Bi-Deniable Inner Product Encryption from LWE

Daniel Apon [*] Xiong Fan [†] Feng-Hao Liu [‡]

Abstract

Deniable encryption (Canetti et al. CRYPTO '97) is an intriguing primitive that provides a security guarantee against not only eavesdropping attacks as required by semantic security, but also stronger coercion attacks performed after the fact. The concept of deniability has later demonstrated useful and powerful in many other contexts, such as leakage resilience, adaptive security of protocols, and security against selective opening attacks. Despite its conceptual usefulness, our understanding of how to construct deniable primitives under standard assumptions is restricted. In particular, from standard assumptions such as Learning with Errors (LWE), we have only multi-distributional or non-negligible advantage deniable encryption schemes, whereas with the much stronger assumption of indistinguishable obfuscation, we can obtain at least fully-secure sender-deniable PKE and computation. How to achieve deniability for other more advanced encryption schemes under standard assumptions remains an interesting open question.

In this work, we construct a bi-deniable inner product encryption (IPE) in the multi-distributional model without relying on obfuscation as a black box. Our techniques involve new ways of manipulating Gaussian noise, and lead to a significantly tighter analysis of noise growth in Dual Regev type encryption schemes. We hope these ideas can give insight into achieving deniability and related properties for further, advanced cryptographic constructions under standard assumptions.

1 Introduction

Deniable encryption, introduced by Canetti et al. [CDNO97] at CRYPTO 1997, is an intriguing primitive that allows Alice to privately communicate with Bob in a way that resists not only eavesdropping attacks as required by semantic security, but also stronger *coercion attacks* performed after the fact. An eavesdropper Eve stages a cocercion attack by additionally approaching Alice (or Bob, or both) *after* a ciphertext is transmitted and demanding to see all secret information: the plaintext, the random coins used by Alice for encryption, and any private keys held by Bob (or Alice) related to the ciphertext. In particular, Eve can use this information to "fully unroll" the *exact transcript* of some deterministic decryption procedure purportedly computed by Bob, as well as verify that the exact coins and decrypted plaintext in fact produce the coerced ciphertext. A secure deniable encryption scheme should maintain privacy of the sensitive data originally communicated between Alice and Bob under the coerced ciphertext (instead substituting a benign yet *convincing* plaintext in the view of Eve), even in the face of such a revealing attack and even if Alice and Bob may not interact during the coercion phase.

Historically, deniable encryption schemes have been challenging to construct. Under standard assumptions, Canetti et al. [CDNO97] constructed a sender-deniable[1] PKE where the distinguishing advantage between real and fake openings is an inverse polynomial depending on the public key size. But it was not

[*]University of Maryland, `dapon@cs.umd.edu`.

[†]Cornell University, `xfan@cs.cornell.edu`.

[‡]Florida Atlantic University, `fenghao.liu@fau.edu`.

[1]We differentiate between sender-deniable, receiver-deniable, and bi-deniable schemes. A bi-deniable scheme is both sender- and receiver-deniable.

until 2011 that Bendlin et al. [BNNO11] showed this limitation is inherent: any non-interactive public-key encryption scheme may be receiver-deniable (resp. bi-deniable) only with *non-negligible* $\Omega(1/\mathsf{size}(\mathsf{pk}))$ distinguishing advantage in the deniability experiment.

Following this, O'Neill, Peikert, and Waters [OPW11] proposed the first constructions of bi-deniable PKE with *negligible* deniability distinguishing advantage: from simulatable PKE generically, as well as from Learning with Errors (LWE [Reg05]) directly. In order to bypass the impossibility result of [BNNO11], O'Neill et al. chose to work in the so-called *multi-distributional* model. In the multi-distributional model of deniability, private keys sk are distributed by a central key authority. In the event that Bob is coerced to reveal a key sk that decrypts chosen ciphertext $\boldsymbol{c}^*$, the key authority distributes a *faking key* fk to Bob, which Bob can use to generate a fake key sk^* (designed to behave identically to sk except on ciphertext $\boldsymbol{c}^*$). If this step is allowed, then O'Neill et al. demonstrate that for their constructions, Eve has at most negligible advantage in distinguishing whether Bob revealed an honest sk or fake sk^*.

A major breakthrough in deniable encryption arrived with the work of Sahai and Waters [SW14], who proposed the first sender-deniable PKE with negligible distinguishing advantage from indistinguishability obfuscation ($i\mathcal{O}$) for P/poly [GGH$^+$13]. The concept of deniability has been demonstrated useful in the contexts of leakage resilience [DLZ15], adaptive security for protocols, and as well as deniable computation (or algorithms) [CGP15, DKR15, GP15]. In addition to coercion resistance, a bi-deniable encryption scheme is a non-committing encryption scheme [CFGN96], as well as a scheme secure under selective opening (SOA) attacks [BHY09], which are of independent theoretical interest.

Despite the apparent theoretical utility in understanding the extent to which cryptographic constructions are deniable, our current knowledge of constructing such a scheme is still limited. From standard assumptions such as LWE, we have only multi-distributional or non-negligible advantage deniable encryption schemes, whereas with the much more powerful assumption of $i\mathcal{O}$, we can obtain at least fully-secure sender-deniable PKE and computation [CGP15, DKR15, GP15]. A significant gap persists between known feasibility results from standard assumptions and the powerful possibilities from stronger assumptions.

In this work, we further narrow this gap by investigating a richer primitive, inner product encryption (IPE) [KSW08, AFV11, BRS13], *without* the use of obfuscation as a black box primitive. We hope that the techniques developed in this work can further shed light on deniability for even richer schemes such as functional encryption [BSW11, GGH$^+$13, BGG$^+$14, GVW15] under standard assumptions.

1.1 Our Results

- Our main contribution is the construction of a (multi-distributional) bi-deniable IPE from the standard Learning with Errors assumption.

Theorem 1.1 (Informal)**.** *Under the standard* LWE *assumption, there exists a payload-hiding public-key inner product encryption scheme, which is also bi-deniable in the multi-distributional model.*

Recall that in an inner product encryption (IPE) scheme, every secret key $\mathsf{sk}_{\boldsymbol{v}}$ is associated with a predicate vector $\boldsymbol{v} \in \mathbb{Z}_q^\ell$, and every ciphertext $\mathsf{ct}_{\boldsymbol{w}}$ is associated with an attribute vector $\boldsymbol{w} \in \mathbb{Z}_q^\ell$. A ciphertext $\mathsf{ct}_{\boldsymbol{w}}$ can be decrypted by a given secret key $\mathsf{sk}_{\boldsymbol{v}}$ to its payload message m only when $\langle \boldsymbol{v}, \boldsymbol{w} \rangle = 0$. Informally, the security notion for an IPE scheme is *collusion resistance*, which means no collection of keys can provide information on a ciphertext's message, if the individual keys are not authorized to decrypt the ciphertext in the first place. Intuitively, a bi-deniable IPE must provide both collusion and coercion resistance. We also provide the first formal security definition for bi-deniable inner product encryption.

- Our second contribution is a new form of the Extended Learning with Errors (eLWE) assumption [OPW11, ASP12, BLP$^+$13], which is convenient in the context of Dual Regev type functional encryption schemes, such as the IPE of Agrawal, Freeman, and Vaikuntanathan [AFV11].

The eLWE assumption is roughly the LWE assumption, but where the distinguisher also receives "hints" on the LWE sample's noise vector $\boldsymbol{x}$ in the form of (perhaps noisy) inner products, i.e. distributions of the form $\{\mathbf{A}, \boldsymbol{b} = \mathbf{A}^T\boldsymbol{s} + \boldsymbol{x}, \boldsymbol{z}, \langle \boldsymbol{z}, \boldsymbol{x} \rangle\}$ where (intuitively) $\boldsymbol{z}$ is a decryption key. Our main result here is a reduction from the standard LWE assumption to our new form of the extended-LWE assumption, eLWE^+, in the case of a prime polynomial-size modulus even if there is no noise on the hints. We show this by extending the LWE to eLWE reduction of Alperin-Sheriff and Peikert [ASP12] to our particular setting.

- As a further contribution, we believe the techniques developed in the course of our cryptosystem's security proof may be of independent interest toward better understanding LWE-based inner product encryption schemes. Details follow.

1.2 Our Techniques

As in the work of O'Neill et al. [OPW11], our approach to bi-deniability relies primarily on a curious property of Dual Regev type [GPV08] secret keys: by correctness of any such scheme, each key $\boldsymbol{z}$ is guaranteed to behave as intended for some $1 - \mathsf{negl}(n)$ fraction of the possible random coins used to encrypt, but system parameters may be set so that each key is also guaranteed to be *faulty* (i.e. fail to decrypt) on some $\mathsf{negl}(n)$ fraction of the possible encryption randomness. More concretely, each secret key $\boldsymbol{z}$ is sampled from an m-dimensional Gaussian distribution, as is the error term $\boldsymbol{x}$ (for LWE public key $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$). For every fixed $\boldsymbol{z}$, with overwhelming probability over the choice of $\boldsymbol{x}$, the vectors $\boldsymbol{z}, \boldsymbol{x} \in \mathbb{Z}_q^m$ will point in highly uncorrelated directions in m-space. However, if the vector $\boldsymbol{z}$ and $\boldsymbol{x}$ happen to point in similar directions, the error magnitude will be *squared* during decryption.

Our scheme is based around the idea that a receiver, coerced on honest key-ciphertext pair $(\boldsymbol{z}, \boldsymbol{c}^*)$, can use the key authority's *faking key* fk to learn the precise error vector $\boldsymbol{x}^*$ used to construct $\boldsymbol{c}^*$. Given $\boldsymbol{x}^*$, $\boldsymbol{z}$, and fk, the receiver re-samples a fresh secret key $\boldsymbol{z}^*$ that is functionally-equivalent to the honest key $\boldsymbol{z}$, except that $\boldsymbol{z}^*$ is strongly correlated with the vector $\boldsymbol{x}^*$ in $\boldsymbol{c}^*$. When the coercer then attempts to decrypt the challenge ciphertext $\boldsymbol{c}^*$ using $\boldsymbol{z}^*$, the magnitude of decryption error will artificially grow and cause the decryption to output the value we want to deny to. Yet, when the coercer attempts to decrypt any other independently-sampled ciphertext $\boldsymbol{c}$, decryption will succeed with overwhelming probability under $\boldsymbol{z}^*$ if it would have under $\boldsymbol{z}$. We emphasize that to properly show coercion resistance, this behavior of $\boldsymbol{z}^*$ should hold *even when* $\mathbf{c}$ *and* $\boldsymbol{c}^*$ *embed the same attribute vector* $\boldsymbol{w}$.

However to push the above argument through formally, we must overcome a number of technical challenges. The first such challenge is an implicit requirement to *very tightly control* the precise noise magnitude of evaluated ciphertexts. In previous functional (and homomorphic) encryption schemes from lattices, the emphasis is placed on upper bounding evaluated noise terms, to ensure that they do not grow too large and cause decryption to fail. Moreover, security (typically) holds for any ciphertext noise level at or above the starting ciphertexts' noises. In short, noise growth during evaluation is nearly always undesirable.

As with previous schemes, we too must upper bound the noise growth of evaluated ciphertexts in order to ensure basic correctness of our IPE. But unlike previous schemes, we must take the step of also (carefully) *lower bounding* the noise growth during the inner product evaluation. This is due to the fact, highlighted above, that producing directional alignment between a key and error term can at most *square* the noise present during decryption. Since coercion resistance requires that it must always be possible to deny any ciphertext originally intended for any honest key, it must be that, with overwhelming probability, every honest key and every honest ciphertext produce evaluated error that is no less than the square root of the maximum noise threshold tolerated by the scheme.

At a high level, our security proof begins at the Fake experiment, where first a ciphertext $\boldsymbol{c}^*$ and its associated noise terms $\boldsymbol{x}^*$ are sampled, then a fake key $\boldsymbol{z}^*$ is generated that artificially fails to decrypt any ciphertext with noise vector (close to) $\boldsymbol{x}^*$. We then proceed through a sequence of statistically-indistinguishable hybrids, to arrive at an intermediate experiment where first the key $\boldsymbol{z}^*$ is sampled uniformly from the space of valid keys, then noise $\boldsymbol{x}^*$ is instead chosen to be correlated with $\boldsymbol{z}^*$. Once we have an honestly-distributed key $\boldsymbol{z}^*$, we can rely on Extended Learning with Errors (or more specifically, on our new assumption eLWE^+) to show that the artificial correlations with key $\boldsymbol{z}^*$ present in the error term $\boldsymbol{x}^*$ do not leak any additional, meaningful information to an efficient distinguisher. Finally we arrive at the Real experiment, where key $\boldsymbol{z}^*$ is honestly distributed and ciphertext $\boldsymbol{c}^*$ is uniform in the ciphertext space.

The most technically demanding stage of our proof arises when arguing statistical indistinguishability between sampling orders: that is, (i) sampling $\boldsymbol{x}^*$ then $\boldsymbol{z}^*$ in the Fake experiment vs. (ii) sampling $\boldsymbol{z}^*$ then $\boldsymbol{x}^*$ in the Real experiment. In more detail, we will follow the general outline of the LWE-based IPE scheme of [AFV11], where a ciphertext $\boldsymbol{c} = \{\boldsymbol{c}_0, \{\boldsymbol{c}_{i,j}\}, c'\}$, and decryption under $\mathsf{sk}_{\boldsymbol{v}}$ proceeds by including a ciphertext $\boldsymbol{c}_{i,j}$ in the summation $\boldsymbol{c}_{\boldsymbol{v}} = \sum_{\boldsymbol{v}} \boldsymbol{c}_{i,j}$ only if the j-th bit of the i-th $\mathbb{Z}_q$-coordinate of $\boldsymbol{v}$ equals 1. Decryption is completed by checking if $c' - \langle \boldsymbol{z}, [\boldsymbol{c}_0|\boldsymbol{c}_{\boldsymbol{v}}]\rangle$ is closer to 0 than not.

In order to simulate the challenge ciphertext during the security proof, we replace each of the $\boldsymbol{c}_{i,j}$ by the m-vector $\mathbf{R}_{i,j}\boldsymbol{c}_0$ for matrices $\mathbf{R}_{i,j}$ sampled randomly from $\{-1,1\}^{m\times m}$. An application of the leftover hash lemma shows the $\boldsymbol{c}_{i,j}$ remain uniformly distributed. At this point in the simulation, the evaluated error term becomes $\boldsymbol{x}_{\boldsymbol{v}} := \mathbf{R}_{\boldsymbol{v}}\boldsymbol{x}^*$, for $\mathbf{R}_{\boldsymbol{v}} = \sum_{\boldsymbol{v}} \mathbf{R}_{i,j}$ computed as before, and for error vector $\boldsymbol{x}^*$ originally planted in the non-evaluated ciphertext component $\boldsymbol{c}_0$. Indeed, it is this specific error term $\boldsymbol{x}_{\boldsymbol{v}}$ with which fake keys $\boldsymbol{z}^*$ sampled in the Fake experiment must be correlated. The key source of difficulty is that, while each coordinate of honest secret keys $\boldsymbol{z}$ and error terms $\boldsymbol{x}^*$ are (effectively) independently sampled from the spherical Gaussian error distribution χ, the coordinates of $\boldsymbol{x}_{\boldsymbol{v}} = \mathbf{R}_{\boldsymbol{v}}\boldsymbol{x}^*$ are in fact *skewed* by the addition of the random "rotation matrices" $\mathbf{R}_{i,j}$. Consequently, the distribution of $\boldsymbol{x}_{\boldsymbol{v}}$ is an *ellipsoidal* Gaussian distribution, not a spherical one. Thus, naively embedding $\boldsymbol{x}_{\boldsymbol{v}}$ into a new key in an identical manner to O'Neill et al. [OPW11] will produce a key $\boldsymbol{z}^*$ with a distribution that is *statistically distinguishable* from honestly sampled keys $\boldsymbol{z}$.

To avoid this pitfall, we need to take special care across our entire scheme and security proof to ensure that every m-vector – every key, every error term, etc. – is sampled as a multi-dimensional Gaussian with an individualized covariance matrix $\mathbf{Q} \in \mathbb{Z}^{m\times m}$, designed to produce just the right output distribution. Our techniques here rely on elementary applications of probability theory and linear algebra, but we believe they provide both a new technical perspective on Dual Regev type encryption and may serve as a fresh set of tools for approaching such schemes.

