ell} \sum_{j=0}^{k} a_{i,j} \mathbf{x}_{i,j}]$. If $\langle \mathbf{a}, \mathbf{b} \rangle = 0 \mod q$, use $sk_{\mathbf{a}}$ and the inversion algorithm of Lemma 5 to compute $[\widetilde{\mathbf{x}} | \mathbf{x}]$ from $[\mathbf{c}_0 | \widetilde{\mathbf{c}}]$, where $\widetilde{\mathbf{x}} = [\mathbf{x}_0 | \sum_{i=1}^{\ell} \sum_{j=0}^{k} a_{i,j} \mathbf{x}_{i,j}]$. Then recover all $\mathbf{x}_{i,j}$ from $\mathbf{c}_{i,j}$ by using $\mathbf{x}$ and the attribute vector $\mathbf{b}$. Finally, It outputs $\mathbf{m}$ if $\langle \mathbf{a}, \mathbf{b} \rangle = 0 \mod q$.

We now describe the sibling L-IPTDF. The evaluation and inversion algorithms are those in IPTDF. We give the parameter generation and inversion key generation algorithms of L-IPTDF as follows.

L-IPTDF.Pg$(n, \ell, \mathbf{b}^*)$ takes as input a security parameter n, a parameter ℓ denoting the length of predicate and attribute vectors, and an attribute vector $\mathbf{b}^* = (b_1^*, ..., b_\ell^*) \in \mathbb{Z}_q^\ell$,

1. Use the algorithm of Lemma 4 to generate a (nearly uniform) $\mathbf{B} \in \mathbb{Z}_q^{\hat{n} \times m}$, together with a basis $\mathbf{T_B}$ for lattice $\Lambda^\perp(\mathbf{B})$ such that $\|\widetilde{\mathbf{T_B}}\| = O(b \cdot \sqrt{\hat{n} \log_b q})$.
2. Choose a uniformly random matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$, and $\mathbf{E} \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}^{(m+n)\times(\hat{n}-n)}$ where $\alpha q = \Theta(\sqrt{n})$, and pairwise independent $\mathbf{R}_{i,j} \leftarrow \mathcal{D}_{\mathbb{Z},b\cdot\omega(\sqrt{\log n})}^{m \times m}$ for $1 \le i \le \ell, 0 \le j \le k$.
3. Set $\mathbf{A}^t = \left[\bar{\mathbf{A}}^t \middle| [\mathbf{I}_m | \bar{\mathbf{A}}^t] \cdot \mathbf{E} \mod q\right]$, and let $\mathbf{A}_{i,j} = \mathbf{A}\mathbf{R}_{i,j} - r^j b_i^* \mathbf{B}$ for $1 \le i \le \ell, 0 \le j \le k$.

Output $PP = (\mathbf{A}, \mathbf{B}, \{\mathbf{A}_{i,j}\}_{1 \le i \le \ell, 0 \le j \le k})$ and $msk = (\mathbf{T_B}, \{\mathbf{R}_{i,j}\}_{1 \le i \le \ell, 0 \le j \le k})$.

L-IPTDF.Kg(PP, f, msk) takes as input public parameters PP, a master secret key msk, and a predicate vector $\mathbf{a} = (a_1, ..., a_\ell) \in \mathbb{Z}_q^\ell$,

1. Define the matrix

$$\mathbf{U_a} = [\mathbf{A} | \sum_{i=1}^{\ell} \sum_{j=0}^{k} a_{i,j} \mathbf{A}_{i,j}] = [\mathbf{A} | \mathbf{A}(\sum_{i=1}^{\ell} \sum_{j=0}^{k} a_{i,j} \mathbf{R}_{i,j}) - \langle \mathbf{a}, \mathbf{b}^* \rangle \mathbf{B}].$$

2. If $\langle \mathbf{a}, \mathbf{b}^* \rangle \neq 0 \mod q$, use the SampleRight algorithm in Lemma 7 to generate a basis $\mathbf{S_a}$ of $\Lambda^\perp(\mathbf{U_a})$ with $\|\widetilde{\mathbf{S_a}}\| \le \sigma\sqrt{2m}$, else abort.

