Public-Key Encryption and Injective Trapdoor Functions from LWE with Large Noise Rate

Liheng Ji[1,2] and Yilei Chen[1,2,3]

[1] Tsinghua University, Beijing, China 100084
jlh23@mails.tsinghua.edu.cn, chenyilei@mail.tsinghua.edu.cn
[2] Shanghai Qi Zhi Institute, Shanghai, China 200232
[3] Shanghai Artificial Intelligence Laboratory, Shanghai, China 200232

Abstract. The hardness of the learning with errors (LWE) problem increases as its noise rate grows. However, all existing LWE-based public-key encryption schemes require the noise rate to be no greater than $o(1/(\sqrt{n}\log n))$. Breaking through this limitation presents an intriguing challenge.

In this paper, we construct public-key encryption (PKE) schemes based on the sub-exponential hardness of decisional LWE with polynomial modulus and noise rate ranging from $O(1/\sqrt{n})$ to $o(1/\log n)$. More concretely, we demonstrate the existence of CPA-secure PKE schemes as long as one of the following three assumptions holds.

(i) $(n^{\omega(1)}, n^{-\omega(1)})-$hardness of decisional LWE with noise rate $O(1/\sqrt{n})$.
(ii) $(2^{\omega(n^{1/c_1})}, 2^{-\omega(n^{1/c_1})})$-hardness of decisional LWE with noise rate $O(1/\sqrt{n^{1-1/c_1}\log n})$ for some constant $c_1 > 1$.
(iii) $(2^{\omega(n/\log^{c_2} n)}, 2^{-\omega(n/\log^{c_2} n)})$-hardness of decisional LWE with noise rate $O(1/\sqrt{\log^{c_2+1} n})$ for some constant $c_2 > 0$.

Here, (t, ϵ)-hardness means no adversary running in time t can gain advantage exceeding ϵ.

We also construct injective trapdoor function (iTDF) families based on similar hardness assumption as our PKE. To achieve this, we give a generalization of Babai's nearest plane algorithm, which finds a "common closest lattice point" for a set of vectors.

In addition, we propose a PKE based on the $(2^{\omega(n^{1/2})}, 2^{-\omega(n^{1/2})})$-hardness of constant noise learning parity with noise (LPN) problem. Our construction is simpler than the construction of Yu and Zhang [CRYPTO 2016] while achieving the same security.

Keywords: Lattice · Learning with Errors · Public-Key Encryption · Injective Trapdoor Function· Learning Parity with Noise.

1 Introduction

1.1 Background

Recently, cryptographers have shown fervent interest in the study of the learning with errors problem (LWE). This enthusiasm is not only due to its capability

of constructing traditional cryptographic primitives, such as public-key encryption (PKE) schemes [Reg05], signatures [GPV08], but also for more advanced usages like fully homomorphic encryption [BV11, GSW13], identity-based encryption [GPV08], and attribute-based encryption [GVW13]. Furthermore, the LWE problem is also conjectured to be hard against quantum computers.

Let us give a brief introduction to the LWE problem. Given a public matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a secret $\mathbf{s} \leftarrow \chi_s^n$, an error $\mathbf{e} \leftarrow \chi_e^m$, the search LWE problem asks the adversary to recover $\mathbf{s}$ from $(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e})$. The decision LWE problem asks the distinguisher to tell whether the LWE samples are sampled according to the real LWE distribution or the uniform random distribution. For certain parameters, decision LWE is proven to be as hard as the search version [Reg05, MP12]. In this paper, we focus on a canonical variant of LWE in [Reg05], where χ_s is the uniform distribution on $\mathbb{Z}_q$, and $\chi_e \sim \Psi_{\alpha,q}$ is the rounded Gaussian distribution with width αq. We call this problem by $\mathsf{LWE}(n, q, \alpha)$, and call α its noise rate. Reductions from hard lattice problems (e.g. SIVP,GapSVP) to LWE have been well studied [Reg05, PRS17]. We present the recent reduction by Aggarwal et al. [ABB+23] which allows the reduction to run in subexponential time.

Lemma 1 ([ABB+23], Theorem 5.5). *Let $n, q \in \mathbb{N}^+$, $\alpha \in (0, 1)$, $\gamma \geq \frac{10\sqrt{n}}{\alpha}$ satisfy $\alpha q \geq 2\sqrt{n}$. There is a quantum reduction from GapSVP_γ for all lattices to decision $\mathsf{LWE}(n, q, \alpha)$ that runs in time $(1 - \frac{4n}{(\alpha\gamma)^2})^{-n/2} \cdot \mathrm{poly}(n)$.*

In the lecture notes of Vaikuntanathan [Vai20], he mentioned the problem of constructing a public-key encryption scheme from LWE with $\alpha \in O(1)$ as an open problem. Currently, most constructions of LWE-PKE require the noise rate to be upper-bounded by $o(1/(\sqrt{n}\log n))$, while some other primitives even need the LWE noise to be sub-exponentially small. This naturally sparks the following questions.

- *What is the largest possible noise rate of LWE that enables public-key encryption?*
- *Can we use large noise LWE to achieve other cryptographic primitives?*

In this paper, we explore answers to the questions above, and the main results are presented in the following subsection.

1.2 Main Results

Public-key encryption schemes from large noise LWE. We manage to construct PKE schemes with chosen-plaintext attack (CPA) security on LWE with polynomial modulus and noise rate ranging from $O(1/\sqrt{n})$ to $o(1/\log n)$.

Theorem 1. *Let n be the security parameter, and $q := q(n) \in \mathsf{poly}(n)$ be the modulus such that $q > \omega(n)$. There exists a PKE scheme with CPA-security if one of the following three conditions holds.*

(i) The $(n^{\omega(1)}, n^{-\omega(1)})$-hardness of decision $\mathsf{LWE}(n, q(n), O(1/\sqrt{n}))$.

(ii) *The* $(2^{\omega(n^{\frac{1}{c_1}})}, 2^{-\omega(n^{\frac{1}{c_1}})})$*-hardness of decision* $\mathsf{LWE}(n, q(n), O(1/\sqrt{n^{1-\frac{1}{c_1}}\log n}))$*, for some constant* $c_1 > 1$*.*

(iii) *The* $(2^{\omega(\frac{n}{\log^{c_2} n})}, 2^{-\omega(\frac{n}{\log^{c_2} n})})$*-hardness of decision* $\mathsf{LWE}(n, q(n), O(1/\sqrt{\log^{c_2+1} n}))$*, for some constant* $c_2 > 0$*.*

Here, (t, ϵ)-hardness means any adversary with running time t has an advantage bounded by ϵ.

Combining with Lemma 1, we show that PKE exists assuming the subexponential hardness of worst-case lattice problems like GapSVP_γ for the approximation factor γ as small as $\tilde{O}(\sqrt{n})$, improving upon the previous best result of $\gamma \in \tilde{O}(n)$. In particular, by letting $\alpha = O(1/\sqrt{\log^{c_2+1} n})$ for some constant $c_2 > 0$, we can choose any $\gamma \in \omega(\sqrt{n \log^{c_2+1} n})$ in Lemma 1 and show PKE exists assuming no quantum algorithm solves GapSVP_γ in time $2^{o(n)}$.

Injective trapdoor functions from large noise LWE. Building on the results of [MP12], we construct a family of injective trapdoor functions (injective TDFs, iTDFs) on the same hardness of search LWE with the same noise rate as our PKE.

Theorem 2. *Let* n *be the security parameter and* $q = q(n) \in \mathsf{poly}(n)$ *be a prime such that* $q > \omega(n)$*. There exists a TDF family if one of the following three conditions holds.*

(i) *The* $(n^{\omega(1)}, n^{-\omega(1)})$*-hardness of search* $\mathsf{LWE}(n, q(n), O(1/\sqrt{n}))$*.*

(ii) *The* $(2^{\omega(n^{\frac{1}{c_1}})}, 2^{-\omega(n^{\frac{1}{c_1}})})$*-hardness of search* $\mathsf{LWE}(n, q(n), O(1/\sqrt{n^{1-\frac{1}{c_1}}\log n}))$*, for some constant* $c_1 > 1$*.*

(iii) *The* $(2^{\omega(\frac{n}{\log^{c_2} n})}, 2^{-\omega(\frac{n}{\log^{c_2} n})})$*-hardness of search* $\mathsf{LWE}(n, q(n), O(1/\sqrt{\log^{c_2+1} n}))$*, for some constant* $c_2 > 0$*.*

Specifically, assuming condition (i) or condition (ii) with $c_1 \geq 2$*, the TDF family is injective.*

In the construction, we need to extend Babai's nearest plane algorithm [Bab86] to identify a "common closest lattice point" for a set of vectors. This is presented in Section 4.