2 Preliminaries

Notations. Let PPT denote probabilistic polynomial time. We use bold uppercase letters to denote matrices, and bold lowercase letters to denote vectors. We let λ be the security parameter, $[n]$ denote the set $\{1, ..., n\}$, and $|\boldsymbol{t}|$ denote the number of bits in a string or vector $\boldsymbol{t}$. We denote the i-th bit value of a string $\boldsymbol{s}$ by $\boldsymbol{s}[i]$. We use $[\cdot|\cdot]$ to denote the concatenation of vectors or matrices, and $||\cdot||$ to denote the norm of vectors or matrices respectively. Unless otherwise stated, we use the ℓ_2 norm throughout our work.

2.1 Multi-Distributional Bi-Deniable IPE: Syntax and Bi-Deniability

In this section, we describe the syntax and bi-deniability security definition of a (multi-distributional) bi-deniable inner product encryption (IPE). A multi-distributional bi-deniable inner product encryption scheme consists of a tuple of algorithms (Setup, Keygen, Enc, Dec, $\mathsf{DenSetup}$, $\mathsf{FakeRCoins}$, $\mathsf{FakeSCoins}$):

$\mathsf{Setup}(1^\lambda)$: On input the security parameter λ, the setup algorithm outputs public parameters pp and master secret key msk.

1. $(\boldsymbol{w}^*, \mathsf{state}_1) \leftarrow \mathcal{A}_1(\lambda)$
2. $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$
3. $c \leftarrow \mathsf{Enc}(\mathsf{pp}, \boldsymbol{w}^*, M; r_S)$
4. $(\boldsymbol{v}^*, \mathsf{state}_2) \leftarrow \mathcal{A}_2^{\mathsf{KG}(\mathsf{msk}, \boldsymbol{w}^*, \cdot)}(\mathsf{pp}, \mathsf{state}_1, c)$
5. $\mathsf{sk}_{\boldsymbol{v}^*} \leftarrow \mathsf{Keygen}(\mathsf{msk}, \boldsymbol{v}^*)$
6. $b \leftarrow \mathcal{A}_3^{\mathsf{KG}(\mathsf{msk}, \boldsymbol{w}^*, \cdot)}(\mathsf{sk}_{\boldsymbol{v}^*}, \boldsymbol{c}, \mathsf{state}_2, r_S)$
7. Output $b \in \{0, 1\}$

(a) $\mathbf{Expt}^{\mathsf{Real}}_{\mathcal{A},M,M'}(1^\lambda)$

1. $(\boldsymbol{w}^*, \mathsf{state}_1) \leftarrow \mathcal{A}_1(\lambda)$
2. $(\mathsf{pp}, \mathsf{msk}, \mathsf{fk}) \leftarrow \mathsf{DenSetup}(1^\lambda)$
3. $c' \leftarrow \mathsf{Enc}(\mathsf{pp}, \boldsymbol{w}^*, M'; r_S)$
4. $(\boldsymbol{v}^*, \mathsf{state}_2) \leftarrow \mathcal{A}_2^{\mathsf{KG}(\mathsf{msk}, \boldsymbol{w}^*, \cdot)}(\mathsf{pp}, \mathsf{state}_1, c')$
5. $r'_S \leftarrow \mathsf{FakeSCoins}(\mathsf{pp}, M, M', r_S)$
6. $\mathsf{sk}_{\boldsymbol{v}^*} \leftarrow \mathsf{FakeRCoins}(\mathsf{pp}, \mathsf{fk}, c', \boldsymbol{v}^*, M, M')$
7. $b \leftarrow \mathcal{A}_3^{\mathsf{KG}(\mathsf{msk}, \boldsymbol{w}^*, \cdot)}(\mathsf{sk}_{\boldsymbol{v}^*}, \boldsymbol{c}, \mathsf{state}_2, r'_S)$
8. Output $b \in \{0, 1\}$

(b) $\mathbf{Expt}^{\mathsf{Fake}}_{\mathcal{A},M,M'}(1^\lambda)$

Figure 1: Security experiments for bi-deniable IPE

$\mathsf{Keygen}(\mathsf{msk}, \boldsymbol{v})$: On input the master secret key msk and a predicate vector $\boldsymbol{v}$, the key generation algorithm outputs a secret key $\mathsf{sk}_{\boldsymbol{v}}$ for vector $\boldsymbol{v}$.

$\mathsf{Enc}(\mathsf{pp}, \boldsymbol{w}, M)$: On input the public parameter pp and an attribute/message pair $(\boldsymbol{w}, M)$, it outputs a ciphertext $c_{\boldsymbol{w}}$.

$\mathsf{Dec}(\mathsf{sk}_{\boldsymbol{v}}, c_{\boldsymbol{w}})$: On input the secret key $\mathsf{sk}_{\boldsymbol{v}}$ and a ciphertext $c_{\boldsymbol{w}}$, it outputs the corresponding plaintext M if $\langle \boldsymbol{v}, \boldsymbol{w} \rangle = 0$; otherwise, it outputs $\perp$.

$\mathsf{DenSetup}(1^\lambda)$: On input the security parameter λ, the deniable setup algorithm outputs pubic parameters pp, master secret key msk and faking key fk.

$\mathsf{FakeRCoins}(\mathsf{pp}, \mathsf{fk}, c, \boldsymbol{v}, M, M')$: On input public parameters pp, faking key fk, a ciphertext $c_{\boldsymbol{w}}$ for message M, a predicate attribute $\boldsymbol{v}$, and desired message M', the receiver faking algorithm output a faked secret key $\mathsf{sk}'_{\boldsymbol{v}}$.

$\mathsf{FakeSCoins}(\mathsf{pp}, r_S, M, M')$: On input public parameters pp, original random coins r_S used in encryption of message M and desired message M', it outputs a faked random coin r'_S.

Correctness. We say the bi-deniable IPE scheme described above is correct, if for any $(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{S}(1^\lambda)$, where $\mathsf{S} \in \{\mathsf{Setup}, \mathsf{DenSetup}\}$, any message M, predicate vector $\boldsymbol{v}$, and any attribute vector $\boldsymbol{w}$ where $\langle \boldsymbol{v}, \boldsymbol{w} \rangle$, we have $\mathsf{Dec}(\mathsf{sk}_{\boldsymbol{w}}, c_{\boldsymbol{v}}) = M$, where $\mathsf{sk}_{\boldsymbol{w}} \leftarrow \mathsf{Keygen}(\mathsf{msk}, \boldsymbol{w})$ and $c \leftarrow \mathsf{Enc}(\mathsf{pp}, \boldsymbol{v}, M)$.

Bi-deniability definition. Let M, M' be two arbitrary messages, not necessarily different. We propose the bi-deniability definition by describing real experiment $\mathbf{Expt}^{\mathsf{Real}}_{\mathcal{A},M,M'}(1^\lambda)$ and faking experiment $\mathbf{Expt}^{\mathsf{Fake}}_{\mathcal{A},M,M'}(1^\lambda)$ regarding adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ below:

where $\mathsf{KG}(\mathsf{msk}, \boldsymbol{w}^*, \cdot)$ returns a secret key $\mathsf{sk}_{\boldsymbol{v}} \leftarrow \mathsf{Keygen}(\mathsf{msk}, \boldsymbol{v})$ if $\langle \boldsymbol{v}, \boldsymbol{w}^* \rangle \neq 0$ and $\perp$ otherwise.

Definition 2.1 (Multi-Distributional Bideniable IPE)**.** *An IPE scheme Π is multi-distributional bi-deniable if for any two messages M, M', any probabilistic polynomial-time adversaries $\mathcal{A}$ where $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$, there is a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\mathbf{Adv}^{\Pi}_{\mathcal{A},M,M'}(1^\lambda) = |\mathbf{Pr}[\mathbf{Expt}^{\mathsf{Real}}_{\mathcal{A},M,M'}(1^\lambda) = 1] - \mathbf{Pr}[\mathbf{Expt}^{\mathsf{Fake}}_{\mathcal{A},M,M'}(1^\lambda) = 1]| \leq \mathsf{negl}(\lambda)$$

2.2 Inner-Product-based Bitranslucent Set Scheme

In this section, we extend the bitranslucent set definition proposed by O'Neill et al. in [OPW11] to an inner-product-based counterpart, i.e. an Inner Product Bi-Translucent Set (IP-BTS) scheme. An IP-BTS scheme is made up of the following algorithms:

$\mathsf{Setup}(1^\lambda)$: On input the security parameter, the normal setup algorithm outputs public parameters pp and master secret key msk.

$\mathsf{DenSetup}(1^\lambda)$: On input the security parameter, the deniable setup algorithm outputs public parameters pp, master secret key msk and faking key fk.

$\mathsf{Keygen}(\mathsf{msk}, \boldsymbol{v})$: On input the master secret key msk and a predicate vector $\boldsymbol{v}$, the key generation algorithm outputs a secret key $\mathsf{sk}_{\boldsymbol{v}}$.

P- and U-samplers $\mathsf{SampleP}(\mathsf{pp}, \boldsymbol{w}; r_S)$ (or $\mathsf{SampleU}(\mathsf{pp}; r_S)$) output some $\boldsymbol{c}_{\boldsymbol{w}}$ (or $\boldsymbol{c}$).

$\mathsf{TestP}(\mathsf{sk}_{\boldsymbol{v}}, \boldsymbol{c}_{\boldsymbol{w}})$: On input a secret key $\mathsf{sk}_{\boldsymbol{v}}$ and a ciphertext $\boldsymbol{c}_{\boldsymbol{w}}$, the P-tester algorithm outputs 1 (accepts) or 0 (rejects).

$\mathsf{FakeSCoins}(\mathsf{pp}, r_S)$: On input public parameters pp and randomness r_S, the sender-faker algorithm outputs randomness r_S^*.

$\mathsf{FakeRCoins}(\mathsf{pp}, \mathsf{fk}, \boldsymbol{c}_{\boldsymbol{w}}, \boldsymbol{v})$: On input public parameters pp, the faking key fk and a ciphertext $\boldsymbol{c}_{\boldsymbol{w}}$, the receiver-faker algorithm outputs a faked secret key $\mathsf{sk}'_{\boldsymbol{v}}$.

Definition 2.2 (IP-BTS)**.** *We say the scheme*

$$\Pi = (\mathsf{Setup}, \mathsf{DenSetup}, \mathsf{Keygen}, \mathsf{SampleP}, \mathsf{SampleU}, \mathsf{TestP}, \mathsf{FakeSCoins}, \mathsf{FakeRCoins})$$

is an inner product bitranslucent set scheme if it satisfies:

1. *(Correctness.) We say an IP-BTS scheme is correct if*
   - *For any* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$*, any vector* $\boldsymbol{v}$*,* $\mathsf{sk}_{\boldsymbol{v}} \leftarrow \mathsf{Keygen}(\mathsf{msk}, \boldsymbol{v})$*,if* $\langle \boldsymbol{v}, \boldsymbol{w} \rangle = 0$ *and* $\boldsymbol{c}_{\boldsymbol{w}} \leftarrow \mathsf{SampleP}(\mathsf{pp}, \boldsymbol{w}, r_S)$*, then* $\mathsf{TestP}(\mathsf{sk}_{\boldsymbol{v}}, \boldsymbol{c}_{\boldsymbol{w}}) = 1$*. Otherwise,* $\mathsf{TestP}(\mathsf{sk}_{\boldsymbol{v}}, \boldsymbol{c}_{\boldsymbol{w}}) = 0$*.*
   - *For any* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$*, any vector* $\boldsymbol{v}$*,* $\mathsf{sk}_{\boldsymbol{v}} \leftarrow \mathsf{Keygen}(\mathsf{msk}, \boldsymbol{v})$*, if* $\boldsymbol{c} \leftarrow \mathsf{SampleU}(\mathsf{pp}, r_S)$*, then* $\mathsf{TestP}(\mathsf{sk}_{\boldsymbol{v}}, \boldsymbol{c}) = 0$*.*
2. *(Indistinguishable public parameters.) The public parameters* pp *generated by the two setup algorithms* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ *and* $(\mathsf{pp}, \mathsf{msk}, \mathsf{fk}) \leftarrow \mathsf{DenSetup}(1^\lambda)$ *should be indistinguishable.*
3. *(Bi-deniability.) We propose the selective bi-deniability definition by describing real experiment* $\mathbf{Expt}^{\mathsf{Real}}_{\mathcal{A}}(1^\lambda)$ *and faking experiment* $\mathbf{Expt}^{\mathsf{Fake}}_{\mathcal{A}}(1^\lambda)$ *regarding adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ *below: where* $\mathsf{KG}(\mathsf{msk}, \boldsymbol{w}^*, \cdot)$ *returns a secret key* $\mathsf{sk}_{\boldsymbol{v}} \leftarrow \mathsf{Keygen}(\mathsf{msk}, \boldsymbol{v})$ *if* $\langle \boldsymbol{v}, \boldsymbol{w}^* \rangle \neq 0$ *and* $\perp$ *otherwise.*

   We say the scheme is selectively bi-deniable, if for any probabilistic polynomial-time adversaries $\mathcal{A}$ *where* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$*, there is a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\mathbf{Adv}^{\Pi}_{\mathcal{A}}(1^\lambda) = |\mathbf{Pr}[\mathbf{Expt}^{\mathsf{Real}}_{\mathcal{A}}(1^\lambda) = 1] - \mathbf{Pr}[\mathbf{Expt}^{\mathsf{Fake}}_{\mathcal{A}}(1^\lambda) = 1]| \leq \mathsf{negl}(\lambda)$$

Finally, there is a generic transformation [CDNO97] from multi-distributional (bi)deniable encryption (with a $\mathsf{negl}(\lambda)$ distinguishing advantage) into a "standard" (i.e. single-distribution) (bi)deniable encryption with $1/\mathsf{poly}(\lambda)$ distinguishing advantage, which is best-possible for receiver-deniable encryption by the lower bound of Bendlin et al. [BNNO11].

(a) $(\boldsymbol{v}^*, \boldsymbol{w}^*, \mathsf{state}_1) \leftarrow \mathcal{A}_1(\lambda)$
(b) $(\mathsf{pp}, \mathsf{msk}, \mathsf{fk}) \leftarrow \mathsf{DenSetup}(1^\lambda)$
(c) $\boldsymbol{c} \leftarrow \mathsf{SampleU}(\mathsf{pp}; r_S)$
(d) $\mathsf{state}_2 \leftarrow \mathcal{A}_2^{\mathsf{KG}(\mathsf{msk}, \boldsymbol{w}^*, \cdot)}(\mathsf{pp}, \mathsf{state}_1, \boldsymbol{c})$
(e) $\mathsf{sk}_{\boldsymbol{v}^*} \leftarrow \mathsf{Keygen}(\mathsf{msk}, \boldsymbol{v}^*)$
(f) $b \leftarrow \mathcal{A}_3^{\mathsf{KG}(\mathsf{msk}, \boldsymbol{w}^*, \cdot)}(\mathsf{sk}_{\boldsymbol{v}^*}, \boldsymbol{c}, \mathsf{state}_2, r_S)$
(g) Output $b \in \{0, 1\}$

(a) $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{Real}}(1^\lambda)$

(a) $(\boldsymbol{v}^*, \boldsymbol{w}^*, \mathsf{state}_1) \leftarrow \mathcal{A}_1(\lambda)$
(b) $(\mathsf{pp}, \mathsf{msk}, \mathsf{fk}) \leftarrow \mathsf{DenSetup}(1^\lambda)$
(c) $\boldsymbol{c} \leftarrow \mathsf{SampleP}(\mathsf{pp}, \boldsymbol{w}^*; r_S)$
(d) $\mathsf{state}_2 \leftarrow \mathcal{A}_2^{\mathsf{KG}(\mathsf{msk}, \boldsymbol{w}^*, \cdot)}(\mathsf{pp}, \mathsf{state}_1, \boldsymbol{c})$
(e) $r'_S \leftarrow \mathsf{FakeSCoins}(\mathsf{pp}; r_S)$
(f) $\mathsf{sk}_{\boldsymbol{v}^*} \leftarrow \mathsf{FakeRCoins}(\mathsf{pp}, \mathsf{fk}, \boldsymbol{c}, \boldsymbol{v}^*)$
(g) $b \leftarrow \mathcal{A}_3^{\mathsf{KG}(\mathsf{msk}, \boldsymbol{w}^*, \cdot)}(\mathsf{sk}_{\boldsymbol{v}^*}, \boldsymbol{c}, \mathsf{state}_2, r'_S)$
(h) Output $b \in \{0, 1\}$

(b) $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{Fake}}(1^\lambda)$

Figure 2: Security experiments for IP-BTS

Remark 2.3. *Correctness for the faking algorithms is implied by the bi-deniability property. In particular, with overwhelming probability over the randomness, the following holds: let* $(\mathsf{pp}, \mathsf{msk}, \mathsf{fk}) \leftarrow \mathsf{DenSetup}(1^\lambda)$, *let* $\boldsymbol{x}, \boldsymbol{y}$ *be any vectors, let* $\mathsf{sk}_{\boldsymbol{y}} \leftarrow \mathsf{Keygen}(\mathsf{msk}, \boldsymbol{y})$, *and let* $\boldsymbol{c}_{\boldsymbol{x}} \leftarrow \mathsf{SampleP}(\mathsf{pp}, \boldsymbol{x}; r_S)$, *then*

- $\mathsf{SampleU}(\mathsf{pp}; \mathsf{FakeSCoins}(\mathsf{pp}, r_S)) = \boldsymbol{c}_{\boldsymbol{x}}$,
- $\mathsf{TestP}(\mathsf{FakeRCoins}(\mathsf{pp}, \mathsf{fk}, \boldsymbol{c}_{\boldsymbol{x}}, \boldsymbol{y}), \boldsymbol{c}_{\boldsymbol{x}}) = 0$, *and*
- *for any other* $\boldsymbol{x}'$, *let* $\boldsymbol{c}' \leftarrow \mathsf{SampleP}(\mathsf{pp}, \boldsymbol{x}'; r'_S)$, *then (with overwhelming probability) we have*

$$\mathsf{TestP}\left(\mathsf{FakeRCoins}(\mathsf{pp}, \mathsf{fk}, \boldsymbol{c}_{\boldsymbol{x}}, \boldsymbol{y}), \boldsymbol{c}'\right) = \mathsf{TestP}(\mathsf{sk}_{\boldsymbol{y}}, \boldsymbol{c}').$$

It is not hard to see that if one of these does not hold, then one can easily distinguish the real experiment from the faking experiment by performing the test prescribed.