4.2 Correctness

We now show that for certain parameter choices, the inversion algrithms of IPTDF and L-IPTDF work correctly with overwhelming probability, and the evaluation of L-IPTDF.Ev$(PP, \mathbf{b}^*, \cdot)$ is lossy.

Lemma 8. *Suppose the parameters γ, β, b satisfy*

$$\gamma^{c-1} \ge 2^{\Omega((\ell(k+1)+1)m/n)}, \text{ and } \gamma \cdot \widetilde{\Omega}(bm\sqrt{n}) \le \beta \le \frac{q}{2\sqrt{2}\sigma m((r-1)\ell(k+1)+1)}.$$

We have:

1. *If $\langle \mathbf{a}, \mathbf{b} \rangle = 0 \mod q$, then with overwhelming probability* IPTDF.Inv *invert correctly, and* L-IPTDF.Inv *invert correctly with $\mathbf{b} \neq \mathbf{b}^*$.*
2. L-IPTDF.Ev$(PP, \mathbf{b}^*, \cdot)$ *is a lossy function with lossiness $\Omega((\ell(k+1)+1)m)$.*

Proof. During the second step of $\texttt{IPTDF.Inv}(PP, \mathbf{S_a}, (C_\mathbf{b}, \mathbf{b}))$, we compute $\widetilde{\mathbf{c}}$, which is equal to

$$\widetilde{\mathbf{c}} = \sum_{i=1}^{\ell}\sum_{j=0}^{k} a_{i,j}\mathbf{c}_{i,j} = \sum_{i=1}^{\ell}\sum_{j=0}^{k} a_{i,j}\mathbf{A}_{i,j}^t\mathbf{x} + \langle \mathbf{a}, \mathbf{b}\rangle \mathbf{B}^t\mathbf{x} + \sum_{i=1}^{\ell}\sum_{j=0}^{k} a_{i,j}\mathbf{x}_{i,j}.$$

If $\langle \mathbf{a}, \mathbf{b}\rangle = 0 \mod q$, then the middle term disappears, leaving $\widetilde{\mathbf{c}} = \sum_{i=1}^{\ell}\sum_{j=0}^{k} a_{i,j}\mathbf{A}_{i,j}^t\mathbf{x} + \sum_{i=1}^{\ell}\sum_{j=0}^{k} a_{i,j}\mathbf{x}_{i,j}$. It follows that

$$[\mathbf{c}_0|\widetilde{\mathbf{c}}] = [\mathbf{A}|\sum_{i=1}^{\ell}\sum_{j=0}^{k} a_{i,j}\mathbf{A}_{i,j}]^t \cdot \mathbf{x} + \widetilde{\mathbf{x}} = [\mathbf{I}_{2m}|\mathbf{U}_\mathbf{a}^t] \cdot [\widetilde{\mathbf{x}}|\mathbf{x}] \mod q,$$

where $\widetilde{\mathbf{x}} = [\mathbf{x}_0|\sum_{i=1}^{\ell}\sum_{j=0}^{k} a_{i,j}\mathbf{x}_{i,j}]$. Since $\mathbf{S_a}$ is a basis of $\Lambda^\perp(\mathbf{U_a})$, according to Lemma 5, if $\|\widetilde{\mathbf{x}}\| \le q/(2\|\widetilde{\mathbf{S_a}}\|)$, one can recover $[\widetilde{\mathbf{x}}|\mathbf{x}]$ by using $\mathbf{S_a}$. From the $\mathbf{c}_{i,j}$, the attribute vector $\mathbf{b}$ and $\mathbf{x}$, one can obtain $\mathbf{x}_{i,j}$ for $1 \le i \le \ell, 0 \le j \le k$. By Lemma 6 and 7 we have $\|\widetilde{\mathbf{S_a}}\| \le \sigma\sqrt{2m}$ with overwhelming probability. Since $\mathbf{m} \in D_\beta^{(\ell(k+1)+1)m+n} \times D_\gamma^{\hat{n}-n}$, and $a_{i,j} \in \{0, ..., r-1\}$, by the triangle inequality, we have

$$\begin{aligned}\|\widetilde{\mathbf{x}}\| \le \|\mathbf{x}_0\| + \|\sum_{i=1}^{\ell}\sum_{j=0}^{k} a_{i,j}\mathbf{x}_{i,j}\| &\le \beta\sqrt{m} + (r-1)\ell(k+1)\beta\sqrt{m} \\ &= (1 + (r-1)\ell(k+1))\beta\sqrt{m}.\end{aligned}$$

For β as in the lemma statement, $\beta \le \frac{q}{2\sqrt{2}\sigma m((r-1)\ell(k+1)+1)}$ is sufficient to recover $\mathbf{m}$, as desired.