Public-key encryption schemes from constant noise LPN. In [YZ16], Yu and Zhang proposed a PKE scheme based on the $(2^{\omega(n^{1/2})}, 2^{-\omega(n^{1/2})})$-hardness of LPN with constant noise. Using a construction similar to our LWE scheme, we get a PKE scheme which is simpler than that in [YZ16] while achieving the same security.

Theorem 3. *Let* n *be the security parameter. There exists a PKE scheme with CPA-security assuming the* $(2^{\omega(n^{1/2})}, 2^{-\omega(n^{1/2})})$*-hardness of* $\mathsf{LPN}(n, \mu)$*, for some constant* $0 < \mu < 1/2$*.*

1.3 Technique Overview

Public-key encryption schemes from large noise LWE. Our starting point is Regev's PKE scheme [Reg05]. Let λ be the security parameter of this PKE scheme. Let $q = q(\lambda) = \mathsf{poly}(\lambda)$ be a prime such that $q > \omega(\lambda)$, $n = \Omega(\lambda \log q) = \Omega(\lambda \log \lambda)$. Let $\Psi_{\alpha,q} := \lfloor D_{\alpha,q} \rceil$ denote the discretized Gaussian distribution with width αq (see Appendix A.2 for a detailed explanation). The key generation algorithm samples $\mathbf{A} \leftarrow \mathbb{Z}_q^{\lambda \times n}, \mathbf{s} \leftarrow \mathbb{Z}_q^{\lambda}$, $\mathbf{e} \leftarrow \Psi_{\alpha,q}^n$, computes $\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e} \bmod q$, and sets $(pk, sk) = ((\mathbf{A}, \mathbf{b}), \mathbf{s})$. For any message $\mathfrak{m} \in \{0,1\}$, the encryption algorithm samples $\mathbf{r} \leftarrow \{0,1\}^n$, and outputs the ciphertext $\mathfrak{c} = (\mathbf{c}_1, c_2) = (\mathbf{A}\mathbf{r}, \langle \mathbf{r}, \mathbf{b} \rangle + \lfloor q/2 \rceil \cdot \mathfrak{m})$. Finally, the decryption algorithm computes $\Delta := c_2 - \mathbf{c}_1^T\mathbf{s} = \langle \mathbf{r}, \mathbf{e} \rangle + \lfloor q/2 \rceil \cdot \mathfrak{m}$, checks whether Δ is closer to 0 or $\lfloor q/2 \rceil$, and outputs 0 or 1 accordingly. Since $\langle \mathbf{r}, \mathbf{e} \rangle \sim \Psi_{\alpha,q}^{\star O(n)}$ ($\Psi_{\alpha,q}^{\star t}$ denotes the distribution of summing up t elements sampled independently from $\Psi_{\alpha,q}$), we have when $\alpha = o(1/(\sqrt{n \log \lambda}))$, this scheme is correct with probability $1 - \mathsf{negl}(\lambda)$ (see Corollary 2 for the calculation). By the hardness of LWE and the leftover hash lemma (Lemma 7), we have $(\mathbf{A}, \mathbf{b}, \mathbf{A}\mathbf{r}, \langle \mathbf{r}, \mathbf{b} \rangle) \approx_c (\mathbf{A}, \mathbf{b}', \mathbf{A}\mathbf{r}, \langle \mathbf{r}, \mathbf{b}' \rangle) \approx_s (\mathbf{A}, \mathbf{b}', \mathcal{U}_q^{\lambda+1})$, where $\mathbf{b}' \sim \mathcal{U}_q^n$, $\approx_c$, $\approx_s$ stands for computational and statistical indistinguishability. This suffices to prove the security of the encryption scheme.

Now we modify this scheme by the following steps.

- **Step 1: Raise the noise rate to $\alpha = O(1/\sqrt{n})$, while keeping the setting of the other parameters unchanged.** Now the new scheme is correct with probability 2/3, which is still sufficient since we can improve the correctness by applying parallel repetition.
- **Step 2: Replace the distribution of $\mathbf{r}$ by $\Xi^{[n;k]}$, where $k = \Theta(\lambda \log \lambda / \log n)$, and $\Xi^{[n;k]}$ represents the uniform distribution over the set**

$$\{\mathbf{v} \in \{0,1\}^n | \text{ the Hamming weight of } \mathbf{v} \text{ is } k\}.$$

  We also let $n = \Omega(\lambda^{>1})$, and $\alpha = O(1/\sqrt{k}) = O(1/\sqrt{\lambda \log \lambda / \log n})$.
  - Security: Since $k = O(n^{<1})$, we have $H_\infty(\Xi^{[n;k]}) = \Theta(k \log n) = \Theta(\lambda \log \lambda)$ (by Lemma 10). Using the leftover hash lemma, we have $(\mathbf{A}, \mathbf{b}', \mathbf{A}\mathbf{r}, \langle \mathbf{r}, \mathbf{b}' \rangle) \approx_s (\mathbf{A}, \mathbf{b}', \mathcal{U}_q^{\lambda+1})$. So the security still holds.
  - Correctness: We have $\langle \mathbf{r}, \mathbf{e} \rangle \sim \Psi_{\alpha,q}^{\star k}$. Since we set $\alpha = O(1/\sqrt{k})$, the scheme is correct with probability 2/3.

  We remark that if the scheme is not required to be efficient, we can let n be subexponential in λ, and then the noise rate ($\alpha = O(1/\sqrt{\lambda \log \lambda / \log n})$) will become even larger. The next step applies this idea more cleverly.
- **Step 3: Change the security parameter of the PKE scheme from λ to n, and let $\lambda \in (\omega(\log n), O(n^{<1})]$.** This modification makes n subexponential in λ, while maintaining the efficiency of the scheme. However, the security assumption becomes tricky: we need to assume the $(n^{\omega(1)}, n^{-\omega(1)})$-hardness of $\mathsf{LWE}(\lambda, q(\lambda), O(1/\sqrt{\lambda \log \lambda / \log n}))$ to make the scheme secure. Actually, this hardness assumption aligns with the subexponential hardness assumption in Theorem 1, and we provide a detailed explanation for this in Lemma 2. We also note that this step works only if α is larger than $O(1/\sqrt{\lambda})$, which means the previous two steps are indeed necessary.

Using a similar method, we can also construct a PKE scheme based on the $(n^{\omega(1)}, n^{-\omega(1)})$-hardness of $\mathsf{LPN}(\lambda, \mu)$, where $\lambda = \Theta(\log^2 n)$, and $0 < \mu < \frac{1}{2}$ is a constant. This aligns with the result in Theorem 3.

Injective trapdoor function from large noise LWE. Next, we construct an iTDF family from LWE with a large noise rate.

We first recall the trapdoor function in [MP12]. Let λ be the security parameter, $q = q(\lambda) = \mathsf{poly}(\lambda)$ be a prime such that $q > \omega(\lambda)$, $\kappa = \lceil \log_2 q \rceil$, $w = \lambda\kappa$. Let $\mathbf{G}_\lambda$ be the gadget matrix. For a random matrix $\bar{\mathbf{A}} \leftarrow \lambda \times (n-w)$, an invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{\lambda\times\lambda}$, the trapdoor generation algorithm samples $\mathbf{R} \leftarrow \{0,1\}^{(n-w)\times w}$, and outputs the public evaluation key $\mathbf{A} = (\bar{\mathbf{A}} \| \mathbf{H}\mathbf{G}_\lambda - \bar{\mathbf{A}}\mathbf{R})$, along with the private inversion key (trapdoor) $\mathbf{R}$. On input $(\mathbf{s}, \mathbf{e})$ such that $\mathbf{s} \leftarrow \mathbb{Z}_q^\lambda$ and $\mathbf{e} \leftarrow \Psi^n_{o(1/(\sqrt{\lambda}\log\lambda)),q}$, the evaluation algorithm (accessible to $\mathbf{A}$) outputs $\mathbf{b} := \mathbf{A}^T\mathbf{s} + \mathbf{e}$. The inversion algorithm (accessible to $\mathbf{A}, \mathbf{R}$), on input $\mathbf{b}$, computes $[\mathbf{R}^T \| \mathbf{I}]\mathbf{b} = \mathbf{G}_\lambda^T\mathbf{H}^T\mathbf{s} + [\mathbf{R}^T \| \mathbf{I}]\mathbf{e}$, uses Babai's nearest plane algorithm on $\Lambda(\mathbf{G}_\lambda^T)$ (the q-ary lattices is defined in Eqn. (2)) to get $\mathbf{H}^T\mathbf{s}$, and then recovers $\mathbf{s}, \mathbf{e}$ in the end.