Remark 2.4 (Adaptive bi-deniability)**.** *We say the IP-BTS scheme is adaptively bi-deniable, if the adversary* $\mathcal{A}$ *does not need to commit to the challenge functionality* $(\boldsymbol{v}^*, \boldsymbol{w}^*)$ *before obtaining public parameters* pp.

Lemma 2.5. *The existence of a inner product bitranslucent set scheme (IP-BTS) implies existence of a multi-distributional bi-deniable IPE scheme, secure under Definition 2.1.*

Proof Sketch. Canetti et al. [CDNO97] gave a simple encoding trick to construct a multi-distributional sender-deniable encryption scheme from a translucent set. O'Neill, Peikert, and Waters [OPW11] gave a similar trick for constructing multi-distributional bi-deniable encryption from a bi-translucent set scheme. We observe a similar trick works here:

Encryption is performed bit-wise on the message M. The normal encryption algorithm encrypts a bit 0 as the pair of samples (U, U) and a bit 1 as (U, P). The IPE simulator encrypts a bit 0 as (P, P) and a bit 1 as (U, P). If the simulator needs to open an encryption of 0 as a 1, he uses $\mathsf{FakeSCoins}$ and $\mathsf{FakeRCoins}$ to make a pair (P, P) appear as (U, P) under TestP. Similarly to open an encryption of 1 as a 0, the simulator can use $\mathsf{FakeSCoins}$ and $\mathsf{FakeRCoins}$ to make a pair (U, P) appear as (U, U) under TestP.

The remainder of the proof is a routine calculation. □

2.3 Lattice Background

Throughout our work, without loss of generality we treat $\mathbb{Z}_q$ as the subset of integers $(-q/2, q/2] \cap \mathbb{Z}$, and define the set $\mathbb{Z}_1 \stackrel{\text{def}}{=} \{-1/2+1/q, -1/2+2/q, \ldots, 1/2-1/q, 1/2\}$ representing the range $(-1/2, 1/2] \subset \mathbb{R}$ with bit-precision $\log_2(q)$. We define the operators $(\bmod\ q)$ and $(\bmod\ 1)$ to map into these sets in the natural way. We note that for any $\boldsymbol{x}_0, \boldsymbol{x}_1 \in \mathbb{Z}_q^n$ and $\boldsymbol{y}_0, \boldsymbol{y}_1 \in \mathbb{Z}_1^n$ where $\boldsymbol{x}_0 = q\boldsymbol{y}_0, \boldsymbol{x}_1 = q\boldsymbol{y}_1$, it holds that $q\langle \boldsymbol{x}_0/q, \boldsymbol{x}_1/q\rangle = \langle \boldsymbol{y}_0, \boldsymbol{y}_1\rangle \in \mathbb{Z}_1$. That is, we have $\langle \boldsymbol{x}_0, \boldsymbol{y}_1\rangle = \langle \boldsymbol{x}_0, \boldsymbol{x}_1/q\rangle \in \mathbb{Z}_1$. (The reader should think of the multiplication operation in our inner product definition as operating on each input-argument, written as a *relative ratio* of the argument's domain's size, q; i.e. over the rationals $\mathbb{Q}$ or, in general, the reals $\mathbb{R}$ modulo 1. In prior works, this is sometimes alternatively denoted by the torus $\mathbb{T}$.)

A full-rank m-dimensional integer lattice $\Lambda \subset \mathbb{Z}^m$ is a discrete additive subgroup whose linear span is $\mathbb{R}^m$. The basis of Λ is a linearly independent set of vectors whose linear combinations are exactly Λ. Every integer lattice is generated as the $\mathbb{Z}$-linear combination of linearly independent vectors $\mathbf{B} = \{\boldsymbol{b}_1, ..., \boldsymbol{b}_m\} \subset \mathbb{Z}^m$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we define the "$q$-ary" integer lattices:

$$\Lambda_q^{\perp} = \{\boldsymbol{e} \in \mathbb{Z}^m | \mathbf{A}\boldsymbol{e} = 0 \bmod q\}, \qquad \Lambda_q^{\mathbf{u}} = \{\boldsymbol{e} \in \mathbb{Z}^m | \mathbf{A}\boldsymbol{e} = \boldsymbol{u} \bmod q\}$$

It is obvious that $\Lambda_q^{\boldsymbol{u}}$ is a coset of $\Lambda_q^{\perp}$.

Let Λ be a discrete subset of $\mathbb{Z}^m$. For any vector $\boldsymbol{c} \in \mathbb{R}^m$, and any positive parameter $\sigma \in \mathbb{R}$, let $\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x}) = \exp(-\pi||\boldsymbol{x} - \boldsymbol{c}||^2/\sigma^2)$ be the Gaussian function on $\mathbb{R}^m$ with center $\boldsymbol{c}$ and parameter σ. Next, we set $\rho_{\sigma,\boldsymbol{c}}(\Lambda) = \sum_{\boldsymbol{x}\in\Lambda} \rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x})$ be the discrete integral of $\rho_{\sigma,\boldsymbol{x}}$ over Λ, which gives the Discrete Gaussian distribution $\mathcal{D}_{\Lambda,\sigma,\boldsymbol{c}}(\boldsymbol{y}) := \frac{\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{y})}{\rho_{\sigma,\boldsymbol{c}}(\Lambda)}$. We will sometimes use the distribution $\mathcal{D}_{\Lambda,\sigma}$, which is understood as centered at the origin, or when the context is clear, we will sometimes use $\mathcal{D}_\sigma$ to denote sampling over $\mathbb{R}$, then rounding to an appropriate element.

More frequently, we will use the generalized multi-dimensional (or m-variate) Discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}_1^m,\mathbf{Q}}$, which denotes sampling a $\mathbb{Z}_1$-valued m-vector with *covariance matrix* $\mathbf{Q} \in \mathbb{Z}_1^{m\times m}$. In order to sample from the distribution $\mathcal{D}_{\mathbb{Z}_1^m,\mathbf{Q}}$, proceed as follows:

- Sample $\boldsymbol{t}' = (t'_1, ..., t'_m) \in \mathbb{R}^m$ independently as $t'_i \leftarrow \mathcal{D}_1$ for $i \in [m]$.
- Find the Cholesky decomposition $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$.
- Output the vector $\boldsymbol{t} := \mathbf{L}\boldsymbol{t}'$ as the sample $\boldsymbol{t} \leftarrow \mathcal{D}_{\mathbb{Z}_1^m,\mathbf{Q}}$.

Recall that the Cholesky decomposition takes as input any positive-definite matrix $\mathbf{Q} \in \mathbb{R}^{m\times m}$ and outputs a lower triangular matrix $\mathbf{L}$ so that $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$. Further, we recall the fact that the sum of two m-variate Gaussians with means μ_1, μ_2 and variances $\mathbf{Q}_1, \mathbf{Q}_2$ is an m-variate Gaussian with mean $\mu_1 + \mu_2$ and variance $\mathbf{Q}_1 + \mathbf{Q}_2$.

Next we show a useful lemma that we need for our construction.

Lemma 2.6. *Let $\mathbf{I}_{m\times m}$ be the m-by-m identity matrix, $\mathbf{R} \in \mathbb{R}^{m\times m}$, and $\mathbf{Q} \stackrel{\text{def}}{=} a^2\mathbf{I}_{m\times m} - b^2\mathbf{R}^T\mathbf{R}$ for positive numbers a, b such that $a > b||\mathbf{R}||$. Then $\mathbf{Q}$ is positive definite.*

Proof. To show that $\mathbf{Q}$ is positive definite, we need to show that for any column vector $\boldsymbol{x}$ of dimension m, we have $\boldsymbol{x}^T \cdot \mathbf{Q} \cdot \boldsymbol{x} > 0$. We prove this by unfolding the matrix $\mathbf{Q}$:

$$\begin{aligned}\boldsymbol{x}^T \cdot \mathbf{Q} \cdot \boldsymbol{x} &= \boldsymbol{x}^T \cdot (a^2\mathbf{I}_{m\times m} - b^2\mathbf{R}^T\mathbf{R}) \cdot \boldsymbol{x} \\ &= a^2\boldsymbol{x}^T\mathbf{I}_{m\times m}\boldsymbol{x} - b^2\boldsymbol{x}^T\mathbf{R}^T\mathbf{R}\boldsymbol{x} \\ &= a^2||\boldsymbol{x}||^2 - b^2||\mathbf{R}\boldsymbol{x}||^2 \\ &> b^2||\mathbf{R}||^2 \cdot ||\boldsymbol{x}||^2 - b^2||\mathbf{R}\boldsymbol{x}||^2.\end{aligned}$$

Since $||\mathbf{R}|| \cdot ||\boldsymbol{x}|| \geq ||\mathbf{R}\boldsymbol{x}||$, we can conclude $\boldsymbol{x}^T \cdot \mathbf{Q} \cdot \boldsymbol{x} > 0$. □

Randomness extraction. We will use the following lemma to argue the indistinghishability of two different distributions, which is a generalization of the leftover hash lemma proposed by Dodis et al. [DRS04]. We use the lattice based leftover hash lemma in [ABB10].

Lemma 2.7 ([ABB10])**.** *Suppose that $m > (n+1)\log q + w(\log n)$. Let $\mathbf{R} \in \{-1,1\}^{m\times k}$ be chosen uniformly at random for some polynomial $k = k(n)$. Let $\mathbf{A}, \mathbf{B}$ be matrix chosen randomly from $\mathbb{Z}_q^{n\times m}, \mathbb{Z}_q^{n\times k}$ respectively. Then, for all vectors $\boldsymbol{w} \in \mathbb{Z}^m$, the two following distributions are statistically close:*

$$(\mathbf{A}, \mathbf{A}\mathbf{R}, \mathbf{R}^T\boldsymbol{w}) \approx (\mathbf{A}, \mathbf{B}, \mathbf{R}^T\boldsymbol{w})$$

Trapdoors and sampling algorithms. We will use the following algorithms to sample short vectors from specified lattices.

Lemma 2.8 ([GPV08])**.** *Let q, n, m be positive integers with $q \geq 2$ and sufficiently large $m = \Omega(n\log q)$. There exists a* PPT *algorithm* $\mathsf{TrapGen}(q, n, m)$ *that with overwhelming probability outputs a pair $(\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \mathbf{T_A} \in \mathbb{Z}^{m\times m})$ such that $\mathbf{A}$ is statistically close to uniform in $\mathbb{Z}_q^{n\times m}$ and $\mathbf{T_A}$ is a basis for $\Lambda_q^{\perp}(\mathbf{A})$ satisfying $||\mathbf{T_A}|| \leq O(n\log q)$.*

Lemma 2.9 ([GPV08, CHKP10, ABB10])**.** *Let $q > 2, m > n$ and $s > ||\mathbf{T_A}|| \cdot w(\sqrt{\log m + m_1})$. There are several polynomial time algorithms as follows:*

- *There is an efficient algorithm* $\mathsf{SampleLeft}(\mathbf{A}, \mathbf{B}, \mathbf{T_A}\boldsymbol{u}, s)$*: It takes in $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a short basis $\mathbf{T_A}$ for lattice $\Lambda_q^{\perp}(\mathbf{A})$, a matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m_1}$, a vector $\boldsymbol{u} \in \mathbb{Z}_q^n$ and a Gaussian parameter s, then outputs a vector $\mathbf{e} \in \mathbb{Z}_q^{m+m_1}$ such that $\boldsymbol{e} \in \Lambda_q^{\boldsymbol{u}}(\mathbf{F})$, where $\mathbf{F} := (\mathbf{A}|\mathbf{B})$, and is statistical close to $\mathcal{D}_{\Lambda_q^{\boldsymbol{u}}(\mathbf{F}),s}$.*

- *There is an efficient algorithm* $\mathsf{SampleRight}(\mathbf{A}, \mathbf{B}, \mathbf{R}, \mathbf{T_B}, \boldsymbol{u}, s)$*: It takes in $\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \mathbf{R} \in \mathbb{Z}_q^{m\times n}$, a matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times n}$, a short basis $\mathbf{T_B}$ for lattice $\Lambda_q^{\perp}(\mathbf{B})$, a vector $\boldsymbol{u} \in \mathbb{Z}_q^n$ and a Gaussian parameter s, then outputs a vector $\boldsymbol{e} \in \mathbb{Z}_q^{m+n}$ such that $\boldsymbol{e} \in \Lambda_q^{\mathbf{u}}(\mathbf{F})$, where $\mathbf{F} := (\mathbf{A}|\mathbf{A}\mathbf{R}+\mathbf{B})$, and is statistical close to $\mathcal{D}_{\Lambda_q^{\boldsymbol{u}}(\mathbf{F}),s}$.*

- *There is an efficient algorithm* $\mathsf{SamplePre}$ *that takes as input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ together with its trapdoor $\mathbf{T_A}$, and a vector $\boldsymbol{u} \in \mathbb{Z}_q^n$ and outputs a matrix $\mathbf{e} \in \mathbb{Z}^m$ from $\mathcal{D}_{\Lambda^{\perp}(\mathbf{A})+\boldsymbol{u},r}$ (up to* $\mathsf{negl}(n)$ *statistical distance.)*

- *There is a deterministic polynomial-time algorithm* $\mathsf{ExtBasis}(\mathbf{A}, \mathbf{T_A}, \mathbf{A}')$ *that takes in an arbitrary $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, whose columns generate the entire group $\mathbf{Z}_q^n$, an arbitrary basis $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ of $\Lambda^{\perp}(\mathbf{A})$, then outputs a basis $\mathbf{T}'$ of $\Lambda^{\perp}(\mathbf{A}|\mathbf{A}')$, such that $||\mathbf{T}|| = ||\mathbf{T_A}||$. Moreover, the same holds even for any given permutation of columns of $\mathbf{A}'$.*

- *There is a deterministic polynomial time algorithm* $\mathsf{Invert}(\mathbf{A}, \mathbf{T_A}, \boldsymbol{b})$ *that, given any $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with its trapdoor $\mathbf{T_A} \in \mathbb{Z}_q^{m\times m}$ such that $||\mathbf{T}|| \cdot w(\sqrt{\log n}) \leq 1/\beta$ for some $\beta > 0$, and $\boldsymbol{b} = \mathbf{A}^T\boldsymbol{s} + \boldsymbol{x}$ for arbitrary $\boldsymbol{s} \in \mathbb{Z}_q^n$ and random $\boldsymbol{x} \leftarrow \mathcal{D}_{\beta}^m$, outputs $\boldsymbol{x}$ with overwhelming probability.*

3 Learning with Errors and Extended Learning with Errors

The LWE problem was introduced by Regev [Reg05], who showed that solving it *on the average* is as hard as (quantumly) solving several standard lattice problems *in the worst case*.