In the third step of $\texttt{L-IPTDF.Pg}$, $\mathbf{A}^t = \left[\bar{\mathbf{A}}^t \middle| [\mathbf{I}_m|\bar{\mathbf{A}}^t] \cdot \mathbf{E} \mod q\right]$ and $\mathbf{A}_{i,j} = \mathbf{A}\mathbf{R}_{i,j} - r^j b_i^* \mathbf{B}$ for $1 \le i \le \ell, 0 \le j \le k$, then in $\texttt{L-IPTDF.Ev}(PP, \mathbf{b}^*, \mathbf{m})$, we compute

$$\begin{aligned}\mathbf{F}_{\mathbf{b}^*} &= [\mathbf{A}|\mathbf{A}_{1,0} + r^0 b_1^*\mathbf{B}|\cdots|\mathbf{A}_{i,j} + r^j b_i^*\mathbf{B}|\cdots|\mathbf{A}_{\ell,k} + r^k b_\ell^*\mathbf{B}] \\ &= [\mathbf{A}|\mathbf{A}\mathbf{R}_{1,0}|\cdots|\mathbf{A}\mathbf{R}_{i,j}|\cdots|\mathbf{A}\mathbf{R}_{\ell,k}] = \mathbf{A}[\mathbf{I}_m|\mathbf{R}_{1,0}|\cdots|\mathbf{R}_{i,j}|\cdots|\mathbf{R}_{\ell,k}].\end{aligned}$$

Let $\mathbf{R} = [\mathbf{I}_m|\mathbf{R}_{1,0}|\cdots|\mathbf{R}_{i,j}|\cdots|\mathbf{R}_{\ell,k}]$, therefore,

$$\begin{aligned}\mathbf{F}_{\mathbf{b}^*}^t &= \mathbf{R}^t \cdot \left[\bar{\mathbf{A}}^t \middle| [\mathbf{I}_m|\bar{\mathbf{A}}^t] \cdot \mathbf{E}\right] = \left[(\bar{\mathbf{A}}\mathbf{R})^t \middle| [\mathbf{R}^t|(\bar{\mathbf{A}}\mathbf{R})^t] \cdot \mathbf{E}\right] \\ &= \left[(\bar{\mathbf{A}}\mathbf{R})^t \middle| [\mathbf{I}_{(\ell(k+1)+1)m}|(\bar{\mathbf{A}}\mathbf{R})^t] \begin{bmatrix}\mathbf{R}^t & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n\end{bmatrix} \cdot \mathbf{E}\right] \\ &= \left[(\bar{\mathbf{A}}\mathbf{R})^t \middle| [\mathbf{I}_{(\ell(k+1)+1)m}|(\bar{\mathbf{A}}\mathbf{R})^t] \cdot \mathbf{E}'\right],\end{aligned}$$

where, $\mathbf{E}' = \begin{bmatrix}\mathbf{R}^t & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n\end{bmatrix} \mathbf{E}$. Note that

$$s_1(\mathbf{E}') \le s_1(\mathbf{R}^t)s_1(\mathbf{E}) \le \widetilde{O}(b\sqrt{m}) \cdot O(\sqrt{mn}) \le \widetilde{O}(bm\sqrt{n}).$$

We have the following,

$$\begin{aligned} C_{\mathbf{b}^*} &= [\mathbf{I}_{(\ell(k+1)+1)m}|\mathbf{F}^t_{\mathbf{b}^*}] \cdot \mathbf{m} \\ &= [\mathbf{I}_{(\ell(k+1)+1)m}|(\bar{\mathbf{A}}\mathbf{R})^t] \cdot ([\mathbf{I}_{(\ell(k+1)+1)m+n}|\mathbf{E}'] \cdot \mathbf{m}) \mod q. \end{aligned}$$