Our idea is applying a transformation resembling what we do for PKE, except that we use the truncated Bernoulli distribution $\tilde{\mathcal{B}}^{\bar{n}}_{k/\bar{n}}$ (see Definition 10) instead of $\Xi^{[\bar{n};k]}$, which has a similar min-entropy to $\Xi^{[\bar{n};k]}$ but enables us to calculate independent bounds for different errors. More concretely, we replace the security parameter by n, and let $\ell = \omega(\log n)$, $\bar{n} = n - \ell w$, $k = \Theta(\lambda \log\lambda / \log n)$. For a random $\bar{\mathbf{A}} \leftarrow \mathbb{Z}_q^{\lambda\times\bar{n}}$ and $\mathbf{H} \in \mathbb{Z}_q^{\lambda\times\lambda}$, the trapdoor generation algorithm samples $\mathbf{R}_1, \ldots, \mathbf{R}_\ell \leftarrow (\tilde{\mathcal{B}}^{\bar{n}}_{k/\bar{n}})^{\otimes w}$, and outputs $\mathbf{A} = (\bar{\mathbf{A}} \| \mathbf{H}\mathbf{G}_\lambda - \bar{\mathbf{A}}\mathbf{R}_1 \| \ldots \| \mathbf{H}\mathbf{G}_\lambda - \bar{\mathbf{A}}\mathbf{R}_\ell)$ with trapdoor $(\mathbf{R}_1, \ldots, \mathbf{R}_\ell)$. On input $(\mathbf{s}, \mathbf{e})$ such that $\mathbf{s} \leftarrow \mathbb{Z}_q^\lambda$ and $\mathbf{e} \leftarrow \Psi^n_{O(1/\sqrt{k}),q}$, the evaluation algorithm outputs $\mathbf{b} := \mathbf{A}^T\mathbf{s} + \mathbf{e}$. As for the inversion algorithm, on input $\mathbf{b}$, it first parses $\mathbf{b} = (\bar{\mathbf{b}}^T \| \underline{\mathbf{b}}_1^T \| \ldots \| \underline{\mathbf{b}}_\ell^T)^T \in \mathbb{Z}_q^{\bar{n}} \times (\mathbb{Z}_q^w)^{\otimes\ell}$, $\mathbf{e} = (\bar{\mathbf{e}}^T \| \underline{\mathbf{e}}_1^T \| \ldots \| \underline{\mathbf{e}}_\ell^T)^T \in \mathbb{Z}_q^{\bar{n}} \times (\mathbb{Z}_q^w)^{\otimes\ell}$, and computes $\hat{\mathbf{b}}_i = \mathbf{R}_i^T\bar{\mathbf{b}} + \underline{\mathbf{b}}_i = \mathbf{G}_\lambda^T\mathbf{H}^T\mathbf{s} + \mathbf{R}_i^T\bar{\mathbf{e}} + \underline{\mathbf{e}}_i$. Then, it only needs to recover $\mathbf{H}^T\mathbf{s}$ from $\{\hat{\mathbf{b}}_i\}_{1\le i\le\ell}$, which enables the calculation of $\mathbf{s}, \mathbf{e}$.

However, the error vector $\mathbf{R}_i^T\bar{\mathbf{e}} + \underline{\mathbf{e}}_i$ is so large that merely applying the traditional nearest plane algorithm results in a very low probability of successfully recovering $\mathbf{H}^T\mathbf{s}$. In other words, we need many times of parallel repetitions to achieve strong correctness (ℓ should be $\omega(2^\lambda)$), which will destroy the structure of our trapdoor function. To deal with this, we make a non-trivial modification to Babai's algorithm, which improves its error correction ability.

Nearest-plane algorithm for large noise vectors. Let L be any n-dimensional lattice, $\mathbf{v} \in L$ be a lattice vector, $\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(\ell)} \in \mathbb{R}^n$ be independent random error vectors. Our goal is to recover $\mathbf{v}$ from $\{\mathbf{v} + \mathbf{e}^{(i)}\}_{1\le i\le\ell}$ when the errors are small.

Let $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be an ordered basis of L, then we can write $\mathbf{v} = \sum_{j=1}^n c_j\mathbf{b}_j$ for some $c_1, \ldots, c_n \in \mathbb{Z}$. The idea is to follow the process of Babai's nearest-plane algorithm [Bab86] , except that we do a parallel repetition (error

correction) when computing each c_j. More concretely, if for most of $\mathbf{e}^{(i)}$, its projection to every orthogonal basis vector $\tilde{\mathbf{b}}_j$ is of length less than $\|\tilde{\mathbf{b}}_j\|/2$, then, by computing the projection of every $\mathbf{v}+\mathbf{e}^{(i)}$ to $\tilde{\mathbf{b}}_j$ and taking a majority, we will recover the projection of $\mathbf{v}$ to $\tilde{\mathbf{b}}_j$. Using the property of Gram-Schmidt orthogonalization, we can recover $c_n, c_{n-1}, \ldots, c_1$ one by one.

1.4 Applications

Oblivious transfers from large noise LWE and constant noise LPN. Gertner et al. [GKM+00] demonstrated that for a PKE scheme with CPA-security, if its public key can be sampled without the knowledge of the corresponding secret key and remains indistinguishable from a legitimately generated public key, then the scheme can be transformed into an oblivious transfer (OT) in a black-box manner. It is straightforward to verify that our PKE schemes, derived from both large noise LWE and constant noise LPN, satisfy these criteria, thereby guaranteeing the existence of OTs based on the same hardness assumptions.

Applications of iTDF from large noise LWE. Bartusek et al. [BKP23] proved the existence of PKE and commitment schemes with publicly-verifiable deletion (PVD) under the assumption that post-quantum iTDF exists, therefore our result directly implies that PKE and commitment with PVD exist assuming large noise LWE is hard against quantum attackers. Furthermore, for any post-quantum primitive $X \in \{$attribute-based encryption, quantum fully-homomorphic encryption, witness encryption, time-revocable encryption$\}$, they also established the existence of X with PVD, assuming the availability of both X and iTDFs. We can therefore plug in our iTDF construction in those constructions.

2 Preliminary

Let n be the security parameter. For a distribution or a set $\mathcal{X}$, $x \leftarrow \mathcal{X}$ denotes sampling x according to the distribution or uniformly at random from $\mathcal{X}$. For any $k \in \mathbb{N}$, define $\chi^{\star k}$ as the distribution obtained by the sum of k elements sampled independently from the distribution χ. For $x \in \mathbb{R}$, let $\lfloor x \rceil$ be the integer closest to x (and the smaller one if there are two closest candidates). For any integer $q \geq 2$, let $\mathcal{U}_q$ denote the uniform distribution on $\mathbb{Z}_q$, and we use $\mathcal{U}$ to denote $\mathcal{U}_2$. For any $a \in \mathbb{Z}_q$, define $|a| := \min\{a, q-a\}$ to represent the absolute difference in modular arithmetic. For any $a \in \mathbb{Z}_q$ and $b \in \mathbb{Z}$, define the value of $(a+b)$ (resp. $(a-b)$, $a \cdot b$) by treating a as an element of $\{0, \ldots, q-1\} \subseteq \mathbb{Z}$ and then adding (resp. subtracting, multiplying) it by b. For any $x > 0$, we use $\log x$ to denote $\log_2 x$.

We use bold uppercase letters to denote matrices, bold lowercase letters to denote column vectors, and standard lowercase letters to represent scalar values. For any matrix $\mathbf{A}$, use $\mathbf{A}^T$ to denote its transpose. If $\mathbf{A}$ is invertible, use $\mathbf{A}^{-1}$ to denote its inverse, and define $\mathbf{A}^{-T} = (\mathbf{A}^{-1})^T$. For any n-dimensional vector $\mathbf{v}$ and any integer $i \in [1, n]$, let $v(i)$ denote the ith component of $\mathbf{v}$. For any

$t \in [1, \infty]$, the ℓ_t-norm of vector $\mathbf{v}$ is denoted by $\|\mathbf{v}\|_t$, with the ℓ_2-norm simply represented as $\|\mathbf{v}\|$. For any two positive integers $k \leq m$, $\Xi^{[m;k]}$ denotes the uniform distribution on the set $\{\mathbf{v} \in \{0,1\}^m \mid \mathsf{Ham}(\mathbf{v}) = k\}$, where $\mathsf{Ham}(\mathbf{v})$ denotes the Hamming weight of the vector $\mathbf{v}$.