Definition 3.1 (LWE)**.** *For an integer $q = q(n) \geq 2$, and an error distribution $\chi = \chi(n)$ over $\mathbb{Z}_q$, the learning with errors problem $\mathsf{LWE}_{n,m,q,\chi}$ is to distinguish between the following pairs of distributions:*

$$\{\mathbf{A}, \boldsymbol{b} = \mathbf{A}^T \boldsymbol{s} + \boldsymbol{x}\} \text{ and } \{\mathbf{A}, \boldsymbol{u}\}$$

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\boldsymbol{u} \xleftarrow{\$} \mathbb{Z}_q^m$, and $\boldsymbol{x} \xleftarrow{\$} \chi^m$.

O'Neill et al. [OPW11] introduced the extended-LWE problem, which allows a "hint" on the error vector $\boldsymbol{x}$ to leak in form of a noisy inner product. They observe a trivial "blurring" argument shows that LWE reduces to eLWE when the hint-noise βq is superpolynomially larger than the magnitude of samples from χ, and also allows for unboundedly many *independent* hint vectors $\langle \boldsymbol{z}, \boldsymbol{x}_i \rangle$ while retaining LWE-hardness.

Definition 3.2 (Extended LWE)**.** *For an integer $q = q(n) \geq 2$, and an error distribution $\chi = \chi(n)$ over $\mathbb{Z}_q$, the extended learning with errors problem $\mathsf{eLWE}_{n,m,q,\chi,\beta}$ is to distinguish between the following pairs of distributions:*

$$\{\mathbf{A}, \boldsymbol{b} = \mathbf{A}^T \boldsymbol{s} + \boldsymbol{x}, \boldsymbol{z}, \langle \boldsymbol{z}, \boldsymbol{b} - \boldsymbol{x} \rangle + x'\} \text{ and } \{\mathbf{A}, \boldsymbol{u}, \boldsymbol{z}, \langle \boldsymbol{z}, \boldsymbol{u} - \boldsymbol{x} \rangle + x'\}$$

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\boldsymbol{u} \xleftarrow{\$} \mathbb{Z}_q^m$, $\boldsymbol{x}, \boldsymbol{z} \xleftarrow{\$} \chi^m$ and $x' \xleftarrow{\$} \mathcal{D}_{\beta q}$.

Further, Alperin-Sheriff and Peikert [ASP12] show that LWE reduces to eLWE with a polynomial modulus and no hint-noise (i.e. $\beta = 0$), even in the case of a bounded number of *independent* hints.

We introduce the following new form of extended-LWE, called eLWE^+, which considers leaking a pair of *correlated hints* on the same noise vector.

Definition 3.3 (Extended LWE Plus)**.** *For integer $q = q(n) \geq 2, m = m(n)$, an error distribution $\chi = \chi(n)$ over $\mathbb{Z}_q$, and a matrix $\mathbf{R} \in \mathbb{Z}_q^{m \times m}$, the extended learning with errors problem $\mathsf{eLWE}^+_{n,m,q,\chi,\beta,\mathbf{R}}$ is to distinguish between the following pairs of distributions:*

$$\{\mathbf{A}, \boldsymbol{b} = \mathbf{A}^T \boldsymbol{s} + \boldsymbol{x}, \boldsymbol{z}_0, \boldsymbol{z}_1, \langle \boldsymbol{z}_0, \boldsymbol{b} - \boldsymbol{x} \rangle + x, \langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{b} - \boldsymbol{x} \rangle + x'\} \text{ and}$$

$$\{\mathbf{A}, \boldsymbol{u}, \boldsymbol{z}_0, \boldsymbol{z}_1, \langle \boldsymbol{z}_0, \boldsymbol{u} - \boldsymbol{x} \rangle + x, \langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{u} - \boldsymbol{x} \rangle + x'\}$$

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\boldsymbol{u} \xleftarrow{\$} \mathbb{Z}_q^m$, $\boldsymbol{x}, \boldsymbol{z}_0, \boldsymbol{z}_1 \xleftarrow{\$} \chi^m$ and $x, x' \xleftarrow{\$} \mathcal{D}_{\beta q}$.

Hardness of extended-LWE$^+$. A simple observation, following prior work, is that when χ is $\mathsf{poly}(n)$-bounded and the hint noise βq (and thus, modulus q) is superpolynomial in n, then $\mathsf{LWE}_{n,m,q,\chi}$ trivially reduces to $\mathsf{eLWE}^+_{n,m,q,\chi,\beta,\mathbf{R}}$ for every $\mathbf{R} \in \mathbb{Z}_q^{m \times m}$ so that $\mathbf{R}\boldsymbol{z}_1$ has $\mathsf{poly}(n)$-bounded norm. This is because, for any $r = \omega(\sqrt{\log n}), c \in \mathbb{Z}$, the statistical distance between $\mathcal{D}_{\mathbb{Z},r}$ and $c + \mathcal{D}_{\mathbb{Z},r}$ is at most $O(|c|/r)$.

However, our cryptosystem will require a polynomial-size modulus q. So, we next consider the case of *prime* modulus q of $\mathsf{poly}(n)$ size and no noise on the hints (i.e. $\beta = 0$). Following [ASP12][2], it will be convenient to swap to the "knapsack" form of LWE, which is: given $\mathbf{H} \leftarrow \mathbb{Z}_q^{(m-n) \times m}$ and $\boldsymbol{c} \in \mathbb{Z}_q^{m-n}$, where either $\mathbf{c} = \mathbf{H}\boldsymbol{x}$ for $\boldsymbol{x} \leftarrow \chi^m$ or $\mathbf{c}$ uniformly random and independent of $\mathbf{H}$, determine which is the case (with non-negligible advantage). The "extended-plus" form of the knapsack problem also reveals a pair of hints $(\boldsymbol{z}_0, \boldsymbol{z}_1, \langle \boldsymbol{z}_0, \boldsymbol{x} \rangle, \langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{x} \rangle)$. Note the equivalence between LWE and knapsack-LWE is proven in [MM11] for $m \geq n + \omega(\log n)$.

[2] We note that a higher quality reduction from LWE to eLWE is given in [BLP$^+$13] in the case of binary secret keys. However for our cryptosystem, it will be more convenient to have secret key coordinates in $\mathbb{Z}_q$, so we extend the reduction of [ASP12] to eLWE^+ instead.

Theorem 3.4. *For $m \geq n + \omega(\log n)$, for every prime $q = \mathsf{poly}(n)$, for every $\mathbf{R} \in \mathbb{Z}_q^{m \times m}$, and for every $\beta \geq 0$,* $\mathbf{Adv}_{\mathcal{BA}}^{\mathsf{LWE}_{n,m,q,\chi}}(1^\lambda) \geq (1/q^2)\mathbf{Adv}_{\mathcal{A}}^{\mathsf{eLWE}^+_{n,m,q,\chi,\beta,\mathbf{R}}}(1^\lambda)$.

Proof. We construct an LWE to eLWE$^+$ reduction $\mathcal{B}$ as follows. $\mathcal{B}$ receives a knapsack-LWE instance $\mathbf{H} \in \mathbb{Z}_q^{(m-n)\times m}, \boldsymbol{c} \in \mathbb{Z}_q^{m-n}$. It samples $\boldsymbol{x}', \boldsymbol{z}_0, \boldsymbol{z}_1 \leftarrow \chi^m$ and uniform $\boldsymbol{v}_0, \boldsymbol{v}_1 \leftarrow \mathbb{Z}_q^{m-n}$. It chooses any $\mathbf{R} \in \mathbb{Z}_q^{m \times m}$, then sets

$$\mathbf{H}' := \mathbf{H} - \boldsymbol{v}_0 \boldsymbol{z}_0^T - \boldsymbol{v}_1 \left(\mathbf{R}\boldsymbol{z}_1\right)^T \in \mathbb{Z}_q^{(m-n)\times m},$$
$$\boldsymbol{c}' := \boldsymbol{c} - \boldsymbol{v}_0 \cdot \langle \boldsymbol{z}_0, \boldsymbol{x}' \rangle - \boldsymbol{v}_1 \cdot \langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{x}' \rangle \in \mathbb{Z}_q^{m-n}.$$

It sends $(\mathbf{H}', \boldsymbol{c}', \mathbf{z}_0, \mathbf{z}_1, \langle \boldsymbol{z}_0, \boldsymbol{x}' \rangle, \langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{x}' \rangle)$ to the knapsack-eLWE$^+$ adversary $\mathcal{A}$, and outputs what $\mathcal{A}$ outputs.

Notice that when $\mathbf{H}, \boldsymbol{c}$ are independent and uniform, so are $\mathbf{H}', \mathbf{c}'$, in which case $\mathcal{B}$'s simulation is perfect.

Now, consider the case when $\mathbf{H}, \mathbf{c}$ are drawn from the knapsack-LWE distribution, with $\mathbf{c} = \mathbf{H}\boldsymbol{x}$ for $\boldsymbol{x} \leftarrow \chi^m$. In this case, $\mathbf{H}'$ is uniformly random over the choice of $\mathbf{H}$, and we have

$$\begin{aligned}
\boldsymbol{c}' &= \mathbf{H}\boldsymbol{x} - \boldsymbol{v}_0 \cdot \langle \boldsymbol{z}_0, \boldsymbol{x}' \rangle - \boldsymbol{v}_1 \cdot \langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{x}' \rangle \\
&= \left(\mathbf{H}' + \boldsymbol{v}_0 \boldsymbol{z}_0^T + \boldsymbol{v}_1 \left(\mathbf{R}\boldsymbol{z}_1\right)^T\right) \boldsymbol{x} - \boldsymbol{v}_0 \cdot \langle \boldsymbol{z}_0, \boldsymbol{x}' \rangle - \boldsymbol{v}_1 \cdot \langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{x}' \rangle \\
&= \mathbf{H}'\boldsymbol{x} + \boldsymbol{v}_0 \cdot \langle \boldsymbol{z}_0, \boldsymbol{x} - \boldsymbol{x}' \rangle + \boldsymbol{v}_1 \cdot \langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{x} - \boldsymbol{x}' \rangle.
\end{aligned}$$

Define the event $E = [E_0 \wedge E_1]$ as

$$E_0 \stackrel{\text{def}}{=} \left[\langle \boldsymbol{z}_0, \boldsymbol{x} \rangle = \langle \boldsymbol{z}_0, \boldsymbol{x}' \rangle\right],$$
$$E_1 \stackrel{\text{def}}{=} \left[\langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{x} \rangle = \langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{x}' \rangle\right].$$

If event E occurs, then the reduction $\mathcal{B}$ perfectly simulates a pseudorandom instance of knapsack-eLWE$^+$ to $\mathcal{A}$, as then $\boldsymbol{v}_0 \cdot \langle \boldsymbol{z}_0, \boldsymbol{x} - \boldsymbol{x}' \rangle + \boldsymbol{v}_1 \cdot \langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{x} - \boldsymbol{x}' \rangle$ vanishes, leaving $\boldsymbol{c}' = \mathbf{H}'\boldsymbol{x}$ for $\mathbf{H}' \leftarrow \mathbb{Z}_q^{(m-n)\times m}$ and $\boldsymbol{x} \leftarrow \chi^m$ as required. Otherwise since q is prime, the reduction $\mathcal{B}$ (incorrectly) simulates an independent and uniform instance of knapsack-eLWE$^+$ to $\mathcal{A}$, as then either one of $\boldsymbol{v}_0 \cdot \langle \boldsymbol{z}_0, \boldsymbol{x} - \boldsymbol{x}' \rangle$ or $\boldsymbol{v}_1 \cdot \langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{x} - \boldsymbol{x}' \rangle$ does not vanish, implying that $\boldsymbol{c}'$ is uniform in $\mathbb{Z}_q^{m-n}$ over the choice of $\boldsymbol{v}_0$ (resp. $\boldsymbol{v}_1$) alone, independent of the choices of $\mathbf{H}'$ and $\boldsymbol{x}$.

It remains to analyze the probability that event E occurs. Because $\boldsymbol{x}$ and $\boldsymbol{x}'$ are i.i.d., we may define the random variable $\mathcal{Z}_0$ that takes values $\langle \boldsymbol{z}_0, \boldsymbol{x}^* \rangle \in \mathbb{Z}_q$ and the random variable $\mathcal{Z}_1$ that takes values $\langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{x}^* \rangle \in \mathbb{Z}_q$ jointly over choice of $\boldsymbol{x}^* \leftarrow \chi^m$, and analyze their collision probabilities independently. Since the collision probability of *any* random variable $\mathcal{Z}$ is at least $1/|\mathsf{Supp}(\mathcal{Z})|$, we have that $\Pr[E] \geq \min CP[\mathcal{Z}_0] \cdot \min CP[\mathcal{Z}_1] = 1/q^2 = 1/\mathsf{poly}(n)$, and the theorem follows. □

4 Tighter Error Analysis

In this section, we provide some useful lemmas for a tighter analysis of the error growth in our IPE construction. Our construction basically follows the IPE construction by Agrawal et al. [AFV11]. The analysis of the scheme requires bounding evaluated noise of the form $\boldsymbol{z}^T \cdot \boldsymbol{x}_v$, where $\boldsymbol{z}$ is a secret key and $\boldsymbol{x}_v$ is the noise of an evaluated ciphertext, which has the form $\boldsymbol{x}_v = \mathbf{R}\boldsymbol{x}$, where $\mathbf{R}$ is a random $\{-1, 1\}^{m \times m}$ matrix (or a sum of several such matrices), and $\boldsymbol{x}$ is the error term of the original ciphertext(s). To explain our tighter analysis, we can think of a simplified version where $\boldsymbol{z}, \boldsymbol{s}$ are samples from the m dimensional

Gaussian distributions with width s, α respectively. (There are other terms in the actual construction, but here for exposition we just focus on the simplified form.)

As discussed in the introduction, in order to achieve deniability while maintaining correctness of decryption, we need to further leverage the gap between $||\boldsymbol{z}^T \cdot \boldsymbol{x}_v||$, and $||\boldsymbol{x}_v^T \cdot \boldsymbol{x}_v||$, where the former refers to the decryption correctness bound, and the latter refers to the deniability bound. We require the former to be small, and the latter to be large. In this work, we carefully bound these terms and show that $||\boldsymbol{z}^T \cdot \boldsymbol{x}_v|| \approx m\alpha s$, and $||\boldsymbol{x}_v^T \cdot \boldsymbol{x}_v|| \approx m^2\alpha s$. The gap of m is crucial so that our parameters have a feasible region. In particular, we will eventually *lose* an additional $\sqrt{m}$ factor in this gap, in order to ensure positive-definiteness of certain matrices in our construction. Therefore, we need this gap to be at least $m^{1/2+\delta}$ for $\delta > 0$ to ensure feasibility in the end.

Our analysis uses a careful application of Hoeffding's inequality on truncated random variables. Basically Hoeffding's inequality shows that for i.i.d. random variables $Y_1, Y_2, \ldots, Y_m$, the probability $Y = Y_1 + \cdots + Y_m - E[Y] > t$ is small for an appropriate setting of t. However, there is a subtlety when we apply this inequality: if the Y_is may possibly take values in a large range, then the bound given by the inequality is not as sharp, and in fact this is exactly our case. To overcome this, we first argue that with high probability, each Y_i take values in a much smaller range w.h.p. Therefore, we can first truncate the random variables Y_i to cut out the large values, which only induces a negligible statistical distance. Then we apply Hoeffding's inequality on the truncated random variables (with a lower upper bound) to obtain a sharper bound overall.

We note that previously, Agrawal et al. [AFV11] showed that $||\boldsymbol{z}^T \cdot \boldsymbol{x}_v|| \leq ||\boldsymbol{z}^T|| \cdot ||\mathbf{R}|| \cdot ||\boldsymbol{x}|| \approx m^{1.5}\alpha s$. This bound is sufficient for the normal IPE setting where only correctness is required. However as discussed above, it is not sufficient for us because a gap of $\sqrt{m}$ is (precisely!) too small to allow a feasible region for our parameters.

Lemma 4.1. *Let* $\mathbf{R}$ *is an* $m \times m$ *be a matrix chosen at random from* $\{-1, 1\}^{m \times m}$*, and* $\boldsymbol{u} = (u_1, ..., u_m) \in \mathbb{R}^m$ *be a vector chosen according to the* m *dimensional Gaussian with width* α*. Then we have*

$$\Pr\left[||\mathbf{R}\boldsymbol{u}||^2 \in \Theta(m^2\alpha^2)\right] > 1 - \mathsf{negl}(m).$$

Proof. We know with overwhelming probability over the choice of $\boldsymbol{u}$, all of its entries have absolute value less than $B = \alpha\omega(\log m)$. Also, we know that with overwhelming probability, we have $||\boldsymbol{u}||^2 = \Theta(m\alpha^2)$. We call a sample typical if it satisfies these two conditions. Note that it is without loss of generality to just consider the typical samples, from a simple union bound argument.