Therefore, it suffices to bound the number of possible values of the form

$$[\mathbf{I}_{(\ell(k+1)+1)m+n}|\mathbf{E}'] \cdot \mathbf{m} \mod q.$$

Define $N_d(s)$ to be the number of integer points in an d-dimensional ball of radius s. For $r \geq \sqrt{d}$, from the volume of the ball and Stirling's approximation, we have $N_d(s) = O(s/\sqrt{d})^d$. Therefore, the number of possible values of $[\mathbf{I}_{(\ell(k+1)+1)m+n}|\mathbf{E}'] \cdot \mathbf{m}$ is at most $N_{(\ell(k+1)+1)m+n}(\|[\mathbf{I}_{(\ell(k+1)+1)m+n}|\mathbf{E}'] \cdot \mathbf{m}\|)$. Since $\mathbf{m} = [\mathbf{x}_0|\mathbf{x}_{1,0}|...|\mathbf{x}_{i,j}|...|\mathbf{x}_{\ell,k}|\mathbf{x}] \in D_\beta^{(\ell(k+1)+1)m+n} \times D_\gamma^{\hat{n}-n}$, we have

$$\begin{aligned} \|[\mathbf{I}_{(\ell(k+1)+1)m+n}|\mathbf{E}'] \cdot \mathbf{m}\| &\leq \beta \cdot \sqrt{(\ell(k+1)+1)m+n} + s_1(\mathbf{E}') \cdot \gamma\sqrt{\hat{n}-n} \\ &\leq \sqrt{(\ell(k+1)+1)m+n} \cdot (\beta + \gamma \cdot s_1(\mathbf{E}')). \end{aligned}$$

Therefore, the number of possible values of $[\mathbf{I}_{(\ell(k+1)+1)m+n}|\mathbf{E}'] \cdot \mathbf{m}$ is at most $O(\beta + \gamma \cdot s_1(\mathbf{E}'))^{(\ell(k+1)+1)m+n}$. For lossiness, observe that the base-2 logarithm of the domain size of $\texttt{L-IPTDF.Ev}(PP, \mathbf{b}^*, \cdot)$ is

$$((\ell(k+1)+1)m+n)\log\beta + (c-1)n\log\gamma.$$

Whereas by the above, and for $\beta \geq \gamma \cdot s_1(\mathbf{E}')$, the base-2 logarithm of the image size of $\texttt{L-IPTDF.Ev}(PP, \mathbf{b}^*, \cdot)$ is at most

$$\begin{aligned} &((\ell(k+1)+1)m+n)\log(O(\beta + \gamma \cdot s_1(\mathbf{E}'))) \\ &\qquad \leq ((\ell(k+1)+1)m+n)\log\beta + O((\ell(k+1)+1)m). \end{aligned}$$

Let $\gamma^{c-1} \geq 2^{\Omega((\ell(k+1)+1)m/n)}$, and for sufficient large constant in $\Omega(\cdot)$, the two quantities above differ by at least $\Omega((\ell(k+1)+1)m)$ as desired. □

4.3 Security

In this subsection, we show that the inner-product function described above is $\Omega((\ell(k+1)+1)m)$-lossy under selective attribute attacker.

Theorem 1. *Suppose β, γ as in Lemma 8. If decisional LWE problem is infeasible with error rate α, then* IPTDF *described above is $\Omega((\ell(k+1)+1)m)$-lossy with sibling* L-IPTDF *described above, under selective attribute adversaries.*

Proof. We define a series of games (Game_0,...,Game_3) where in Game_0, an adversary $\mathcal{A}$ is against IPTDF, that is, the public parameter generation, key generation algorithms are from IPTDF. While in Game_3, $\mathcal{A}$ is against L-IPTDF, that is, the parameter generation, key generation algorithms are from L-IPTDF. We show that the adversary's views in the first game and the last game are indistinguishable.