For any $\mu \in (0,1)$, let $\mathcal{B}_\mu$ denote the Bernoulli distribution with parameter μ. For any finite set $\mathcal{S}$, denote by $|\mathcal{S}|$ the number of elements in $\mathcal{S}$. Denote by $\mathsf{poly}(n)$ some polynomial function of n, by $\mathsf{negl}(n)$ some negligible function of n, and by $n^{<1}$ (resp. $n^{>1}$) some function n^c where c is a constant in $(0,1)$ (resp. $(1,\infty)$). When we say some event $\mathcal{E}$ happens with overwhelming probability, it means $\Pr[\mathcal{E}] = 1 - \mathsf{negl}(n)$. "PPT" stands for probabilistic polynomial time. The statistical distance between two random variables X and Y is defined as $\mathsf{SD}(X,Y) := \frac{1}{2}\sum_x |\Pr[X = x] - \Pr[Y = x]|$, where the sum is over all possible outcomes x. If $\mathsf{SD}(X,Y) = \mathsf{negl}(n)$, we say X, Y are statistically indistinguishable, denoted by $X \approx_s Y$. If for every PPT adversary $\mathcal{A}$, $|\Pr[\mathcal{A}(X) = 1] - \Pr[\mathcal{A}(Y) = 1]| = \mathsf{negl}(n)$, we say X, Y are computationally indistinguishable, denoted by $X \approx_c Y$. We have $X \approx_s Y$ implies $X \approx_c Y$. Denote the min-entropy of X by $H_\infty(X) := -\log(\max_x \Pr[X = x])$. The statistical distance, indistinguishability, and min-entropy of probability distributions are defined similarly.

Due to space limitations, we postpone the rest of this section to Appendix A.

3 Public-Key Encryption Scheme from LWE with Large Noise

In this section, we construct CPA-secure PKE schemes on subexponential hardness of LWE with large noise rates. Concretely, Subsection 3.1 and 3.2 are dedicated to proving the following theorem.

Theorem 4. *Let n be the security parameter for the PKE scheme. For any $\lambda \in (\omega(\log n), O(n^{<1})]$, and any prime $q = q(\lambda) \in \mathsf{poly}(\lambda)$ such that $q > \omega(\lambda)$, there exists a PKE scheme with CPA-security assuming the $(n^{\omega(1)}, n^{-\omega(1)})$-hardness of decision $\mathsf{LWE}(\lambda, q(\lambda), O(1/\sqrt{\lambda \log \lambda / \log n}))$.*

The following lemma gives a better illustration of the hardness assumption in Theorem 4.

Lemma 2. *For any $\lambda \in (\omega(\log n), O(n^{<1})]$, and any $q(\lambda) \in \mathsf{poly}(\lambda)$,*

(i) if $\lambda = \Theta(n^{c_0})$ for some constant $0 < c_0 < 1$, we have the $(n^{\omega(1)}, n^{-\omega(1)})$-hardness of decision (resp., search) $\mathsf{LWE}(\lambda, q(\lambda), O(1/\sqrt{\lambda \log \lambda / \log n}))$ is implied by the hardness of the decision (resp., search) $\mathsf{LWE}(n, q(n), O(1/\sqrt{n}))$.

(ii) if $\lambda = \Theta(\log^{c_1} n)$ for some constant $c_1 > 1$, we have the $(n^{\omega(1)}, n^{-\omega(1)})$-hardness of decision (resp., search) $\mathsf{LWE}(\lambda, q(\lambda), O(1/\sqrt{\lambda \log \lambda / \log n}))$ is implied by the $(2^{\omega(n^{\frac{1}{c_1}})}, 2^{-\omega(n^{\frac{1}{c_1}})})$-hardness of the decision (resp., search) $\mathsf{LWE}(n, q(n), O(1/\sqrt{n^{1-\frac{1}{c_1}} \log n}))$.

(iii) *if* $\lambda = \Theta(\log n(\log\log n)^{c_2})$ *for some constant* $c_2 > 0$*, we have the* $(n^{\omega(1)}, n^{-\omega(1)})$*-hardness of decision (resp., search)* $\mathsf{LWE}(\lambda, q(\lambda), O(1/\sqrt{\lambda \log \lambda / \log n}))$ *is implied by the* $(2^{\omega(\frac{n}{\log^{c_2} n})}, 2^{-\omega(\frac{n}{\log^{c_2} n})})$*-hardness of the decision (resp., search)* $\mathsf{LWE}(n, q(n), O(1/\sqrt{\log^{c_2+1} n}))$.

Remark 1. The PKE proposed by Regev [Reg05] is based on the $(n^{\omega(1)}, n^{-\omega(1)})$-hardness of decision $\mathsf{LWE}(n, q(n), o(1/(\sqrt{n}\log n)))$. The LWE problem in (i) improves the noise by an $\omega(\log n)$ factor, and thus has a stronger hardness. As for the LWE problems in (ii) and (iii), we do not know how to compare their hardness to that in (i). However, they can be viewed as a trade-off between the secret length λ and the noise rate, which has not appeared in Regev's PKE [Reg05].

Proof. In the following, we do not distinguish between the search problem and the decision problem, as their proofs are the same.

(i) When $\lambda = \Theta(n^{c_0})$, we have $n^{\omega(1)} = \lambda^{\omega(1)}$, and $O(1/\sqrt{\lambda \log \lambda / \log n}) = O(1/\sqrt{\lambda \log \lambda / \log(\lambda^{1/c_0})}) = O(1/\sqrt{\lambda})$.
The $(n^{\omega(1)}, n^{-\omega(1)})$-hardness of $\mathsf{LWE}(n, q(n), O(1/\sqrt{n}))$ implies the $(\lambda^{\omega(1)}, \lambda^{-\omega(1)})$-hardness of $\mathsf{LWE}(\lambda, q(\lambda), O(1/\sqrt{\lambda}))$, which is exactly the $(n^{\omega(1)}, n^{-\omega(1)})$-hardness of $\mathsf{LWE}(\lambda, q(\lambda), O(1/\sqrt{\lambda \log \lambda / \log n}))$.

(ii) When $\lambda = \Theta(\log^{c_1} n)$, we have $n^{\omega(1)} = 2^{\Theta(\lambda^{1/c_1})}$, and $O(1/\sqrt{\lambda \log \lambda / \log n}) = O(1/\sqrt{(\lambda \log \lambda)/\lambda^{1/c_1}}) = O(1/\sqrt{\lambda^{1-1/c_1} \log \lambda})$.
The $(2^{\omega(n^{\frac{1}{c_1}})}, 2^{-\omega(n^{\frac{1}{c_1}})})$-hardness of $\mathsf{LWE}(n, q(n), O(1/\sqrt{n^{1-\frac{1}{c_1}} \log n}))$ implies the $(2^{\omega(\lambda^{\frac{1}{c_1}})}, 2^{-\omega(\lambda^{\frac{1}{c_1}})})$-hardness of $\mathsf{LWE}(\lambda, q(\lambda), O(1/\sqrt{\lambda^{1-\frac{1}{c_1}} \log \lambda}))$, which is exactly the $(n^{\omega(1)}, n^{-\omega(1)})$-hardness of $\mathsf{LWE}(\lambda, q(\lambda), O(1/\sqrt{\lambda \log \lambda / \log n}))$.

(iii) When $\lambda = \Theta(\log n(\log\log n)^{c_2})$, we have $\log \lambda = \Theta(\log\log n + c_2 \log\log\log n) = \Theta(\log\log n)$. Substituting back, we find $\lambda = \Theta(\log n \log^{c_2} \lambda)$, leading to $\log n = \Theta(\frac{\lambda}{\log^{c_2} \lambda})$. Then $n^{\omega(1)} = 2^{\omega(\frac{\lambda}{\log^{c_2} \lambda})}$, and $O(1/\sqrt{\lambda \log \lambda / \log n}) = O(1/\sqrt{\lambda \log \lambda / (\frac{\lambda}{\log^{c_2} \lambda})}) = O(1/\sqrt{\log^{c_2+1} \lambda})$.
The $(2^{\omega(\frac{n}{\log^{c_2} n})}, 2^{-\omega(\frac{n}{\log^{c_2} n})})$-hardness of $\mathsf{LWE}(n, q(n), O(1/\sqrt{\log^{c_2+1} n}))$ implies the $(2^{\omega(\frac{\lambda}{\log^{c_2} \lambda})}, 2^{-\omega(\frac{\lambda}{\log^{c_2} \lambda})})$-hardness of $\mathsf{LWE}(\lambda, q(\lambda), O(1/\sqrt{\log^{c_2+1} \lambda}))$, which is exactly the $(n^{\omega(1)}, n^{-\omega(1)})$-hardness of $\mathsf{LWE}(\lambda, q(\lambda), O(1/\sqrt{\lambda \log \lambda / \log n}))$.