Then we consider a fixed typical choice of vector $\boldsymbol{u} = (u_1, ..., u_m) \in \mathbb{R}^m$. We write the inner product of $\boldsymbol{r}^T \cdot \boldsymbol{u}$ where $\boldsymbol{r} = (r_1, \ldots, r_m)$ is sampled uniformly from $\{-1, 1\}^m$. We observe that $\mathbb{E}\left[||\boldsymbol{r}^T \cdot \boldsymbol{u}||^2\right] = \mathbb{E}\left[\sum_{i=1}^m r_i^2 u_i^2 + \sum_{i<j\leq m} r_i r_j u_i u_j\right] = \sum_{i=1}^m u_i^2 = ||\boldsymbol{u}||^2$. This is because each r_i, r_j are independent and have mean 0.

Now, for such a fixed $\boldsymbol{u}$ we denote random variables $X_1, \ldots, X_m$ be i.i.d. samples of $\boldsymbol{r}^T\boldsymbol{u}$. It is not hard to see that

- $||\mathbf{R}\boldsymbol{u}||^2 = X_1^2 + X_2^2 + \cdots + X_m^2$, (one can view X_i as the i-th entry of $\mathbf{R}\boldsymbol{u}$),
- $\mathbb{E}\left[||\mathbf{R}\boldsymbol{u}||^2\right] = m||\boldsymbol{u}||^2$.

Next we claim that for each i, $X_i^2 \leq mB^2\omega(\log m)$ with overwhelming probability. By Hoeffding's inequality, we have

$$\Pr\left[\left|\sum_{j\in[m]} r_j u_j\right| > t\right] < 2e^{\frac{2t^2}{m\cdot 4B^2}}.$$

This is because each $r_j u_j \in [-B, B]$. (Recall that we consider a fixed $\boldsymbol{u}$ for the typical case). By setting $t = \sqrt{m}B\omega(\log m)$, we have $\Pr[|X_i| > t] < \mathsf{negl}(m)$. Thus $X_i^2 \leq mB^2\omega(\log m)$ with overwhelming probability. So we can consider truncated versions of X_i^2's, where we cut out the large samples. This will only induce a negligible statistical distance, and change the expectation by a negligible amount. For simplicity of presentation, we still use the notation X_i^2's in the following arguments, but the reader should keep in mind that they were truncated.

Next again we apply Hoeffding's inequality to the X_i^2's to obtain

$$\Pr\left[\left|||\mathbf{R}\boldsymbol{u}||^2 - m||\boldsymbol{u}||^2\right| > t'\right] < 2e^{-\frac{2t'^2}{\sum_{i=1}^m (mB^2\omega(\log m))^2}} = 2e^{-\frac{2t'^2}{m^3B^4\omega(\log m)}}.$$

By taking $t' = m||\boldsymbol{u}||^2/2$, we have

$$\Pr\left[\left|||\mathbf{R}\boldsymbol{u}||^2 - m||\boldsymbol{u}||^2\right| > t'\right] < 2e^{-\frac{||\boldsymbol{u}||^4}{2mB^4\omega(\log m)}}.$$

Since $\boldsymbol{u}$ is typical, we know that $||\boldsymbol{u}||^2 = \Theta(m\alpha^2)$. Also recall that $B = \alpha\omega(\log m)$. So we have

$$\Pr\left[||\mathbf{R}\boldsymbol{u}||^2 \in \Theta(m^2\alpha^2)\right] > 1 - 2e^{-\frac{m}{\omega(\log m)}} = 1 - \mathsf{negl}(m).$$

This completes the proof. ☐

Using the same argument as above, we can show the following lemma.

Lemma 4.2. *Let $\boldsymbol{u} = (u_1, ..., u_m) \in \mathbb{R}^m$ be an m dimensional Gaussian sample with width α. Then*

$$\Pr\left[||\mathbf{R}_{(\mathbf{t})}\boldsymbol{u}||^2 \in \Theta(tm^2\alpha^2)\right] > 1 - \mathsf{negl}(m),$$

where $\mathbf{R}_{(\mathbf{t})}$ is sampled as follows: first sample t matrices $\mathbf{R}_1, \ldots, \mathbf{R}_t$ at random from $\{-1, 1\}^m$, and then set $\mathbf{R}_{(\mathbf{t})} = \sum_{i=1}^t \mathbf{R}_i$.

Lemma 4.3. *Let $\boldsymbol{z}, \boldsymbol{x}$ be m-dimensional Gaussian distributions with width s, α, respectively, and $\mathbf{R}$ is a $\{-1, 1\}^{m\times m}$ matrix sampled uniformly at random. Then $|\boldsymbol{z}^T\mathbf{R}\boldsymbol{x}| \leq ms\alpha\omega(\log m)$ with overwhelming probability.*

Proof. Let $r_{i,j}$ be the (i, j)-th entry of $\mathbf{R}$, $\boldsymbol{z} = (z_1, \ldots, z_m)$, and $\boldsymbol{x} = (x_1, \ldots, x_m)$. Then $|\boldsymbol{z}^T\mathbf{R}\boldsymbol{x}|$ can be written as $\sum_{i,j\in[m]} \boldsymbol{r}_{i,j}z_ix_j$. Now we argue that for fixed vectors $\boldsymbol{z}, \boldsymbol{x}$, the probability that

$$\left|\sum_{i,j\in[m]} \boldsymbol{r}_{i,j}z_ix_j\right| > \sqrt{\sum_{i,j\in[m]} |z_ix_j|^2} \cdot \omega(\log m)$$

is small.

We observe that each $r_{i,j}z_ix_j$ is an independent random variables taking values between $(-|z_ix_j|, |z_ix_j|)$, and has mean 0. Thus, we can apply Hoeffding's inequality:

$$\Pr\left(\left|\sum_{i,j\in[m]} \boldsymbol{r}_{i,j}z_ix_j\right| > t\right) < 2\exp\left\{-\frac{2t^2}{\sum_{i,j\in[m]}(2|z_ix_j|)^2}\right\}.$$

By taking $t = \sqrt{\sum_{i,j\in[m]} |z_ix_j|^2} \cdot \omega(\log m)$, we have

$$\Pr\left(\sum_{i,j\in[m]} \boldsymbol{r}_{i,j}z_ix_j > t\right) < \mathsf{negl}(m).$$

We know that with overwhelming probability, all (absolute values of) entries of $\boldsymbol{z}$ are less than $s\omega(\log m)$ and all entries in $\boldsymbol{x}$ are less than $\alpha\omega(\log m)$. So we know that with overwhelming probability $|z_i x_j| \leq s\alpha\omega(\log m)$. This is equivalent to saying that with overwhelming probability over the choices of $\boldsymbol{x}, \boldsymbol{z}$, we have $t \leq ms\alpha\omega(\log m)$. This completes the proof. □

5 Multi-Distributional Bideniable IPE

Let λ be the security parameter. Let ℓ be the length of predicate/attribute vectors. Let n, q, m be positive integers. Set $k = \lfloor \log_2 q \rfloor$. Let $\alpha, \beta, \gamma, s \in [0,1]$ be positive real Gaussian parameters. We will use the gadget matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ along with a "good" basis $\mathbf{T}_{\mathbf{G}}$, as introduced in [MP12]. For fixed q as above, recall that the set $\mathbb{Z}_1 \stackrel{\text{def}}{=} \{-1/2 + 1/q, -1/2 + 2/q, \ldots, 1/2 - 1/q, 1/2\}$ is the range $(-1/2, 1/2] \subset \mathbb{R}$ "modulo 1" represented with bit-precision $\log_2(q)$.

Our construction of multi-distributional bideniable encryption for inner product predicates BiDenIPE = (Setup, DenSetup, KeyGen, SampleP, SampleU, TestP, FakeSCoins, FakeRCoins) uses a semantically secure public key encryption $\Pi = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ with message space $\mathcal{M}_\Pi = \mathbb{Z}_q^{m \times m}$ and ciphertext space $\mathcal{C}_\Pi$, and is described as follows:

- $\mathsf{Setup}(1^\lambda, 1^\ell)$: On input security parameter λ and predicate/attribute vector length parameter ℓ, do:
  1. Run $\mathsf{TrapGen}(q, n, m)$ to obtain a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and trapdoor basis $\mathbf{T}_{\mathbf{A}} \subset \Lambda_q^{\perp}(\mathbf{A})$.
  2. Sample $\ell \cdot (1+k)$ uniform matrices $\mathbf{A}_{i,j} \in \mathbb{Z}_q^{n \times m}$ for $i = 1, ..., \ell, j = 0, ..., k$, and a uniform vector $\mathbf{u} \in \mathbb{Z}_q^n$.
  3. Compute a public/secret key pair $(\mathsf{pk}', \mathsf{sk}')$ for a semantically secure public key encryption $(\mathsf{pk}', \mathsf{sk}') \leftarrow \mathsf{Gen}'(1^\lambda)$.
  4. Output public parameters pp and master secret key msk as

$$\mathsf{pp} = (\mathsf{pk}', \mathbf{A}, \{\mathbf{A}_{i,j}\}, \mathbf{u}), \qquad \mathsf{msk} = (\mathbf{T}_{\mathbf{A}}, \mathsf{sk}')$$

- $\mathsf{DenSetup}(1^\lambda, 1^\ell)$: On input security parameter λ and predicate/attribute vector length parameter ℓ, the deniable setup algorithm runs the same computation as setup algorithm, and outputs

$$\mathsf{pp} = (\mathsf{pk}', \mathbf{A}, \{\mathbf{A}_{i,j}\}, \mathbf{u}), \qquad \mathsf{msk} = (\mathbf{T}_{\mathbf{A}}, \mathsf{sk}'), \qquad \mathsf{fk} = (\mathbf{T}_{\mathbf{A}}, \mathsf{sk}')$$

- $\mathsf{Keygen}(\mathsf{pp}, \mathsf{msk}, \mathbf{v})$: On input public parameters pp, master secret key msk, and a predicate vector $\mathbf{v} = (v_1, ..., v_\ell) \in \mathbb{Z}_q^\ell$, do:
  1. For $i = 1, ..., \ell$, decompose v_i into its bit representation as: $v_i = \sum_{j=0}^{k} v_{i,j} \cdot 2^j$, where $v_{i,j} \in \{0, 1\}$.
  2. Define the matrices

$$\mathbf{C}_{\mathbf{v}} = \sum_{i=1}^{\ell} \sum_{j=0}^{k} v_{i,j} \mathbf{A}_{i,j} \in \mathbb{Z}_q^{n \times m}, \qquad \mathbf{A}_{\mathbf{v}} = [\mathbf{A} | \mathbf{C}_{\mathbf{v}}] \in \mathbb{Z}_q^{n \times 2m}.$$

  3. Sample vector $\boldsymbol{z} = (\boldsymbol{z}_0 | \boldsymbol{z}_1)$, using

$$(\boldsymbol{z}_0 | \boldsymbol{z}_1) \leftarrow \mathsf{SampleLeft}(\mathbf{A}, \mathbf{C}_{\boldsymbol{v}}, \mathbf{T}_{\mathbf{A}}, \boldsymbol{u}, sq)$$

  such that $[\mathbf{A} | \mathbf{C}_{\boldsymbol{v}}] \cdot \binom{\boldsymbol{z}_0}{\boldsymbol{z}_1} = \boldsymbol{u}$.

4. Output the secret key $\mathsf{sk}_{\mathbf{v}} = \boldsymbol{z}$.

- $\mathsf{SampleP}(\mathsf{pp}, \mathbf{w})$: On input public parameters pk and attribute vector $\mathbf{w} = (w_1, ..., w_\ell) \in \mathbb{Z}_q^\ell$, do:
  1. Choose a uniformly random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$. Then sample noise vector $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}_1^m, \alpha^2 \mathbf{I}_{m\times m}}$ and noise term $x \leftarrow \mathcal{D}_{\mathbb{Z}_1, \alpha}$.
  2. Let $\mathbf{c}_0 := (\mathbf{A}^T \mathbf{s}/q) + \mathbf{x}$.
  3. For $i = 1, ..., \ell$ and $j = 0, ..., k$, do:
     (a) Sample uniform matrix $\mathbf{R}_{i,j} \in \{-1, 1\}^{m\times m}$.
     (b) Let $\mathbf{c}_{i,j} := ((\mathbf{A}_{i,j} + 2^j w_i \mathbf{G})^T \mathbf{s}/q) + \mathbf{R}_{i,j}^T \mathbf{x}$.
     (c) Use public key encryption to encrypt matrix $\mathbf{R}_{i,j}$, i.e. $\mathbf{S}_{i,j} \leftarrow \mathsf{Enc}'(\mathsf{pk}', \mathbf{R}_{i,j})$.
  4. Let $c' := (\mathbf{u}^T \mathbf{s}/q) + x$.
  5. Output the P-sample $\mathbf{c} = (\mathbf{c}_0, \{\mathbf{c}_{i,j}\}, c', \{\mathbf{S}_{i,j}\})$.

- $\mathsf{SampleU}(\mathsf{pp})$: For $i = 1, ..., \ell$ and $j = 0, ..., k$, let $\mathbf{S}_{i,j} \leftarrow \mathsf{Enc}'(\mathsf{pk}', \mathbf{0}_{m\times m})$, and output $(\{\mathbf{S}_{i,j}\}, \mathbf{c})$ for uniform $\mathbf{c} \in \mathbb{Z}_1^m \times (\mathbb{Z}_1^m)^{\ell\times k+1} \times \mathbb{Z}_1 \times \mathcal{C}_\Pi^{\ell\times k+1}$.

- $\mathsf{TestP}(\mathsf{pp}, \mathsf{sk}_{\mathbf{v}}, \mathbf{c})$: On input public parameters pp, secret key $\mathsf{sk}_{\mathbf{v}} = \boldsymbol{z}$ for predicate vector $\mathbf{v}$, and a purported P-sample $\mathbf{c} = (\mathbf{c}_0, \{\mathbf{c}_{i,j}\}, c') \in \mathbb{Z}_1^m \times (\mathbb{Z}_1^m)^{\ell\times k+1} \times \mathbb{Z}_1$, do:
  1. Define the binary expansion of vector $\mathbf{v}$ as Step 1 in key generation algorithm and compute: $\mathbf{c}_{\mathbf{v}} = \sum_{i=1}^{\ell} \sum_{j=0}^{k} v_{i,j} \mathbf{c}_{i,j}$.
  2. Compute $c = c' - \langle \mathbf{z}, \mathbf{c}^* \rangle \in (-1/2, 1/2]$, where $\mathbf{c}^* = (\mathbf{c}_0 | \mathbf{c}_{\mathbf{v}})$.
  3. Accept $\boldsymbol{c}$ as a valid P-sample if $|c|$ is closer to 0 than $1/4$; otherwise reject $\boldsymbol{c}$.

- $\mathsf{FakeSCoins}(\mathbf{c})$: Simply output the P-sample $\mathbf{c}$ as the randomness r_{Sender}^* that would cause $\mathsf{SampleU}$ to output $\mathbf{c}$.