Game$_0$: $\mathcal{A}$ submits a challenge attribute vector $\mathbf{b}^* = (b_1^*, ..., b_\ell^*)$ before setup. The challenger uses the algorithm of Lemma 4 to obtain $(\mathbf{A}, \mathbf{T_A}) \in \mathbb{Z}_q^{\hat{n} \times m} \times \mathbb{Z}^{m \times m}$, and chooses uniformly random matrices $\mathbf{B}, \mathbf{A}_{i,j} \in \mathbb{Z}_q^{\hat{n} \times m}$ for $1 \leq i \leq \ell, 0 \leq j \leq k$. The challenger gives $(\mathbf{A}, \mathbf{B}, \{\mathbf{A}_{i,j}\}_{1 \leq i \leq \ell, 0 \leq j \leq k})$ to $\mathcal{A}$, and keeps $\mathbf{T_A}$ as the master secret key. When $\mathcal{A}$ queries an inversion key for a predicate vector $\mathbf{a}$ with $\langle \mathbf{a}, \mathbf{b}^* \rangle \neq 0 \mod q$, the challenger uses $\mathbf{T_A}$ to respond an inversion key $\mathbf{S_a}$ by invoking algorithm `SampleLeft` from Lemma 6, the distribution of $\mathbf{S_a}$ depends on $\mathbf{U_a}$ and σ .

Game$_1$: This game is identical to Game$_0$ except that, the challenger changes the way to generate $\mathbf{A}_{i,j}$ for $1 \leq i \leq \ell, 0 \leq j \leq k$. Instead, the challenger first chooses pairwise independent $\mathbf{R}_{i,j} \in \mathcal{D}^{m \times m}_{\mathbb{Z}, b \cdot \omega(\sqrt{\log n})}$ for $1 \leq i \leq \ell, 0 \leq j \leq k$. Let $\mathbf{A}_{i,j} = \mathbf{A}\mathbf{R}_{i,j} - r^j b_i^* \mathbf{B}$.

Game$_2$: This game is identical to Game$_1$ except that the challenger changes the way to generate the master secret key and respond the key-extraction query. Instead, the challenger uses the algorithm of Lemma 4 to obtain $(\mathbf{B}, \mathbf{T_B})$, and chooses $\mathbf{A} \leftarrow \mathbb{Z}_q^{\hat{n} \times m}$ uniformly at random. When $\mathcal{A}$ queries an inversion key for a predicate vector $\mathbf{a}$ with $\langle \mathbf{a}, \mathbf{b}^* \rangle \neq 0 \mod q$, the challenger uses $\mathbf{T_B}$ and $\mathbf{R}_{i,j}$ for $1 \leq i \leq \ell, 0 \leq j \leq k$ to respond an inversion key $\mathbf{S_a}$ by invoking algorithm `SampleRight` from Lemma 7, the distribution of $\mathbf{S_a}$ depends on $\mathbf{U_a}$ and σ.

Game$_3$: This game is identical to Game$_2$ except that the challenger changes the way to generate $\mathbf{A}$. Instead, the challenger chooses a uniformly random matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$, and $\mathbf{E} \leftarrow \mathcal{D}^{(m+n) \times (\hat{n}-n)}_{\mathbb{Z}, \alpha q}$ where $\alpha q = \Theta(\sqrt{n})$. Set $\mathbf{A}^t = \left[\bar{\mathbf{A}}^t \middle| [\mathbf{I}_m | \bar{\mathbf{A}}^t] \cdot \mathbf{E} \mod q \right]$.

It's obvious that Game$_0$ is the `IPTDF` security definition and Game$_3$ is the `L-IPTDF` security definition. We now show that the adversary's views between the adjacent games are indistinguishable.

The only difference of Game$_0$ of Game$_1$ is the way $\mathbf{A}_{i,j}$ generated. In Game$_1$, $\mathbf{A}$ is uniform, therefore $\mathbf{A}, \mathbf{A}\mathbf{R}_{1,0}, ..., \mathbf{A}\mathbf{R}_{\ell,k}$ is statistically close to a uniform string of the same size by Lemma 3, and so is $\mathbf{A}, \mathbf{A}\mathbf{R}_{1,0} - r^0 b_1^* \mathbf{B}, ..., \mathbf{A}\mathbf{R}_{\ell,k} - r^k b_\ell^* \mathbf{B}$. Thus, the adversary's views between Game$_0$ and Game$_1$ are statistically indistinguishable.