□

By combining Theorem 4 and Lemma 2, we have the following corollary.

Corollary 1 (Theorem 1). *Let* n *be the security parameter, and* $q = q(n) \in \mathsf{poly}(n)$ *be a prime such that* $q > \omega(n)$*. There exists a construction of PKE scheme with CPA-security assuming one of the following three conditions.*

(i) *The* $(n^{\omega(1)}, n^{-\omega(1)})$*-hardness of decision* $\mathsf{LWE}(n, q(n), O(1/\sqrt{n}))$.

(ii) *The* $(2^{\omega(n^{\frac{1}{c_1}})}, 2^{-\omega(n^{\frac{1}{c_1}})})$*-hardness of decision* $\mathsf{LWE}(n, q(n), O(1/\sqrt{n^{1-\frac{1}{c_1}} \log n}))$*, for some constant* $c_1 > 1$.

(iii) *The* $(2^{\omega(\frac{n}{\log^{c_2} n})}, 2^{-\omega(\frac{n}{\log^{c_2} n})})$*-hardness of decision* $\mathsf{LWE}(n, q(n), O(1/\sqrt{\log^{c_2+1} n}))$*, for some constant* $c_2 > 0$.

3.1 Single-Bit LWE-PKE Scheme with Weak Correctness

In this subsection, we transform Regev's LWE-PKE [Reg05] into a single-bit public-key encryption scheme $\Pi_\lambda^{LWE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with 2/3-correctness.

Construction Let n be the security parameter of the PKE scheme, $\lambda \in (\omega(\log n), O(n^{<1})]$ be the dimension of the LWE secret, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ be a prime such that $q > \omega(\lambda)$. Let $k = \Theta(\lambda \log \lambda / \log n)$ s.t. $H_\infty(\Xi^{[n:k]}) > 2(\lambda+1)\log q$. (For the existence of such a k, please refer to Lemma 10.) Let $\chi_r \sim \Xi^{[n;k]}$, $\alpha \leq \frac{1}{10\sqrt{k}}$.

- The message space is $\mathcal{M} = \{0,1\}$.
- $\mathsf{KeyGen}(1^n)$: Given the security parameter 1^n, it samples $\mathbf{A} \leftarrow \mathbb{Z}_q^{\lambda \times n}$, as well as $\mathbf{s} \leftarrow \mathbb{Z}_q^\lambda$, $\mathbf{e} \leftarrow (\Psi_{\alpha,q})^n$. Then it computes $\mathbf{b} := (\mathbf{A}^T\mathbf{s} + \mathbf{e}) \bmod q$, and sets $(pk, sk) := ((\mathbf{A}, \mathbf{b}), \mathbf{s})$.
- $\mathsf{Enc}(pk, \mathfrak{m})$: Given the public key $pk = (\mathbf{A}, \mathbf{b})$ and the message $\mathfrak{m} \in \mathcal{M}$, it samples $\mathbf{r} \leftarrow \chi_r$, and output $\mathfrak{c} := (\mathbf{c}_1, c_2)$ as ciphertext, where $\mathbf{c}_1 := \mathbf{A}\mathbf{r}$ and $c_2 := (\mathbf{r}^T\mathbf{b} + \lfloor q/2 \rceil \cdot \mathfrak{m}) \mod q$.
- $\mathsf{Dec}(sk, \mathfrak{c})$: Given the secret key $sk = \mathbf{s}$ and the ciphertext $\mathfrak{c}$, parse $\mathfrak{c}$ into $(\mathbf{c}_1, c_2)$, calculate $\Delta := (c_2 - \mathbf{c}_1^T\mathbf{s}) \bmod q$. If $|\Delta| < \lfloor q/2 \rceil/2$, output 0. Otherwise, output 1.

Theorem 5 (Weak correctness). *Let $\lambda \in (\omega(\log n), O(n^{<1})]$, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ be a prime such that $q > \omega(\lambda)$, $k = \Theta(\lambda \log \lambda / \log n)$ s.t. $H_\infty(\Xi^{[n:k]}) > 2(\lambda+1)\log q$, $\alpha \leq \frac{1}{10\sqrt{k}}$. Then Π_λ^{LWE} has $\frac{2}{3}$-correctness.*

Proof. $\Delta = (c_2 - \mathbf{c}_1^T\mathbf{s}) \bmod q = (\mathbf{r}^T\mathbf{e} + \lfloor q/2 \rceil \cdot \mathfrak{m}) \bmod q$. Let $e' := \mathbf{r}^T\mathbf{e}$, then $e' \sim (\Psi_{\alpha,q})^{\star k}$. By Corollary 2, $\Pr[|e'| < \lfloor q/2 \rceil/2] > 1 - \exp(-\frac{(100k)\pi}{72k}) > 2/3$. □

The CPA security of Π_λ^{LWE} follows straightforwardly from the hardness of decision $\mathsf{LWE}(\lambda, q, \alpha)$ and an application of LHL (the leftover hash lemma, Lemma 7), and we postpone the formal proof to Appendix B.

Theorem 6 (CPA security). *Let $\lambda \in (\omega(\log n), O(n^{<1})]$, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ such that $q > \omega(\lambda)$, $k = \Theta(\lambda \log \lambda / \log n)$ s.t. $H_\infty(\Xi^{[n:k]}) > 2(\lambda+1)\log q$, $\alpha \leq \frac{1}{10\sqrt{k}}$. Assume decision $\mathsf{LWE}(\lambda, q, \alpha)$ is hard, then Π_λ^{LWE} is IND-CPA secure.*

3.2 Multi-Bit LWE-PKE Scheme with Strong Correctness

In this subsection, we transform the scheme Π_λ^{LWE} in subsection 3.1 into a strongly correct one, denoted by $\widetilde{\Pi}_\lambda^{LWE} = (\widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Dec}})$. The idea is to use a similar transformation as in [PVW08, Subsection 7.2], along with an error-correcting code.

Construction Let $\lambda \in (\omega(\log n), O(n^{<1})]$, $q = q(\lambda) \in (\omega(\lambda), \mathsf{poly}(\lambda))$ be a prime, $\ell = O(\lambda)$, $k = \Theta((\lambda+\ell)\log\lambda/\log n)$ s.t. $H_\infty(\Xi^{[n:k]}) > 2(\lambda+\ell)\log q$. Let $\chi_r \sim \Xi^{[n;k]}$, $\alpha \leq \frac{1}{10\sqrt{k}}$. Let $\mathbf{C} \in \{0,1\}^{\ell \times m}$ be a binary error correcting code that corrects $\ell/3$ independent errors, where $m = O(\ell)$. (The construction of such error-correcting code can be found in [Jus72].)