- $\mathsf{FakeRCoins}(\mathsf{pp}, \mathsf{fk}, \mathbf{c}, \mathbf{v})$: On input the public parameters pp, faking key fk, a ciphertext $\boldsymbol{c}$ and an attribute vector $\boldsymbol{v}$:
  1. If $\langle v, w \rangle \neq 0$, then output $\mathsf{sk}_{\boldsymbol{v}} = \mathsf{Keygen}(\mathsf{msk}, \boldsymbol{v})$.
  2. Otherwise, first parse ciphertext as $\mathbf{c} = (\mathbf{c}_0, \{\mathbf{c}_i\}, c', \{\mathbf{S}_{i,j}\})$, and use algorithm $\boldsymbol{x} \leftarrow \mathsf{Invert}(\mathbf{A}, \mathbf{T}_{\mathbf{A}}, \boldsymbol{c}_0)$. Then for $i = 1, ..., \ell$ and $j = 0, ..., k$, use public key decryption to decrypt $\mathbf{S}_{i,j}$ to get $\mathbf{R}_{i,j} \in \{-1, 1\}^{m\times m}$, i.e. $\mathbf{R}_{i,j} := \mathsf{Dec}'(\mathsf{sk}', \mathbf{S}_{i,j})$. Then sample a properly distributed secret key $\boldsymbol{z}$, using
  $$\boldsymbol{z} \leftarrow \mathsf{SampleLeft}(\mathbf{A}, \mathbf{T}_{\mathbf{A}}, \mathbf{C}_{\boldsymbol{v}}, \boldsymbol{u}, sq)$$
  where matrix $\mathbf{C}_{\mathbf{v}} = \sum_{i=1}^{\ell} \sum_{j=0}^{k} v_{i,j} \mathbf{A}_{i,j} \in \mathbb{Z}_q^{n\times m}$, .
  3. Sample correlation coefficient $\mu \leftarrow \mathcal{D}_\gamma$ and sample correlation vectors to be $\boldsymbol{y}_0 \leftarrow \mathcal{D}_{\mathbb{Z}^m, \beta^2 q^2 \mathbf{I}_{m\times m}}$ and $\boldsymbol{y}_1 \leftarrow (\mu \boldsymbol{x}_{\boldsymbol{v}} + \mathcal{D}_{\mathbb{Z}^m, \mathbf{Q}})\, q$, where $\mathbf{R}_{\boldsymbol{v}} \stackrel{\text{def}}{=} \sum_{i=1}^{\ell} \sum_{j=0}^{k} v_{i,j} \mathbf{R}_{i,j}$, where $\boldsymbol{x}_{\boldsymbol{v}} \stackrel{\text{def}}{=} \mathbf{R}_{\boldsymbol{v}}^T \boldsymbol{x}$, and where
  $$\mathbf{Q} \stackrel{\text{def}}{=} \beta^2 \mathbf{I}_{m\times m} - \gamma^2 \alpha^2 \mathbf{R}_{\boldsymbol{v}}^T \mathbf{R}_{\boldsymbol{v}}. \tag{1}$$

  Recall in order to sample from the (ellipsoidal) distribution $\mathcal{D}_{\mathbb{Z}^m, \mathbf{Q}}$:
  - Sample $\boldsymbol{t}' = (t_1', ..., t_m') \in \mathbb{R}^m$ independently as $t_i' \leftarrow D_1$ for $i \in [m]$.
  - Find the Cholesky decomposition $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$ for some lower triangular matrix $\mathbf{L}$. (This is possible by Lemma 2.6 and our parameter setting.)

- Output the vector $\boldsymbol{t} := \mathbf{L}\boldsymbol{t}'$ as the sample $\boldsymbol{t} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\mathbf{Q}}$.

4. Let $\boldsymbol{y} = [\boldsymbol{y}_0|\boldsymbol{y}_1] \in \mathbb{Z}^{2m}$. Sample and output the faked secret key $\mathsf{sk}'_{\mathbf{v}} = \boldsymbol{z}^*$ as the vector

$$\boldsymbol{z}^* \leftarrow \boldsymbol{y} + \mathsf{SamplePre}(\mathbf{A}, \mathbf{C}_{\boldsymbol{v}}, \mathbf{T}_{\mathbf{A}}, \boldsymbol{z} - \boldsymbol{y}, q\sqrt{s^2 - \beta^2})$$

where $\mathbf{A}_{\mathbf{v}} = [\mathbf{A}|\mathbf{C}_{\mathbf{v}}] \in \mathbb{Z}_q^{n\times 2m}$.

5.1 Correctness and Security Proof

Theorem 5.1. *Assuming the hardness of extended-*$\mathsf{LWE}_{q,\beta'}$ *and semantically secure public key encryption* $\Pi = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$, *the above algorithms form a secure attribute-based bitranslucent set scheme regarding Definition 2.2.*

Proof. Lemma 5.2 below shows the correctness property. The indistinguishability property follows directly by Lemma 2.8. The bi-deniability property is proven in Lemma 5.3 below. □

Lemma 5.2. *For parameters specified in Section 5.2, the IP-BTS defined above satisfies the correctness property in Definition 2.2.*

Proof. As we mentioned in Remark 2.3, the correctness of faking algorithms is implied by the bi-deniability property. Therefore, we only need to prove the correctness of normal decryption algorithm. For inner product $\langle \boldsymbol{v}, \boldsymbol{w}\rangle = 0$, we have

$$\begin{aligned}
\boldsymbol{c}_{\boldsymbol{v}} &= \sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}\boldsymbol{c}_{i,j} = \sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}((\mathbf{A}_{i,j} + 2^j w_i\mathbf{G})^T\boldsymbol{s}/q + \mathbf{R}_{i,j}^T\boldsymbol{x}) \\
&= (\sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}\mathbf{A}_{i,j})^T\boldsymbol{s}/q + \langle \boldsymbol{v}, \boldsymbol{w}\rangle\mathbf{G}^T\boldsymbol{s}/q + \sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}(\mathbf{R}_{i,j}^T\boldsymbol{x}) \\
&= (\sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}\mathbf{A}_{i,j})^T\boldsymbol{s}/q + \sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}(\mathbf{R}_{i,j}^T\boldsymbol{x})
\end{aligned}$$

Then we set $\boldsymbol{c}^* = (\boldsymbol{c}_0|\boldsymbol{c}_{\boldsymbol{v}})$, which can be parsed as follows:

$$\begin{aligned}
\boldsymbol{c}^* = (\boldsymbol{c}_0|\boldsymbol{c}_{\boldsymbol{v}}) &= [\mathbf{A}|\sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}\mathbf{A}_{i,j}]^T\boldsymbol{s}/q + [\boldsymbol{x}|\sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}\mathbf{R}_{i,j}^T\boldsymbol{x}] \\
&= \mathbf{A}_{\boldsymbol{v}}^T\boldsymbol{s}/q + [\boldsymbol{x}|\sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}\mathbf{R}_{i,j}^T\boldsymbol{x}]
\end{aligned}$$

Recall that secret key $\mathsf{sk}_{\boldsymbol{v}} = \boldsymbol{z}$ satisfying $\mathbf{A}_{\boldsymbol{v}}\boldsymbol{z} = \boldsymbol{u}$, then for $c = c' - \langle \boldsymbol{z}/q, \boldsymbol{c}^*\rangle$, it holds that

$$\begin{aligned}
c = c' - \langle \boldsymbol{z}, \boldsymbol{c}^*\rangle &= (\boldsymbol{u}^T\boldsymbol{s}/q + x) - \boldsymbol{u}^T\boldsymbol{s}/q - \boldsymbol{z}/q[\boldsymbol{x}|\sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}\mathbf{R}_{i,j}^T\boldsymbol{x}] \\
&= x - \langle \boldsymbol{z}, [\boldsymbol{x}|\sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}\mathbf{R}_{i,j}^T\boldsymbol{x}]\rangle
\end{aligned}$$

Now we want to calculate a bound for the final noise term. To do so, we apply Lemma 4.3 over the $\sum_{i=1}^{\ell}\sum_{j=0}^{k}$ to obtain the correctness constraint for evaluated noise

$$2\ell\log(q)ms\alpha\omega(\log(m)) < 1/4.$$

So by setting the parameters appropriately, as in Section 5.2, we have that

$$|x - \langle \boldsymbol{z}, [\boldsymbol{x}|\sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}\mathbf{R}_{i,j}^T\boldsymbol{x}]\rangle| < 1/4,$$

and the lemma follows. □

Lemma 5.3. *Assuming the hardness of extended-*$\mathsf{LWE}_{q,\beta'}$ *and semantically secure public key encryption* $\Pi = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$, *the IP-BTS scheme described above is bi-deniable in Definition 2.2.*

Proof. First, we notice that because $\mathsf{SampleU}$ simply outputs its random coins as a uniformly random $\boldsymbol{c} \in \mathbb{Z}_1^m \times (\mathbb{Z}_1^m)^{\ell\times k+1} \times \mathbb{Z}_1 \times \mathcal{C}_{\Pi}^{\ell\times k+1}$, we can use $\boldsymbol{c}$ as the coins.

We prove the bi-deniability property by a sequence of hybrids H_i with details as follows:

Hybrid H_0: Hybrid H_0 is the view of adversary $\mathcal{A}$ in the right-hand faking experiment in the definition of IP-BTS bi-deniability. We use the fact that algorithm Invert successfully recovers noise vector $\boldsymbol{x}$ from $\boldsymbol{c}$ with overwhelming probability over all randomness in the experiment.

Hybrid H_1: In hybrid H_1, we will embed matrices $\mathbf{R}_{i,j}$ and vector $\boldsymbol{w}$ in the public parameters pp.

Recall that in hybrid H_0, the matrices $\{\mathbf{A}_{i,j}\}_{i\in[\ell],j\in[k]}$ are sampled at random for each ciphertext. In hybrid H_1, we will modify this as follows: Let $\boldsymbol{w}^* = (w_1^*, ..., w_\ell^*)$ be the challenge attribute vector that adversary $\mathcal{A}$ intends to attack. We sample random matrices $\mathbf{R}_{i,j}^* \in \{-1,1\}^{m\times m}$ for $i \in [\ell], j \in [k]$, which will also be used in the generation of challenge ciphertext, and set the matrices $\{\mathbf{A}_{i,j}\}_{i\in[\ell],j\in[k]}$ to be

$$\mathbf{A}_{i,j} = \mathbf{A}\mathbf{R}_{i,j}^* - 2^j w_i^*\mathbf{G}$$

where matrix $\mathbf{G}$ is the gadget matrix with short trapdoor $\mathbf{T_G}$. The rest of the hybrid is unchanged.

Hybrid H_2: In hybrid H_2, we switch the ciphertexts $\mathbf{S}_{i,j}$ to encryptions of zero.

Recall that in hybrid H_1, we encrypt the randomness matrix $\mathbf{R}_{i,j}^*$ for $i = 1, ..., \ell, j = 0, ..., k$ using semantically secure PKE Π, i.e. $\mathbf{S}_{i,j} \leftarrow \mathsf{Enc}'(\mathsf{pk}', \mathbf{R}_{i,j}^*)$. In hybrid H_2, we just set $\mathbf{S}_{i,j} = \mathsf{Enc}'(\mathsf{pk}', \mathbf{0})$ to be encryption of zero matrix $\mathbf{0} \in \mathbb{Z}^{m\times m}$ to replace the encryptions of matrices $\mathbf{R}_{i,j}^*$.

Hybrid H_3: In hybrid H_3, we change the order of how we generate $\mathbf{A}, \boldsymbol{u}$ in the public parameters pp, and the generation of challenge secret key $\boldsymbol{z}^*$.

Let $\mathbf{A}$ be a random matrix in $\mathbb{Z}_q^{n\times m}$. The construction of $\{\mathbf{A}_{i,j}\}_{i\in[\ell],j\in[k]}$ remains the same as hybrid H_1. Sample error vector $\boldsymbol{x}^* \in \mathcal{D}_{\mathbb{Z}^m,\alpha^2\mathbf{I}_{m\times m}}$ that would be used in algorithm $\mathsf{SampleP}$ later and compute evaluated error $\boldsymbol{x}_{\boldsymbol{v}^*}^* = \sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}^*\mathbf{R}_{i,j}^* \cdot \boldsymbol{x}^*$, where $v_i^* = \sum_{j=0}^{k} v_{i,j}^* \cdot 2^j$. Set vectors $\boldsymbol{y}_0 \leftarrow \mathcal{D}_{\mathbb{Z}^m,\beta^2q^2\mathbf{I}_{m\times m}}$ and $\boldsymbol{y}_1$ as the same way in $\mathsf{FakeRCoins}$ algorithm, i.e. $\boldsymbol{y}_1 \leftarrow \mu q\boldsymbol{x}_{\boldsymbol{v}^*}^* + \mathcal{D}_{\mathbb{Z}^m,\mathbf{Q}}$, and $\boldsymbol{z}^* \leftarrow \boldsymbol{y} + \mathcal{D}_{\mathbb{Z}^{2m}-\boldsymbol{y},(s^2-\beta^2)q^2\mathbf{I}_{m\times m}}$. Then set matrix $\mathbf{A}_{\boldsymbol{v}^*} = [\mathbf{A}|\sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}^*\mathbf{A}_{i,j}]$ and set $\boldsymbol{u} = \mathbf{A}_{\boldsymbol{v}^*} \cdot \boldsymbol{z}^*$. Moverover, since $\mathbf{A}$ is a random matrix, which means we do not have the trapdoor of $\mathbf{A}$ to answer the key queries for predicate vector $\boldsymbol{v}$, we will use the trapdoor $\mathbf{T_G}$ to answer key queries. Consider a secret key query for predicate vector $\boldsymbol{v}$, such that $\langle \boldsymbol{v}, \boldsymbol{w}^* \rangle \neq 0$. To respond, we first decompose v_i^* to its bit expression $v_i^* = \sum_{j=0}^{k} v_{i,j} \cdot 2^j$ for $i = 1, ..., \ell$, and set

$$\mathbf{A}_{\boldsymbol{v}^*} = [\mathbf{A}|\sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}^* \mathbf{A}_{i,j}] = [\mathbf{A}|\mathbf{A}(\sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}^* \mathbf{R}_{i,j}^*) - \langle \boldsymbol{v}, \boldsymbol{w}^* \rangle \mathbf{G}]$$

Then sample $\mathsf{sk}_{\boldsymbol{v}} = \boldsymbol{z}$, using

$$\boldsymbol{z} = \mathsf{SampleRight}(\mathbf{A}, (\sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}^* \mathbf{R}_{i,j}^*), \langle \boldsymbol{v}, \boldsymbol{w}^* \rangle \mathbf{G}, \mathbf{T}_{\mathbf{G}}, \boldsymbol{u}, sq)$$

To answer P-sample queries, $\mathsf{SampleP}$ is the same as hybrid H_1 except using error vectors $\boldsymbol{x}^*$ and matrix $\mathbf{G}$. It first computes and outputs $\boldsymbol{c}^* = (\boldsymbol{c}_0^*, \{\boldsymbol{c}_{i,j}^*\}, c^{*'})$, i.e. $\boldsymbol{c}_0^* = \mathbf{A}^T \boldsymbol{s}/q + \boldsymbol{x}^*, \boldsymbol{c}_{i,j}^* = \mathbf{R}_{i,j}^{*T}(\mathbf{A}^T \boldsymbol{s}/q + \boldsymbol{x}^*), c^{*'} = (\langle \mathbf{u}, \mathbf{s} \rangle/q) + x^*$, then for $i = 1, .., \ell, j = 0, ..., k$, encrypts matrix $\mathbf{S}_{i,j} \leftarrow \mathsf{Enc}'(\mathbf{R}_{i,j}^*, \mathsf{pk}')$ using semantically secure public key encryption Π. For faking receiver coins algorithm, $\mathsf{FakeRCoins}$, simply output the vector $\boldsymbol{z}^*$ pre-sampled in the generation of vector $\boldsymbol{u}$ before.

Hybrid H_4: In hybrid H_4, we change the order in which we generate vector $\boldsymbol{y}$ and error vector $\boldsymbol{x}^*$.

First, we directly sample the $2m$-dimensional correlation vector $\boldsymbol{y} := (\boldsymbol{y}_0|\boldsymbol{y}_1) \leftarrow \mathcal{D}_{\mathbb{Z}^{2m}, \beta^2 q^2 \mathbf{I}_{2m \times 2m}}$ at once. (From $\boldsymbol{y}$, we compute $\boldsymbol{z}^*$ as in previous hybrids.) Next, we generate $\boldsymbol{c}_0^*$'s error term as $\boldsymbol{x}^* := \nu \mathbf{R}_{\boldsymbol{v}}^* \boldsymbol{y}_1/q + \mathcal{D}_{\mathbb{Z}^m, \mathbf{Q}'}$, where $\nu \leftarrow \mathcal{D}_\tau$, $\tau \stackrel{\text{def}}{=} \gamma\alpha^2/\beta^2$ and $\mathcal{D}_{\mathbb{Z}^m, \mathbf{Q}'}$ is sampled as $\mathbf{L}'\mathcal{D}_{\mathbb{Z}_1^m, \mathbf{I}_{m \times m}}$ for

$$\mathbf{Q}' = \mathbf{L}'\mathbf{L}'^T \stackrel{\text{def}}{=} \alpha^2 \mathbf{I}_{m \times m} - \tau^2 \beta^2 \mathbf{R}_{\boldsymbol{v}}^* \mathbf{R}_{\boldsymbol{v}}^{*T}. \quad (2)$$

Additionally, we modify the challenge ciphertext to be

$$\boldsymbol{c}_0^* = \mathbf{A}^T \boldsymbol{s}/q + \boldsymbol{x}^*, \quad \boldsymbol{c}_{i,j}^* = \mathbf{R}_{i,j}^{*T} \boldsymbol{c}_0^*/q, \quad c^{*'} = \langle \boldsymbol{u}, \boldsymbol{s} \rangle/q + \mathcal{D}_{\mathbb{Z},\alpha}$$

Observe that this induces an evaluated error term during decryption of the challenge ciphertext under secret keys $\mathsf{sk}_{\boldsymbol{v}}$ of the form $\boldsymbol{x}_{\boldsymbol{v}}^* = \mathbf{R}_{\boldsymbol{v}}^{*T} \boldsymbol{x}^* = \nu \mathbf{R}_{\boldsymbol{v}}^{*T} \mathbf{R}_{\boldsymbol{v}}^* \boldsymbol{y}_1/q + \mathbf{R}_{\boldsymbol{v}}^{*T} \mathcal{D}_{\mathbb{Z}^m, \mathbf{Q}'}$.