The differences of Game$_1$ and Game$_2$ are the way $\mathbf{A}, \mathbf{B}$ generated and the way to answer key-extraction queries. By Lemma 4, the distributions of $\mathbf{A}, \mathbf{B}$ in Game$_1$ and Game$_2$ are statistically close. Therefore, the distributions of $\mathbf{U_a}$ in Game$_1$ and Game$_2$ are statistically close. In Game$_2$, note that $\mathbf{U_a} = [\mathbf{A} | \mathbf{A}(\sum_{i=1}^{\ell} \sum_{j=0}^{k} a_{i,j} \mathbf{R}_{i,j}) - \langle \mathbf{a}, \mathbf{b}^* \rangle \mathbf{B}]$, it can invoke the algorithm `SampleRight` by Lemma 7 as long as $\langle \mathbf{a}, \mathbf{b}^* \rangle \neq 0 \mod q$. By Lemma 6 and Lemma 7, for sufficiently large σ, the distribution of $\mathbf{S_a}$ in Game$_1$ and Game$_2$ are statistically close.

Apparently, matrix $\mathbf{A}$ in Game_2 and Game_3 is computational indistinguishable if the decisional LWE problem is infeasible. Summarize the above discussions, we complete the proof □

4.4 Parameter Selection

We can extract from the above description the parameters required for correctness and security of the system. By Lemma 8 we require

$$\gamma^{c-1} \geq 2^{\Omega((\ell(k+1)+1)m/n)}, \text{ and } \gamma \cdot \widetilde{\Omega}(bm\sqrt{n}) \leq \beta \leq \frac{q}{2\sqrt{2}\sigma m((r-1)\ell(k+1)+1)}.$$

For security, we require $\alpha q = \Theta(\sqrt{n})$. The constants c and γ depend on the relationship of ℓ, m, and n. We need $\gamma^{c-1} \geq 2^{\Omega((\ell(k+1)+1)m/n)}$. In order to generate $\mathbf{A}$ with a trapdoor, we have $m = \Theta(\hat{n} \log_b q)$, so we need $\gamma > q^{\Theta((\ell(k+1)+1)/\log b)\cdot c/c-1}$. For any desired constant $C > 1$ and $\ell = O(\log n)$, we can choose constants $k > 1, c > 1$ and choose $b = \Theta(n)$ such that $\gamma \leq q^{1/C}$. The additional constraints imposed by our security reduction are as follows. From the description of `IPTDF.Pg`, we have $\|\widetilde{\mathbf{T}_\mathbf{A}}\| = O(b\sqrt{\hat{n}\log_b q})$ by Lemma 4, in order to respond the key-extraction queries by `SampleLeft`, σ subjects to the requirement that

$$\sigma \geq \|\widetilde{\mathbf{T}_\mathbf{A}}\| \cdot \omega(\sqrt{\log m}) = O(b\sqrt{\hat{n}\log_b q}) \cdot \omega(\sqrt{\log m}).$$

From the description of `L-IPTDF.Pg`, we have $\|\widetilde{\mathbf{T}_\mathbf{B}}\| = O(b\sqrt{\hat{n}\log_b q})$ by Lemma 4, in order to respond the key-extraction queries by `SampleRight`, σ subjects to the requirement that

$$\sigma \geq \|\widetilde{\mathbf{T}_\mathbf{B}}\| \cdot s_1(\sum_{i=1}^{\ell}\sum_{j=0}^{k} a_{i,j}\mathbf{R}_{i,j}) \cdot \omega(\sqrt{\log m}).$$

Since $\mathbf{R}_{i,j}$ are chosen from $\mathcal{D}^{m\times m}_{\mathbb{Z},b\omega(\sqrt{\log n})}$, and $a_{i,j} \in \{0, ..., r-1\}$, it follows that $s_1(\sum_{i=1}^{\ell}\sum_{j=0}^{k} a_{i,j}\mathbf{R}_{i,j}) \leq \widetilde{O}((r-1)\ell(k+1)\cdot b\sqrt{m})$ with overwhelming probability. We see that it suffices to choose

$$\sigma \geq O(b\sqrt{\hat{n}\log_b q}) \cdot \widetilde{O}((r-1)\ell(k+1)b\sqrt{m}) \cdot \omega(\sqrt{\log m}).$$