- The message space is $\mathcal{M} = \{0,1\}^m$.
- $\widetilde{\mathsf{KeyGen}}(1^n)$: Let $\mathbf{A} \leftarrow \mathbb{Z}_q^{\lambda \times n}$. Let $\mathbf{s}_1, \ldots, \mathbf{s}_\ell \leftarrow \mathbb{Z}_q^\lambda$, $\mathbf{e}_1, \ldots, \mathbf{e}_\ell \leftarrow (\Psi_{\alpha,q})^n$, and set $\mathbf{S} := [\mathbf{s}_1 \| \ldots \| \mathbf{s}_\ell] \in \mathbb{Z}_q^{\lambda \times \ell}$, $\mathbf{E} := [\mathbf{e}_1 \| \ldots \| \mathbf{e}_\ell] \in \mathbb{Z}^{n \times \ell}$. Compute $\mathbf{B} := (\mathbf{A}^T\mathbf{S} + \mathbf{E}) \bmod q$, and set $(pk, sk) := ((\mathbf{A}, \mathbf{B}), \mathbf{S})$.
- $\widetilde{\mathsf{Enc}}(pk, \mathfrak{m})$: Given the public key $pk = (\mathbf{A}, \mathbf{B})$ and the message $\mathfrak{m} \in \mathcal{M}$, it samples $\mathbf{r} \leftarrow \chi_r$, and output $\mathfrak{c} := (\mathbf{c}_1^T, \mathbf{c}_2)$ as ciphertext, where $\mathbf{c}_1 := \mathbf{A}\mathbf{r}$ and $\mathbf{c}_2 := (\mathbf{r}^T\mathbf{B} + \lfloor q/2 \rceil \cdot \mathbf{C} \cdot \mathfrak{m}) \bmod q$.
- $\widetilde{\mathsf{Dec}}(sk, \mathfrak{c})$: Parse $\mathfrak{c}$ into $(\mathbf{c}_1, \mathbf{c}_2)$, calculate $(\Delta_1, \ldots, \Delta_\ell)^T := (\mathbf{c}_2 - \mathbf{c}_1^T \cdot \mathbf{S}) \bmod q$. Define $\mathbf{d} := (d_1, \ldots, d_\ell)^T$, where

$$\forall 1 \le i \le \ell,\ d_i := \begin{cases} 0 \text{ if } |\Delta_i| < \lfloor q/2 \rceil / 2 \\ 1 \quad \text{otherwise.} \end{cases}$$

  Output $\mathbf{C}^{-1} \cdot \mathbf{d}$, where $\mathbf{C}^{-1}$ is the decoding function.

Theorem 7 (Strong correctness). *Let $\lambda \in (\omega(\log n), O(n^{<1})]$, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ be a prime such that $q > \omega(\lambda)$, $\ell = O(\lambda)$, $k = \Theta((\lambda+\ell)\log\lambda/\log n)$ s.t. $H_\infty(\Xi^{[n:k]}) > 2(\lambda+\ell)\log q$. Let $\chi_r \sim \Xi^{[n;k]}$, $\alpha \le \frac{1}{10\sqrt{k}}$. Then $\widetilde{\Pi}_\lambda^{LWE}$ is correct.*

Proof (sketch). Assume $(u_1, \ldots, u_\ell)^T = \mathbf{C} \cdot \mathfrak{m}$ Through a calculation similar to the proof in Theorem 5, we have $\Pr[d_i = u_i] > 2/3$ for every $i \in [1, \ell]$. Therefore, we can successfully decode $\mathfrak{m}$ from $\mathbf{d}$ with overwhelming probability. □

Theorem 8 (CPA security). *Let $\lambda \in (\omega(\log n), O(n^{<1})]$, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ be a prime such that $q > \omega(\lambda)$, $\ell = O(\lambda)$, $k = \Theta((\lambda+\ell)\log\lambda/\log n)$ s.t. $H_\infty(\Xi^{[n:k]}) > 2(\lambda+\ell)\log q$. Let $\chi_r \sim \Xi^{[n;k]}$, $\alpha \le \frac{1}{10\sqrt{k}}$. Assume decision $\mathsf{LWE}(\lambda, q, \alpha)$ is hard, then $\widetilde{\Pi}_\lambda^{LWE}$ is IND-CPA secure.*

Proof (sketch). The hybrid argument is similar to the proof of Theorem 6. First, by the hardness of $\mathsf{LWE}(\lambda, q, \alpha)$, we can replace every column of $\mathbf{B}$ by a uniform vector sequentially. Then, an application of LHL (Lemma 7) concludes the proof. □

4 Nearest-Plane Algorithm for Large Noise Vectors

In this section, we give a generalization of Babai's nearest-plane algorithm [Bab86], which finds a "common closest lattice point" for a set of real vectors. More concretely, let $\tilde{\mathbf{B}} = (\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_\kappa)$ be an orthogonal basis of some lattice L. For any vector $\mathbf{v} \in L$, consider a set of independent error vectors $\{\mathbf{e}^{(i)} \in \mathbb{R}^\kappa\}_{1 \le i \le \ell}$. For each i, define $\mathbf{z}^{(i)} := \mathbf{v} + \mathbf{e}^{(i)}$. We show in the following theorem that, if for every i and j, the length of the projection of $\mathbf{e}^{(i)}$ onto $\tilde{\mathbf{b}}_j$ is less than $\|\tilde{\mathbf{b}}_j\|_2/2$ with a probability greater than $2/3$, then $\mathbf{v}$ and $\{\mathbf{e}^{(i)}\}_{1 \le i \le \ell}$ can be recovered with overwhelming probability.

Theorem 9. *Let $\kappa = \kappa(n) \in \mathsf{poly}(n)$, $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_\kappa)$ be an ordered basis of any κ-dimensional lattice L, and let its orthogonal basis be $\tilde{\mathbf{B}} = (\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_\kappa)$.*

Let $\ell = \omega(\log n)$. For random vectors $\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(\ell)}$ sampled independently from some distribution over $\mathbb{R}^\kappa$, if

$$\forall 1 \le i \le \ell, 1 \le j \le \kappa,\ \Pr_{\mathbf{e}^{(i)}}\left[|\langle \mathbf{e}^{(i)}, \tilde{\mathbf{b}}_j \rangle| < \frac{\|\tilde{\mathbf{b}}_j\|^2}{2}\right] > \frac{2}{3},$$

then there is a PPT algorithm, which takes $\mathbf{B}$ and $\{\mathbf{v}+\mathbf{e}^{(i)}\}_{1\le i\le \ell}$ for some $\mathbf{v} \in L$ as input, outputs $\mathbf{v}$ and $\{\mathbf{e}^{(i)}\}_{1\le i\le \ell}$ correctly with $1 - \mathsf{negl}(n)$ probability.

Algorithm 1 Nearest-Plane Algorithm for Larger Noise Vectors

function FindClosestVector$^{\kappa,\ell}(L, \mathbf{B}, \{\mathbf{z}^{(i)}\}_{1\le i\le \ell})$
  Orthogonalize $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_\kappa)$ in the forward order and get $\tilde{\mathbf{B}} = (\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_\kappa)$.
  for $i = 1$ **to** ℓ **do**
    Initialize $\mathbf{x}^{(i)} := \mathbf{z}^{(i)}$
  end for
  for $j = \kappa$ **to** 1 **do**
    Find majority in $\{\lfloor \frac{\langle \mathbf{x}^{(i)}, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle} \rceil\}_{1\le i\le \ell}$, and denote it by c_j.
    for $i = 1$ **to** ℓ **do**
      $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - c_j \mathbf{b}_j$.
    end for
  end for
  Let $\mathbf{v} := \mathbf{z}^{(1)} - \mathbf{x}^{(1)}$, and $\mathbf{e}^{(i)} := \mathbf{x}^{(i)}$ for each $1 \le i \le \ell$.
  return $\mathbf{v}, \{\mathbf{e}^{(i)}\}_{1\le i\le \ell}$.
end function

Proof. The detailed steps of the algorithm are given in Algorithm 1, and we show its correctness in the following. Assume the hidden lattice vector is $\mathbf{v} = \sum_{j=1}^{\kappa} c'_j \mathbf{b}_j$, where $c'_j \in \mathbb{Z}$. We abuse the notation that as j goes from κ to 1, we denote by $\mathbf{x}_j^{(i)}$ the value of $\mathbf{x}^{(i)}$ at the *beginning* of each corresponding iteration. Denote by $\mathbf{x}_0^{(i)}$ the value of $\mathbf{x}^{(i)}$ *after* all κ iterations. If we have with overwhelming probability that

$$\mathbf{x}_0^{(i)} = \mathbf{e}^{(i)} \wedge (\forall 1 \le j \le \kappa,\ \mathbf{x}_j^{(i)} = \sum_{h=1}^{j} c'_h \mathbf{b}_h + \mathbf{e}^{(i)}),$$

then we can get the correct $\mathbf{v}$ by subtracting $\mathbf{e}^{(1)}$ from $\mathbf{z}^{(1)}$. To show this, we only need to prove $c'_j = c_j$ with overwhelming probability for all $1 \le j \le \kappa$.