Hybrid H_5: In hybrid H_5, we change the order in which we generate secret key $\boldsymbol{z}^*$ and vector $\boldsymbol{y}$.

First, we directly sample the $2m$-dimensional secret key $\boldsymbol{z}^* = (\boldsymbol{z}_0^*|\boldsymbol{z}_1^*) \leftarrow \mathcal{D}_{\mathbb{Z}_q^{2m}, s^2 q^2 \mathbf{I}_{m \times m}}$. (This determines $\mathsf{sk}_{\boldsymbol{v}^*}$ and vector $\boldsymbol{u}$ in pp.) Next, we generate the correlation vector as $\boldsymbol{y} = (\boldsymbol{y}_0|\boldsymbol{y}_1) := \boldsymbol{z}^*/2 + \mathcal{D}_{\mathbb{Z}^{2m}, (\beta^2 - s^2/4) q^2 \mathbf{I}_{2m \times 2m}}$. The remainder of the hybrid remains roughly the same. In particular, the challenge ciphertext $\boldsymbol{c}^*$ (and its noise term $\boldsymbol{x}^*$) is generated from $\mathbf{y}$ in the same manner as Hybrid H_4. We break the noise term $\boldsymbol{x}^*$ into two terms $\boldsymbol{x}^* = \boldsymbol{x}^{(1)} + \boldsymbol{x}^{(2)} + \nu \mathbf{R}_{\boldsymbol{v}}^* \boldsymbol{y}_1/q$, where $\boldsymbol{x}^{(1)} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \beta'^2 \mathbf{I}_{m \times m}}$ and $\boldsymbol{x}^{(2)} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \mathbf{Q}' - \beta'^2 \mathbf{I}_{m \times m}}$. We set $\beta' = \alpha/2$

Hybrid H_6: In hybrid H_6, we change how the challenge ciphertext is generated using Extended-LWE$^+$.

First, sample uniformly random vector $\boldsymbol{b} \in \mathbb{Z}_q^m$ and set the challenge ciphertext as

$$\boldsymbol{c}_0^* = \boldsymbol{b}/q + \boldsymbol{x}^{(2)} + \nu \mathbf{R}_{\boldsymbol{v}}^* \boldsymbol{y}_1/q, \quad \boldsymbol{c}_{i,j}^* = \mathbf{R}_{i,j}^{*T} \boldsymbol{c}_0^*, \quad c^{*'} = \boldsymbol{z}^{*T}[\mathbf{I}|\mathbf{R}_{\boldsymbol{v}}^*]^T(\boldsymbol{b}/q - \boldsymbol{x}^{(1)}) + D_{\mathbb{Z}_1, \alpha}$$

where matrix $\mathbf{R}_{\boldsymbol{v}}^* = \sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}^* \mathbf{R}_{i,j}^*$ and vectors $\boldsymbol{x}_{i,j}$ are sampled as in H_4.

Hybrid H_7: In hybrid H_7, we change the challenge ciphertext to be uniformly random. That is, $\mathsf{SampleP}$ samples uniform vectors $\boldsymbol{c}_0^* \in \mathbb{Z}_1^m, \boldsymbol{c}_{i,j}^* \in \mathbb{Z}_1^m, c^{*'} \in \mathbb{Z}_1$ and outputs ciphertext $\boldsymbol{c}^* = (\boldsymbol{c}_0^*, \{\boldsymbol{c}_{i,j}^*\}, c^{*'})$.

Claim 5.4. *Hybrids* H_0 *and* H_1 *are statistically indistinguishable.*

Proof. Observe the only difference between hybrids H_0 and H_1 is the generation of matrices $\{\mathbf{A}_{i,j}\}_{i\in[\ell],j\in[k]}$, i.e. $\mathbf{A}_{i,j} = \mathbf{A}\mathbf{R}^*_{i,j} - 2^j w^*_i \mathbf{G}$, where matrix $\mathbf{G}$ is the gadget matrix with short trapdoor $\mathbf{T}_{\mathbf{G}}$ and $\mathbf{R}^*_{i,j} \xleftarrow{\$} \{-1,1\}^{m\times m}$. Then by Leftover Hash Lemma 2.7, the distribution $(\mathbf{A}, \{\mathbf{A}\mathbf{R}^*_{i,j}\}_{i\in[\ell],j\in[k]})$ is statistically close to the distribution $(\mathbf{A}, \{\mathbf{A}_{i,j}\}_{i\in[\ell],j\in[k]})$, where matrices $\mathbf{A}_{i,j}$ are uniformly random over $\mathbb{Z}^{m\times m}$. Hence, hybrid H_0 and H_1 are statistically indistinguishable. ☐

Claim 5.5. *Assuming the semantic security of PKE* $\Pi = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$, *hybrids* H_1 *and* H_2 *are computationally indistinguishable.*

Proof. Observe there is only one difference between hybrids H_1 and H_2: In the challenge ciphertext, the encryptions (under PKE Π) of the random rotation matrices $\mathbf{R}^*_{i,j}$ are replaced by encryptions of 0. If an efficient adversary $\mathcal{A}$ distinguishes between the H_1-encryptions of $\mathbf{R}^*_{i,j}$ and the H_2-encryptions of 0 with non-negligible probability, then we can construct an efficient reduction $\mathcal{B}$ that uses $\mathcal{A}$ to break the semantic security of Π with similar probability. ☐

Claim 5.6. *Hybrids* H_2 *and* H_3 *are statistically indistinguishable.*

Proof. Observe there are three differences between hybrid H_2 and H_3: The generation of matrices $\mathbf{A}, \mathbf{D}$, the generation of challenge secret key $\mathsf{sk}_{\boldsymbol{v}^*}$ and the computation method to answer secret key queries. By the property of algorithm $\mathsf{TrapGen}$ in Lemma 2.8, the distribution of matrix $\mathbf{A}$ in hybrid H_2 is statistically close to uniform distribution from which matrix $\mathbf{A}$ in hybrid H_3 is sampled.

For secret key queries, in hybrid H_2, we sample vector $\boldsymbol{z} = (\boldsymbol{z}_0|\boldsymbol{z}_1)$, using

$$\boldsymbol{z} = (\boldsymbol{z}_0|\boldsymbol{z}_1) \leftarrow \mathsf{SampleLeft}(\mathbf{A}, \mathbf{C}_{\boldsymbol{v}}, \mathbf{T}_{\mathbf{A}}, \boldsymbol{u}, sq)$$

While in hybrid H_3, we sample vector $\boldsymbol{z} = (\boldsymbol{z}_0|\boldsymbol{z}_1)$, using

$$\boldsymbol{z} = \mathsf{SampleRight}(\mathbf{A}, (\sum_{i=1}^{\ell}\sum_{j=0}^{k} v^*_{i,j}\mathbf{R}^*_{i,j}), \langle \boldsymbol{v}, \boldsymbol{w}^*\rangle \mathbf{G}, \mathbf{T}_{\mathbf{G}}, \boldsymbol{u}, sq)$$

By setting the parameters appropriately as specified in Section 5.2 and the properties of algorithms $\mathsf{SampleLeft}$ and $\mathsf{SampleRight}$ in Lemma 2.9, the secret key answers to queries are statistically close.

By Leftover Hash Lemma, the distribution $([\mathbf{A}|\mathbf{C}_{\boldsymbol{v}^*}], [\mathbf{A}|\mathbf{C}_{\boldsymbol{v}^*}]\cdot \boldsymbol{z}^*)$ and $([\mathbf{A}|\mathbf{C}_{\boldsymbol{v}^*}], \boldsymbol{u})$, where matrix $\mathbf{C}_{\mathbf{v}^*} = \sum_{i=1}^{\ell}\sum_{j=0}^{k} v^*_{i,j}\mathbf{A}_{i,j} \in \mathbb{Z}_q^{n\times m}$, are statistically close, which means matrix $\boldsymbol{u}$ in both hybrids are statistically close. ☐

Claim 5.7. *Hybrids* H_3 *and* H_4 *are statistically identical.*

Proof. The only difference between the two experiments in the choice of $\boldsymbol{x}^*$ and $\boldsymbol{y}$ – in particular, the choice of the $\boldsymbol{y}_1$ component of $\boldsymbol{y} = (\boldsymbol{y}_0|\boldsymbol{y}_1)$. We will show that the joint distribution of $(\boldsymbol{x}^*, \boldsymbol{y}_1) \in (\mathbb{Z}^m)^2$ is identically distributed between the two experiments:

In Hybrid H_3, $\boldsymbol{y}_1$ is sampled as $\boldsymbol{y}_1 \leftarrow (\mu \boldsymbol{x}^*_{\boldsymbol{v}} + \mathcal{D}_{\mathbb{Z}^m,\mathbf{Q}})q$ where $\mathbf{Q} = \beta^2\mathbf{I}_{m\times m} - \gamma^2\alpha^2\mathbf{R}^{*T}_{\boldsymbol{v}}\mathbf{R}^*_{\boldsymbol{v}}$ with $\boldsymbol{x}^* \leftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha^2\mathbf{I}_{m\times m}}$ and $\boldsymbol{x}^*_{\boldsymbol{v}} = \sum_{i=1}^{\ell}\sum_{j=0}^{k} v_{i,j}\mathbf{R}^{*T}_{i,j}\boldsymbol{x}^* = \mathbf{R}^{*T}_{\boldsymbol{v}}\boldsymbol{x}^*$. Therefore in H_3, we may write the joint distribution of $(\boldsymbol{x}^*, \boldsymbol{y}_1)$ as $\mathbf{T}_1 \cdot \mathcal{D}_{\mathbb{Z}^{2m},\mathbf{I}_{2m\times 2m}}$, where $\mathbf{T}_1 \stackrel{\text{def}}{=} \begin{pmatrix} \alpha\mathbf{I}_{m\times m} & \mathbf{0}_{m\times m} \\ \gamma\alpha q\mathbf{R}^{*T}_{\boldsymbol{v}} & \mathbf{L} \end{pmatrix}$ for $\mathbf{Q} = \mathbf{L}\mathbf{L}^T \in \mathbb{Z}^{m\times m}$ via the Cholesky decomposition due to Lemma 2.6.

In Hybrid H_4, $\boldsymbol{y} = (\boldsymbol{y}_0|\boldsymbol{y}_1)$ is sampled as $\boldsymbol{y} \leftarrow \mathcal{D}_{\mathbb{Z}^{2m},\beta^2q^2\mathbf{I}_{m\times m}}$. Then, $\boldsymbol{x}^*$ is generated as $\boldsymbol{x}^* = \nu\mathbf{R}^*_{\boldsymbol{v}}\mathbf{y}_1/q + \mathcal{D}_{\mathbb{Z}^m,\mathbf{Q}'}$ where $\nu \leftarrow \mathcal{D}_\tau$, $\tau \stackrel{\text{def}}{=} \gamma\alpha^2/\beta^2$ and $\mathbf{Q}' = \alpha^2\mathbf{I}_{m\times m} - \tau^2\beta^2\mathbf{R}^*_{\boldsymbol{v}}\mathbf{R}^{*T}_{\boldsymbol{v}}$. Therefore, in H_4, we may write the joint distribution of $(\boldsymbol{x}^*, \boldsymbol{y}_1)$ as $\mathbf{T}_2 \cdot \mathcal{D}_{\mathbb{Z}^{2m},\mathbf{I}_{2m\times 2m}}$, where $\mathbf{T}_2 \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{L}' & \tau\beta\mathbf{R}^*_{\boldsymbol{v}} \\ \mathbf{0}_{m\times m} & \beta q \mathbf{I}_{m\times m} \end{pmatrix}$ for $\mathbf{Q}' = \mathbf{L}'\mathbf{L}'^T \in \mathbb{Z}^{m\times m}$ via the Cholesky decomposition due to Lemma 2.6.

We claim equality of the following systems of equations:

$$\mathbf{T}_1\mathbf{T}_1^T = \begin{pmatrix} \alpha^2\mathbf{I}_{m\times m} & \gamma\alpha^2 q\mathbf{R}^*_{\boldsymbol{v}} \\ \gamma\alpha^2 q\mathbf{R}^{*T}_{\boldsymbol{v}} & \gamma^2\alpha^2q^2\mathbf{R}^{*T}_{\boldsymbol{v}}\mathbf{R}^*_{\boldsymbol{v}} + \mathbf{L}\mathbf{L}^T \end{pmatrix} = \begin{pmatrix} \mathbf{L}'\mathbf{L}'^T + \tau^2\beta^2\mathbf{R}^*_{\boldsymbol{v}}\mathbf{R}^{*T}_{\boldsymbol{v}} & \tau\beta^2 q\mathbf{R}^*_{\boldsymbol{v}} \\ \tau\beta^2 q\mathbf{R}^{*T}_{\boldsymbol{v}} & \beta^2q^2\mathbf{I}_{m\times m} \end{pmatrix} = \mathbf{T}_2\mathbf{T}_2^T.$$

This fact may be seen quadrant-wise by our choice of $\tau = \gamma\alpha^2/\beta^2$ and the settings of $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$ and $\mathbf{Q}' = \mathbf{L}'\mathbf{L}'^T$ in Equations (1) and (2). It then follows that $(\mathbf{T}_2^{-1}\mathbf{T}_1)(\mathbf{T}_2^{-1}\mathbf{T}_1)^T = \mathbf{I}_{2m\times 2m}$, implying $\mathbf{T}_1 = \mathbf{T}_2\mathbf{Q}^*$ for some orthogonal matrix $\mathbf{Q}^*$. Because the spherical Gaussian $\mathcal{D}_{\mathbb{Z}^{2m},\mathbf{I}_{2m\times 2m}}$ is invariant under rigid transformations, we have $\mathbf{T}_1 \cdot \mathcal{D}_{\mathbb{Z}^{2m},\mathbf{I}_{2m\times 2m}} = \mathbf{T}_2\mathbf{Q}^* \cdot \mathcal{D}_{\mathbb{Z}^{2m},\mathbf{I}_{2m\times 2m}} = \mathbf{T}_2 \cdot \mathcal{D}_{\mathbb{Z}^{2m},\mathbf{I}_{2m\times 2m}}$, and the claim follows. $\square$

Claim 5.8. *Hybrids* H_4 *and* H_5 *are statistically indistinguishable.*

Proof. Observe the main difference between hybrids H_4 and H_5 is the order of generation of vectors $\boldsymbol{y}$ and $\boldsymbol{z}^*$: In the hybrid H_4, we first sample $\boldsymbol{y} = (\boldsymbol{y}_0|\boldsymbol{y}_1) \leftarrow \mathcal{D}_{\mathbb{Z}^{2m},\beta^2q^2\mathbf{I}_{2m\times 2m}}$ and set $\boldsymbol{z}^* \leftarrow \boldsymbol{y} + \mathcal{D}_{\mathbb{Z}^{2m}-\boldsymbol{y},q^2(s^2-\beta^2)\mathbf{I}_{2m\times 2m}}$, while in hybrid H_5, we first sample $\boldsymbol{z}^* \leftarrow \mathcal{D}_{\mathbb{Z}^{2m},s^2q^2\mathbf{I}_{2m\times 2m}}$, and set $\boldsymbol{y} = (\boldsymbol{y}_0|\boldsymbol{y}_1) := \boldsymbol{z}^*/2 + \mathcal{D}_{\mathbb{Z}^{2m},(\beta^2-s^2/4)q^2\mathbf{I}_{2m\times 2m}}$. By setting parameters appropriately as in Section 5.2, these two distributions are statistically close. $\square$

Claim 5.9. *Assuming the hardness of extended-*$\mathsf{LWE}^+_{n,m,q,D_{\mathbb{Z}^m,\beta'},\mathbf{R}}$ *for any adversarially chosen distribution over matrices* $\mathbf{R} \in \mathbb{Z}_q^{m\times m}$*, then hybrids* H_5 *and* H_6 *are computationally indistinguishable.*

Proof. Suppose $\mathcal{A}$ has non-negligible advantage in distinguishing hybrid H_5 and H_6, then we use $\mathcal{A}$ to construct an extended-LWE$^+$ algorithm $\mathcal{B}$ as follows:

Invocation. $\mathcal{B}$ invokes adversary $\mathcal{A}$ to commit to a challenge attribute vector $\boldsymbol{w}^* = (w_1^*, ..., w_\ell^*)$ and challenge predicate vector $\boldsymbol{v}^* = (v_1^*, ..., v_\ell^*)$. Then $\mathcal{B}$ specifies $\mathbf{R}$ by sampling $\mathbf{R}^*_{i,j}$ as in the hybrids, and sets $\mathbf{R} = \mathbf{R}^*_v$. Then it receives an extended-LWE$^+$ instance for the matrix $\mathbf{R} = \mathbf{R}^*_v$ as follows:

$$\{\mathbf{A}, \boldsymbol{b} = \mathbf{A}\boldsymbol{s} + \boldsymbol{x}, \boldsymbol{z}_0, \boldsymbol{z}_1, \langle \boldsymbol{z}_0, \boldsymbol{b} - \boldsymbol{x}\rangle + x, \langle \mathbf{R}\boldsymbol{z}_1, \boldsymbol{b} - \boldsymbol{x}\rangle + x'\}$$

where $\mathbf{A} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^{n\times m}$, $\boldsymbol{s} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^n$, $\boldsymbol{u} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^m$, $\boldsymbol{x}, \boldsymbol{z}_0, \boldsymbol{z}_1 \stackrel{\$}{\leftarrow} \chi^n$ and $x, x' \stackrel{\$}{\leftarrow} \chi$. Algorithm $\mathcal{B}$ aims to leverage adversary $\mathcal{A}$'s output to solve the extended-LWE$^+$ assumption.