To satisfy the more stringent of the above two conditions, we set $\sigma = \Theta(r\ell k n^{3.5})$. For correctness and lossiness, it suffices to take

$$q^{1/C} \cdot \widetilde{\Omega}(bm\sqrt{n}) \leq \beta \leq q/(2\sqrt{2}\sigma m(r-1)(\ell(k+1)+1)).$$

In order to satisfy all the constraints, it is sufficient to set $r = q^{1/C'}$ for some constant $C' > 1$ (therefore $k = C'$ is a constant), and sufficiently large q such that

$$q^{1-1/C-2/C'} \geq \widetilde{\Omega}(m^2 n^5 \ell^2 k^2) = \widetilde{\Omega}(n^7 \ell^2 k^2).$$

The following selection of parameters satisfies all of these constrains. For a given $\ell = O(\log n)$ and constant C, C', set

$$\begin{array}{ccc} m = 20cn, & \beta = n^{3+\delta}, & \gamma = n^{\delta}, \\ \sigma = \lceil r\ell kn^{3.5} \rceil, & \alpha = (n^{7+\delta}\ell^2 k^2)^{-1}, & q = \text{the prime nearest to } \lceil n^{7.5+\delta}\ell^2 k^2 \rceil, \end{array}$$

where $n^{\delta} = \lceil q^{1/C+2/C'} \rceil$, and constant c is set as in the analysis, and $b = n$. Observe that the above setting of parameters satisfies all the constrains, the security of the scheme can be based on the hardness of approximating `SIVP` and `GapSvp` to within a factor of $\widetilde{O}(n/\alpha) = \widetilde{O}(n^{8+\delta}\ell^2 k^2)$ in the worst case by quantum algorithms.

5 Applications

In this section, we describe some applications of our `IPTDF`. These applications include chosen-plaintext secure `IPE` schemes and chosen-ciphertext secure `IPE` scheme. Katz, Sahai, and Waters [19] introduce two basic security notions of `IPE`: *payload hiding* and *attribute hiding*. Payload hiding guarantees that no efficient adversary can obtain any information about the encrypted message, but allows information about attributes to be revealed. Attribute hiding is a stronger notion which guarantees in addition that no efficient adversary can obtain any information about the attribute associated with a ciphertext.

Chosen-Plaintext Secure Inner-Product Encryption. A straightforward application of `IPTDF` is for inner-product encryption (`IPE`). By the lossiness of `IPTDF`s, it is easy for us to obtain a payload hiding `IPE` scheme (via hardcore bits) under selectively chosen-plaintext adversaries. However, as mentioned in Sec. 4.1 we have to append the attribute vector after the function value, therefore, anyone can learn the information of the attribute from the function value. In this case we can not achieve attribute hiding `IPE` schemes using our `IPTDF`, we leave it as a future work to construct `IPTDF`s whose attribute information is hidden in the function value.

Due to its lossiness, our `IPTDF` together with a pairwise independent hash function h imply an `IPE` scheme for multi-bit messages (with length $O(\ell n)$). The ciphertext of the `IPE` scheme consists of $c = (\texttt{IPTDF.Ev}(PP, \mathbf{b}, x), h(x) \oplus m)$, where x is randomly chosen from the domain of `IPTDF` and h, and m is the message. Our concrete construction of `IPTDF` is inspired by [2], then the efficiency of our `IPE` scheme is almost the same as the one in [2] except that our scheme can encrypt multi-bit messages simultaneously. However, our scheme only supports attribute vectors with logarithmic length, while the scheme in [2] supports attribute vectors with polynomial length.