We prove this by induction on j downward from κ to 1. For every $j = \kappa - 1, \ldots, 1$, assume for all $h = j+1, \ldots, \kappa$, we have $c_h = c'_h$ holds with overwhelming probability. (In the base case where $j = \kappa$, we do not make any assumption.) Then we have for $i = 1, \ldots, \ell$, $\mathbf{x}_j^{(i)} = \sum_{h=1}^{j} c'_h \mathbf{b}_h + \mathbf{e}^{(i)}$. By the property of

Gram-Schmidt orthogonalization, it holds that

$$\langle \mathbf{x}_j^{(i)}, \tilde{\mathbf{b}}_j \rangle = \langle \sum_{h=1}^{j} c'_h \mathbf{b}_h + \mathbf{e}^{(i)}, \tilde{\mathbf{b}}_j \rangle = \langle c'_j \tilde{\mathbf{b}}_j + \mathbf{e}^{(i)}, \tilde{\mathbf{b}}_j \rangle.$$

Let $\tilde{\mathbf{e}}_j^{(i)} := \frac{\langle \mathbf{e}^{(i)}, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle} \cdot \tilde{\mathbf{b}}_j$ be the projection of $\mathbf{e}^{(i)}$ to $\tilde{\mathbf{b}}_j$ for each $j = 1, \ldots, \kappa$. It follows that $\langle \mathbf{x}_j^{(i)}, \tilde{\mathbf{b}}_j \rangle = \langle c'_j \tilde{\mathbf{b}}_j + \tilde{\mathbf{e}}_j^{(i)}, \tilde{\mathbf{b}}_j \rangle$, which gives $\lfloor \frac{\langle \mathbf{x}_j^{(i)}, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle} \rceil = c'_j + \lfloor \frac{\|\tilde{\mathbf{e}}_j^{(i)}\|_2}{\|\tilde{\mathbf{b}}_j\|_2} \rceil$. Additionally, we have by our assumption that

$$\Pr[|\langle \tilde{\mathbf{e}}_j^{(i)}, \tilde{\mathbf{b}}_j \rangle| < \frac{\|\tilde{\mathbf{b}}_j\|_2^2}{2}] = \Pr[|\langle \mathbf{e}^{(i)}, \tilde{\mathbf{b}}_j \rangle| < \frac{\|\tilde{\mathbf{b}}_j\|_2^2}{2}] > \frac{2}{3},$$

therefore, $\Pr[\frac{\|\tilde{\mathbf{e}}_j^{(i)}\|_2}{\|\tilde{\mathbf{b}}_j\|_2} < 1/2] > 2/3$. Define $\xi_j^{(i)}$ as the indicator function for the event that $\lfloor \frac{\|\tilde{\mathbf{e}}_j^{(i)}\|_2}{\|\tilde{\mathbf{b}}_j\|_2} \rceil = 0$. Then we have $\Pr[\xi_j^{(i)} = 1] > 2/3$. Let $\xi_j = \sum_{i=1}^{\ell} \xi_j^{(i)}$, we have $\mathbb{E}[\xi_j] > 2\ell/3$. By Chernoff bound, we have

$$\Pr[\xi_j > \ell/2] > 1 - e^{-\frac{1}{2} \cdot \frac{2\ell}{3} \cdot (\frac{1}{4})^2} = 1 - n^{-\omega(1)}.$$

Therefore, with overwhelming probability, for more than a half of $i \in [1, \ell]$, the event $\lfloor \frac{\|\tilde{\mathbf{e}}_j^{(i)}\|_2}{\|\tilde{\mathbf{b}}_j\|_2} \rceil = 0$ happens. And then we can get $c_j = c'_j$ by taking a majority in $\{\lfloor \frac{\langle \mathbf{x}^{(i)}, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle} \rceil\}_{1 \le i \le \ell}$. □

5 Injective Trapdoor Functions from Large Noise LWE

This section focuses on proving the following theorem.

Theorem 10 (Theorem 2). *Let n be the security parameter, and $q = q(n) \in \mathsf{poly}(n)$ be a prime such that $q > \omega(n)$. There exists a construction of TDF family assuming one of the following three conditions.*

(i) The $(n^{\omega(1)}, n^{-\omega(1)})$-hardness of search $\mathsf{LWE}(n, q(n), O(1/\sqrt{n}))$.

(ii) The $(2^{\omega(n^{\frac{1}{c_1}})}, 2^{-\omega(n^{\frac{1}{c_1}})})$-hardness of search $\mathsf{LWE}(n, q(n), O(1/\sqrt{n^{1-\frac{1}{c_1}} \log n}))$, for some constant $c_1 > 1$.

(iii) The $(2^{\omega(\frac{n}{\log^{c_2} n})}, 2^{-\omega(\frac{n}{\log^{c_2} n})})$-hardness of search $\mathsf{LWE}(n, q(n), O(1/\sqrt{\log^{c_2+1} n}))$, for some constant $c_2 > 0$.

Specifically, assuming condition (i) or condition (ii) with $c_1 \ge 2$, we have the TDF family is injective.

In order to prove Theorem 10, we generalize the trapdoor in [MP12], and construct a TDF family $\mathcal{T}_\lambda$ from search $\mathsf{LWE}(\lambda, q, \alpha)$, where $\lambda \in (\omega(\log n), O(n^{<1})]$, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ is a prime such that $q > \omega(\lambda)$, $k = \Theta(\lambda \log \lambda / \log n)$,

$\alpha \leq \frac{1}{10\sqrt{k}}$. When $\lambda = \Omega(\log^2 n)$, we prove the injectivity of $\mathcal{T}_\lambda$. The result in Theorem 10 then follows straightforwardly by applying Lemma 2.

The organization of this section is as follows. In Subsection 5.1, we revisit the concepts of primitive lattices from [MP12] and establish several bounds pertinent to our analysis. In Subsection 5.2, we show how to invert LWE on primitive lattices. In Subsection 5.3, we show how to generate an LWE instance with auxiliary information (trapdoor) that enables us to recover the secret and the noise. In Subsection 5.4, we give the formal construction of injective trapdoor function family $\mathcal{T}_\lambda$, as well as the proof of its injectivity and invertibility. We also remark that the conclusions in Subsection 5.1 and Subsection 5.2 do not require q to be prime.

5.1 Primitive Lattices

We first recall some notions in [MP12]. Let $\kappa = \lceil \log_2 q \rceil$, and define $\mathbf{g} := (1, 2, \ldots, 2^{\kappa-1})^T$. A short basis for $\Lambda^\perp(\mathbf{g}^T)$ (the q-ary lattices is defined in Eqn. (2)), denoted by $\mathbf{T}_\mathbf{g}$, is constructed as follows.

- When $q = 2^\kappa$, let

$$\mathbf{T}_\mathbf{g} := (\mathbf{t}_1, \ldots, \mathbf{t}_\kappa) := \begin{bmatrix} 2 & & & \\ -1 & 2 & & \\ & -1 & \ddots & \\ & & & 2 \\ & & & -1 \; 2 \end{bmatrix}.$$

- Alternatively, if q is not a power of 2, let

$$\mathbf{T}_\mathbf{g} := (\mathbf{t}_1, \ldots, \mathbf{t}_\kappa) := \begin{bmatrix} 2 & & & q_0 \\ -1 & 2 & & q_1 \\ & -1 & \ddots & q_2 \\ & & 2 & q_{\kappa-2} \\ & & -1 & q_{\kappa-1} \end{bmatrix},$$

  where $(q_0, \ldots, q_{k-1}) \in \{0,1\}^\kappa$ is the bit decomposition of $q = \sum_{i=0}^{\kappa-1} 2^i \cdot q_i$.

Since $\mathbf{T}^*_\mathbf{g} := (\mathbf{t}^*_1, \ldots, \mathbf{t}^*_\kappa)$ is a basis of $\Lambda^\perp(\mathbf{g}^T)^*$, we have $q \cdot \mathbf{T}^*_\mathbf{g}$ is a basis for $\Lambda(\mathbf{g}) = q\Lambda^\perp(\mathbf{g}^T)^*$.

Define the gadget matrix as $\mathbf{G}_n := \mathsf{diag}(\mathbf{g}^T, \mathbf{g}^T, \ldots, \mathbf{g}^T) \in \mathbb{Z}_q^{n \times n\kappa}$, and let $\mathbf{T}_{\mathbf{G}_n} := \mathsf{diag}(\mathbf{T}_\mathbf{g}, \mathbf{T}_\mathbf{g}, \ldots, \mathbf{T}_\mathbf{g}) \in \mathbb{Z}_q^{n\kappa \times n\kappa}$, then we have $\mathbf{T}_{\mathbf{G}_n}$ is a basis for $\Lambda^\perp(\mathbf{G}_n)$. When $q = 2^\kappa$, orthogonalize $\mathbf{T}_\mathbf{g}$ in the reverse order, and we have $\tilde{\mathbf{T}}_\mathbf{g} = 2 \cdot \mathbf{I}_\kappa$. When q is not a power of 2, orthogonalize $\mathbf{T}_\mathbf{g}$ in the forward order, and we have $\tilde{\mathbf{T}}_\mathbf{g}$ is still a short basis by the following lemma.