Setup. $\mathcal{B}$ generates matrices $\{\mathbf{A}_{i,j}\}_{i\in[\ell],j\in[k]}$ as specified in hybrid H_1. Then, $\mathcal{B}$ sets challenge secret key $\mathsf{sk}_{\boldsymbol{v}^*} = \boldsymbol{z}^* = (\boldsymbol{z}_0^*|\boldsymbol{z}_1^*) = (\boldsymbol{z}_0|\boldsymbol{z}_1)$ from extended-LWE$^+$ instance and computes vector $\boldsymbol{u}$ as in hybrid H_5.

Secret key queries. $\mathcal{B}$ answers adversary $\mathcal{A}$'s secret key queries as in hybrid H_2.

Challenge ciphertext. $\mathcal{B}$ answers adversary $\mathcal{A}$'s P-sample query by setting

$$\boldsymbol{c}_0^* = \boldsymbol{b}/q + \boldsymbol{x}^{(2)} + \nu\mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{y}_1/q, \quad \boldsymbol{c}^*_{i,j} = \mathbf{R}^{*T}_{i,j}\boldsymbol{c}_0^*, \quad c^{*'} = \boldsymbol{u}^T\boldsymbol{s}/q + \mathcal{D}_{\mathbb{Z}_1,\alpha}$$

Faking receiver coin query. $\mathcal{B}$ answers adversary $\mathcal{A}$'s faking receiver coin query by outputting the extended-LWE instance's vector $\mathsf{sk}_{\boldsymbol{v}^*} = \boldsymbol{z}^*$.

Output. $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs.

We can rewrite the expression of $c^{*'}$ to be

$$\begin{aligned}
c^{*'} &= ([\mathbf{A}^*|\mathbf{A}^*\mathbf{R}^*_{\boldsymbol{v}}]\binom{\boldsymbol{z}_0^*}{\boldsymbol{z}_1^*})^T\boldsymbol{s}/q + \mathcal{D}_{\mathbb{Z}_1,\alpha} \\
&= ((\boldsymbol{z}_0^*|\boldsymbol{z}_1^*)\binom{\mathbf{A}^{*T}}{\mathbf{R}^{*T}_{\boldsymbol{v}}\mathbf{A}^{*T}})\boldsymbol{s}/q + D_{\mathbb{Z}_1,\alpha} = \boldsymbol{z}_0^*\mathbf{A}^{*T}\boldsymbol{s}/q + \boldsymbol{z}_1^*\mathbf{R}^{*T}_{\boldsymbol{v}}\mathbf{A}^{*T}\boldsymbol{s}/q + \mathcal{D}_{\mathbb{Z}_1,\alpha} \\
&= \langle \boldsymbol{z}_0^*, \boldsymbol{b}/q - \boldsymbol{x}^{(1)}\rangle + \langle \mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{z}_1^*, \boldsymbol{b}/q - \boldsymbol{x}^{(1)}\rangle + \mathcal{D}_{\mathbb{Z}_1,\alpha}.
\end{aligned}$$

We can see that if the eLWE^+ instance's vector $\boldsymbol{b}$ is pseudorandom, then the distribution simulated by $\mathcal{B}$ is exactly the same as H_5. If $\boldsymbol{b}$ is truly random and independent, then the distribution simulated by $\mathcal{B}$ is exactly the same as H_6. Therefore, if $\mathcal{A}$ can distinguish H_5 from H_6 with non-negligible probability, then $\mathcal{B}$ can break the $\mathsf{eLWE}^+_{n,m,q,\mathcal{D}_{(\alpha/2)q},\alpha',\mathbf{R}^*_{\boldsymbol{v}}}$ problem for some $\alpha' \geq 0$ with non-negligible probability. ☐

Claim 5.10. *Hybrids* H_6 *and* H_7 *are statistically indistinguishable.*

Proof. The only difference in these two hybrids is the choice of $(\boldsymbol{c}_0^*, \boldsymbol{c}_{i,j}^*, c^{*'})$. In hybrid H_6, we first observe that $\boldsymbol{c}_0^*$ is uniformly random, so $\mathbf{R}^{*T}_{i,j}(\boldsymbol{b}/q + \boldsymbol{x}^{(2)})$ is also uniformly random for each i, j, by the leftover hash lemma (Lemma 2.7) and our setting of parameters. Therefore, $(\boldsymbol{c}_0^*, \boldsymbol{c}_{i,j}^*)$ are uniformly random (in their marginal distributions). Thus, it remains to show that that $c^{*'}$ is still uniformly random even conditioned on fixed samples of $(\boldsymbol{c}_0^*, \boldsymbol{c}_{i,j}^*)$.

As calculated above, we have the following expression:

$$c^{*'} = \langle \boldsymbol{z}_0^*, \boldsymbol{b}/q - \boldsymbol{x}^{(1)}\rangle + \langle \mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{z}_1^*, \boldsymbol{b}/q - \boldsymbol{x}^{(1)}\rangle + \mathcal{D}_{\mathbb{Z}_1,\alpha}.$$

We note that $\boldsymbol{b}/q - \boldsymbol{x}^{(1)} = \boldsymbol{c}_0^* - \boldsymbol{x}^{(1)} - \boldsymbol{x}^{(2)} - \nu\mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{y}_1/q$. If we can show that

$$\left\langle \mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{z}_1^*, \nu\mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{y}_1/q \right\rangle$$

is close to the uniform distribution (modulo 1), then $c^{*'}$ will also be close to the uniform distribution (modulo 1), as $c^{*'}$ is masked by this uniformly random number.

Recall that in the hybrids, we set $\boldsymbol{y}_1^* = \boldsymbol{z}_1^*/2 + (\text{shift})$, so it is sufficient for us to analyze $\left\langle \mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{z}_1^*, \nu\mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{z}_1^*/q \right\rangle = \nu\left\langle \mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{z}_1^*, \mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{z}_1^*/q \right\rangle = \nu||\mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{z}_1^*||^2/q$. By applying Lemma 4.2 to the most conservative case (i.e. the Hamming weight of $\boldsymbol{v}$ is 1), we obtain that with overwhelming probability,

$$||\mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{z}_1^*||^2/q \geq \frac{m}{4q}||\boldsymbol{z}_1^*||^2.$$

We recall that $\boldsymbol{z}_1^*$ is sampled from Gaussian with width sq, so its two-norm squared (i.e. ℓ_2^2-norm) is at least $m(sq)^2/2$ with overwhelming probability (by a Chernoff bound argument). Thus, the distribution $\nu||\mathbf{R}^*_{\boldsymbol{v}}\boldsymbol{z}_1^*||^2/q$ is a Gaussian distribution with width at least

$$d = \tau(ms)^2 q/4 = \frac{\gamma(\alpha m s)^2 q}{4\beta^2}.$$

We recall again that ν was sampled from a Gaussian with parameter $\tau = \gamma\alpha^2/\beta^2$. By our setting of parameters, we have $d \geq \omega(\log n)$. A Gaussian with such width is statistically close to uniform in the domain $\mathbb{Z}_1$. This completes the proof. ☐

This completes the proof of Lemma 5.3. Further, Theorem 5.1 follows from Lemmas 5.2 and 5.3. A (multi-distributional) bi-deniable IPE from LWE then follows from Lemma 2.5 and Theorems 3.4 and 5.1. ☐

Parameters	Description	Setting
n, m	lattice dimension	$n = \lambda, m = n^2 \log n$
ℓ	attribute/predicate vector length	$\ell = \sqrt{n}$
q	modulus (resp. bit-precision)	smallest prime $\geq n^3 \log^{4+2\delta}(n)$
α	sampling error terms $\boldsymbol{x}, x$	$\frac{1}{n^{2.5}\log^{3+\delta}(n)}$
β	sampling correlation vector $\boldsymbol{y}$	$1/2$
γ	sampling correlation coefficient μ	$\frac{1}{n \log^{1.5}(n)}$
s	sampling secret key $\boldsymbol{z}$	$3/4$

Table 1: Parameter Description and Simple Example Setting

5.2 Parameter Setting

The parameters in Table 1 are selected in order to satisfy the following constraints (where for simplicity, we choose $\ell := \sqrt{n}, \beta := 1/2$):

- To ensure correctness in Lemma 5.2, we have $8\ell\log(q)ms\alpha\omega(\log(m)) < 1$.
- To ensure deniability in Hybrid H_7, we have $d/\omega(\log(n)) > \frac{\gamma(\alpha ms)^2 q}{4\beta^2\omega(\log(n))} > 1$.
- To ensure large enough LWE noise, we need $\alpha \geq (\sqrt{n}\log^{1+\delta} n)/q$.
- To apply the leftover hash lemma, we need $m \geq 2n\log(q)$.
- To ensure that that the matrix $\mathbf{Q}$ in $\mathsf{FakeRCoins}$ is positive definite, we have $\beta \geq \alpha\gamma\ell\log^{1+\delta}(q)\sqrt{m}$; that is, $1/\gamma \geq (\alpha/\beta)\ell\log^{1+\delta} q\sqrt{m}$. This constraint will also imply that in the security proof, both $\mathbf{Q}'$ and $\mathbf{Q}' - \beta'^2\mathbf{I}_{m\times m}$ are positive definite. (Note $\beta' = \alpha/2$.)
- To ensure hybrid H_3 is well-defined, we have $s > \beta$ and $\beta > s/2$. Let $s := (3/2)\beta$.

For a small constant $\delta > 0$ $\big($and since $q, m \in \mathsf{poly}(n)\big)$, we obtain the constraint:

$$\gamma q > \frac{\ell^2 \log^{4+2\delta}(n)}{\sqrt{m}}.$$

For example, choosing $\ell := \sqrt{n}$ and $\beta := 1/2$ as in Table 1 gives the following feasibility region (primarily bounded between the deniability and positive-definitiveness constraints):

$$\frac{\log^{1+\delta}(n)}{n^2} \leq \gamma \leq \frac{1}{n\log^{1.5}(n)}.$$

We note that this region is satisfiable — i.e. it has "slack" of approximately $\widetilde{\Theta}(\sqrt{m})$. Choosing ℓ as $n^{\epsilon/2}$, for $1/2 < \epsilon < 2$, reduces this feasibility gap from $m^{1/2}$ to $m^{\epsilon'} > 0$, for $\epsilon' > 0$ $\big($up to $\mathsf{poly}(\log(n))$ factors$\big)$.

Regev [Reg05] showed that for $q > \sqrt{m}/\beta'$, an efficient algorithm for $\mathsf{LWE}_{n,m,q,\chi}$ for $\chi = \mathcal{D}_{\beta' q}$ $\big($and $\beta' q \geq \sqrt{n}\omega(\log(n))\big)$ implies an efficient quantum algorithm for approximating the SIVP and GapSVP problems, to within $\tilde{O}(n/\beta')$ approximation factors in the worst case. Our example parameter setting yields a bi-deniable IPE based on the (quantum) hardness of solving $\mathsf{SIVP}_{\widetilde{O}(n^{9.5})}$, respectively $\mathsf{GapSVP}_{\widetilde{O}(n^{9.5})}$. (We write this term to additionally absorb the $(1/q^2)$ loss from our LWE to eLWE^+ reduction.) We leave further optimizing the lattice problem approximation factor to future work, though we speculate it may prove innately hard (or at least require new, very different ideas) to improve the approximation factor beyond $\widetilde{O}(n^{1.5+\epsilon'})^2 = \widetilde{O}(n^{3+\epsilon''})$, for $\epsilon', \epsilon'' > 0$, even assuming a completely tight LWE to eLWE^+ reduction.

Acknowledgments

We thank Jonathan Katz for multiple, useful discussions about deniable encryption. We also thank Jacob Alperin-Sheriff for carefully explaining the extended LWE reductions in [ASP12] and [BLP$^+$13]. We also thank Vadim Lyubashevsky for his insightful comments on sampling from Gaussian-type distributions.

This work is funded in part by NSF award #1223623, NSF grants CNS-1314857, CNS-1453634, CNS-1518765, and CNS-1514261, as well as award 70NANB15H328 from the U.S. Department of Commerce, National Institute of Standards and Technology. Xiong Fan's work was done in part at the University of Maryland and at Yahoo. Feng-Hao Liu's work was done in part at the University of Maryland.

This work was done in part while a subset of the authors were visiting the Simons Institute for the Theory of Computing, particularly during the Mathematics of Modern Cryptography 2015 workshop, supported by the Simons Foundation and by the DIMACS/Simons Collaboration in Cryptography through NSF grant CNS-1523467.

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, Heidelberg, May 2010.

[AFV11] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 21–40. Springer, Heidelberg, December 2011.

[ASP12] Jacob Alperin-Sheriff and Chris Peikert. Circular and KDM security for identity-based encryption. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 334–352. Springer, Heidelberg, May 2012.

[BGG$^+$14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Heidelberg, May 2014.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, Heidelberg, April 2009.

[BLP$^+$13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 575–584. ACM Press, June 2013.

[BNNO11] Rikke Bendlin, Jesper Buus Nielsen, Peter Sebastian Nordholt, and Claudio Orlandi. Lower and upper bounds for deniable public-key encryption. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 125–142. Springer, Heidelberg, December 2011.

[BRS13] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private subspace-membership encryption and its applications. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 255–275. Springer, Heidelberg, December 2013.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, March 2011.

[CDNO97] Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. Deniable encryption. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 90–104. Springer, Heidelberg, August 1997.

[CFGN96] Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *28th ACM STOC*, pages 639–648. ACM Press, May 1996.

[CGP15] Ran Canetti, Shafi Goldwasser, and Oxana Poburinnaya. Adaptively secure two-party computation from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 557–585. Springer, Heidelberg, March 2015.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, Heidelberg, May 2010.

[DKR15] Dana Dachman-Soled, Jonathan Katz, and Vanishree Rao. Adaptively secure, universally composable, multiparty computation in constant rounds. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 586–613. Springer, Heidelberg, March 2015.

[DLZ15] Dana Dachman-Soled, Feng-Hao Liu, and Hong-Sheng Zhou. Leakage-resilient circuits revisited - optimal number of computing components without leak-free hardware. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 131–158. Springer, Heidelberg, April 2015.

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 523–540. Springer, Heidelberg, May 2004.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

[GP15] Sanjam Garg and Antigoni Polychroniadou. Two-round adaptively secure MPC from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 614–637. Springer, Heidelberg, March 2015.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 503–523, 2015.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, Heidelberg, April 2008.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 465–484. Springer, Heidelberg, August 2011.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Heidelberg, April 2012.

[OPW11] Adam O'Neill, Chris Peikert, and Brent Waters. Bi-deniable public-key encryption. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 525–542. Springer, Heidelberg, August 2011.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.