Chosen-Ciphertext Secure Inner-Product Encryption. Peikert and Waters [27] gave a framework to construct chosen-ciphertext secure public key encryption schemes from lossy trapdoor functions. Our inner-product lossy trapdoor function also works in this framework. Following the framework in [27], to

obtain a payload hiding IPE scheme under selectively chosen-ciphertext adversaries, one can combine an IPTDF and an All-But-One (ABO) [27] trapdoor function with a strongly unforgeable one-time signature. The ciphertext of the IPE scheme consists of $c = (vk, \texttt{IPTDF.Ev}(PP, \mathbf{b}, x), \texttt{ABO-TDF.Ev}(vk, x), h(x) \oplus m, \sigma)$, where x is randomly chosen from the domain of IPTDF and ABO-TDF, h is a pairwise independent hash function, m is the message, and σ is the one-time signature of $\texttt{IPTDF.Ev}(PP, \mathbf{b}, x), \texttt{ABO-TDF.Ev}(vk, x)$, and $h(x) \oplus m$ under the signing key associated to vk.

In order to base the resulting IPE scheme on lattices, we need to provide an ABO trapdoor function based on lattices.[2] We have two ways to address this problem. The first one is to use the original lattice based ABO trapdoor function presented in [27]. However, this construction brings large public key size. We prefer to the second one, and as a by-product, we give a generic construction for ABO trapdoor functions. We observe that the lossy sibling of IBTDF is actually an ABO lossy trapdoor function if we view an identity as a *branch*. The lossy identity hiding property is exactly the hidden branch property of ABO trapdoor functions (even the adversary can access an oracle to obtain other inversion keys). Bellare et al. presented a "direct" construction of IBTDF from lattices. Use this ABO trapdoor function and our IPTDF scheme, we obtain the first chosen-ciphertext secure IPE scheme based on lattices.

Note that a generic method to construct chosen-ciphertext secure IPE scheme is the CHK/BK [15,12] transform. The CHK/BK transform transforms a 2-level chosen-plaintext secure hierarchical IPE scheme into chosen-ciphertext secure IPE scheme. However, the only lattice based IPE scheme in [2] seems difficult to be extended into a hierarchical IPE scheme. Therefore, it seems difficult to obtain a chosen-ciphertext secure IPE scheme from the scheme in [2] using CHK/BK transform.

Acknowledgements. Xiang Xie would like to thank Chris Peikert for helpful discussions. The authors would like to thank the anonymous reviewers for their useful comments.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
2. Agrawal, S., Freeman, D.M., Vaikuntanathan, V.: Functional Encryption for Inner Product Predicates from Learning with Errors. In: Lee, D.H. (ed.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 21–40. Springer, Heidelberg (2011)
3. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. Theory of Computing Systems, 1–19 (2011)
4. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)

[2] It is easy to get strongly unforgeable one-time signatures from lattices.

5. Bellare, M., Brakerski, Z., Naor, M., Ristenpart, T., Segev, G., Shacham, H., Yilek, S.: Hedged Public-Key Encryption: How to Protect against Bad Randomness. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 232–249. Springer, Heidelberg (2009)
6. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and Impossibility Results for Encryption and Commitment Secure under Selective Opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009)
7. Bellare, M., Kiltz, E., Peikert, C., Waters, B.: Identity-Based (Lossy) Trapdoor Functions and Applications. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 228–245. Springer, Heidelberg (2012)
8. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, SP 2007, pp. 321–334. IEEE (2007)
9. Boldyreva, A., Fehr, S., O'Neill, A.: On Notions of Security for Deterministic Encryption, and Efficient Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)
10. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
11. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. SIAM J. of Computing 32(3), 586–615 (2003)
12. Boneh, D., Katz, J.: Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
13. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. Theory of Cryptography, 253–273 (2011)
14. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. Theory of Cryptography, 535–554 (2007)
15. Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
16. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
17. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 197–206. ACM (2008)
18. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 89–98. ACM (2006)
19. Katz, J., Sahai, A., Waters, B.: Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
20. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
21. Micciancio, D., Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)

22. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM Journal on Computing 37(1), 267 (2007)
23. Okamoto, T., Takashima, K.: Hierarchical Predicate Encryption for Inner-Products. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 214–231. Springer, Heidelberg (2009)
24. Okamoto, T., Takashima, K.: Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010)
25. Okamoto, T., Takashima, K.: Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 591–608. Springer, Heidelberg (2012)
26. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp. 333–342. ACM (2009)
27. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 187–196. ACM (2008)
28. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, pp. 84–93. ACM (2005)
29. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
30. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
31. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)