Lemma 3. *When q is not a power of 2, let $\tilde{\mathbf{T}}_\mathbf{g} := (\tilde{\mathbf{t}}_1, \ldots, \tilde{\mathbf{t}}_\kappa)$ be the orthogonalization of $\mathbf{T}_\mathbf{g}$ in the forward order. It holds that*

- $\forall 1 \le i < \kappa,\ \|\tilde{\mathbf{t}}_i\|_1 = \frac{2^{i+2}+1}{2^i+1} \in [3, 4)$;
- $\|\tilde{\mathbf{t}}_\kappa\|_1 = \frac{3q}{2^\kappa+1} \in (\frac{3}{2}, 3)$;
- $\forall 1 \le i < \kappa,\ \|\tilde{\mathbf{t}}_i\|_2 = \sqrt{\frac{4^{i+1}-1}{4^i-1}} \in (2, \sqrt{5}]$;
- $\|\tilde{\mathbf{t}}_\kappa\|_2 = \sqrt{\frac{3q^2}{4^\kappa-1}} \in (\frac{\sqrt{3}}{2}, \sqrt{3})$.

In Lemma 3, we extend [MP12, Lemma 4.3] by figuring out more details on the bounds of both ℓ_1-norm and ℓ_2-norm for each $\tilde{\mathbf{t}}_i$. The proof is postponed to Appendix C.

5.2 Inverting Large Noise LWE on Primitive Lattices

For $\mathbf{s} \in \mathbb{Z}_q^n$ and specific errors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_\ell \in \mathbb{Z}_q^{n\kappa}$, let $\mathbf{b}_i := \mathbf{G}_n^T\mathbf{s} + \mathbf{e}_i$ for all $1 \le i \le \ell$. We then describe an algorithm that, given $(\mathbf{b}_1, \ldots, \mathbf{b}_\ell)$, recovers $\mathbf{s}$ and $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_\ell$ with overwhelming probability.

For simplicity, we set $n = 1$ without loss of generality. (In cases where $n > 1$, we can decompose each $\mathbf{b}_i$ into n vectors in $\mathbb{Z}_q^\kappa$, apply the inversion algorithm to each of them, and then concatenate the results.) Now our task is to recover $s \in \mathbb{Z}_q$ and $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_\ell \in \mathbb{Z}_q^\kappa$ from $(\mathbf{b}_1 = s\mathbf{g} + \mathbf{e}_1, \ldots, \mathbf{b}_\ell = s\mathbf{g} + \mathbf{e}_\ell)$.

If we view $s\mathbf{g}$ as a lattice point on $\Lambda(\mathbf{g})$, then we can use the generalized nearest plane algorithm in section 4 to recover it. We further explain this in the following lemma.

Lemma 4. *Adopt the notations of* $\mathbf{g}, \mathbf{T}_\mathbf{g}, \mathbf{T}_\mathbf{g}^*, \tilde{\mathbf{T}}_\mathbf{g}$, *and* $\{\mathbf{t}_j, \mathbf{t}_j^*, \tilde{\mathbf{t}}_j\}_{1\le j\le\kappa}$ *in Subsection 5.1. If* $\mathbf{e}_1, \ldots, \mathbf{e}_\ell \in \mathbb{Z}^\kappa$ *are independent random vectors and satisfy*

$$\forall 1 \le i \le \ell, 1 \le j \le \kappa,\ \Pr_{\mathbf{e}_i}[|\langle \mathbf{e}_i, \tilde{\mathbf{t}}_j\rangle| < q/2] > \frac{2}{3},$$

then there is an algorithm that for any $s \in \mathbb{Z}_q$, *it takes* $(\mathbf{b}_1 = s\mathbf{g} + \mathbf{e}_1, \ldots, \mathbf{b}_\ell = s\mathbf{g} + \mathbf{e}_\ell)$ *as input, and recovers* $(s, \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_\ell)$ *with overwhelming probability.*

Particularly, when $q = 2^\kappa$, *the condition is simplified as*

$$\forall 1 \le i \le \ell, 1 \le j \le \kappa,\ \Pr_{\mathbf{e}_i}[|e_i(j)| < q/4] > \frac{2}{3}.$$

Proof. Since $s\mathbf{g} \in \Lambda(\mathbf{g})$, we can view it as a "common closest lattice vector" of $\{\mathbf{b}_i\}_{1\le i\le\ell}$. Denote by $\tilde{\mathbf{T}}_\mathbf{g}^*$ the orthogonalization of $\mathbf{T}_\mathbf{g}^*$. Since $q \cdot \mathbf{T}_\mathbf{g}^*$ is a basis for $\Lambda(\mathbf{g})$, we have $q \cdot \tilde{\mathbf{T}}_\mathbf{g}^*$ is an orthogonal basis of $\Lambda(\mathbf{g})$. By Theorem 9, we only need to prove

$$\forall 1 \le i \le \ell, 1 \le j \le \kappa, \Pr_{\mathbf{e}_i}[|\langle \mathbf{e}_i, q\tilde{\mathbf{t}}_j^*\rangle| < \|q\tilde{\mathbf{t}}_j^*\|^2/2] > 2/3,$$

which is easy to prove since

$$\Pr_{\mathbf{e}_i}[|\langle \mathbf{e}_i, q\tilde{\mathbf{t}}_j^*\rangle| < \|q\tilde{\mathbf{t}}_j^*\|^2/2] = \Pr_{\mathbf{e}_i}[|\langle \mathbf{e}_i, \tilde{\mathbf{t}}_j\rangle| < q/2] > 2/3,$$

where the equation holds by $\tilde{\mathbf{t}}_j = \frac{\tilde{\mathbf{t}}_j^*}{\|\tilde{\mathbf{t}}_j^*\|^2}$, and the "$>$" holds by assumption.

Particularly, when $q = 2^\kappa$, we do Gram-Schmidt orthogonalization in the reverse order and get $\tilde{\mathbf{T}}_{\mathbf{g}} = 2\mathbf{I}$. Then $\tilde{\mathbf{T}}_{\mathbf{g}}^* = \mathbf{I}/2$, and thus

$$\forall 1 \le i \le \ell, 1 \le j \le \kappa, \Pr_{\mathbf{e}_i}[|\langle \mathbf{e}_i, \tilde{\mathbf{t}}_j \rangle| < q/2] = \Pr_{\mathbf{e}_i}[|e_i(j)| < q/4].$$

Substitute this back into the condition and then we finish the proof. □

For completeness, we present the inversion process for $\mathbf{b} = \mathbf{G}_n^T\mathbf{s} + \mathbf{e}$ in Algorithm 2.

Algorithm 2 Inverting Large Noise LWE on Primitive Lattices

function InvertPrimitiveLWE$^{n,\kappa,\ell}(\{\mathbf{b}_i\}_{1\le i\le \ell})$
  if $q = 2^\kappa$ **then**
    Let $\hat{\mathbf{T}}_{\mathbf{g}}^* := (\mathbf{t}_\kappa^*, \mathbf{t}_{\kappa-1}^*, \ldots, \mathbf{t}_1^*)$.
    Let $\hat{\mathbf{T}}_{\mathbf{G}_n}^* := \mathsf{diag}(\hat{\mathbf{T}}_{\mathbf{g}}^*, \hat{\mathbf{T}}_{\mathbf{g}}^*, \ldots, \hat{\mathbf{T}}_{\mathbf{g}}^*) \in \mathbb{Z}_q^{n\kappa \times n\kappa}$
    return FindClosestVector$(\Lambda(\mathbf{G}_n^T), \hat{\mathbf{T}}_{\mathbf{G}_n}^*, \{\mathbf{b}_i\}_{1\le i\le \ell})$
  else
    return FindClosestVector$(\Lambda(\mathbf{G}_n^T), \mathbf{T}_{\mathbf{G}_n}^*, \{\mathbf{b}_i\}_{1\le i\le \ell})$
  end if
end function

5.3 Generating Large Noise LWE with Trapdoor

Trapdoor Generation Let $\lambda \in (\omega(\log n), O(n^{<1})]$, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ be a prime such that $q > \omega(\lambda)$, $k = \Theta(\lambda \log \lambda / \log n)$, $\alpha \le \frac{1}{10\sqrt{k}}$, $\ell = \omega(\log n)$, $\kappa = \lceil \log q \rceil$, $w = \lambda\kappa$, $\bar{n} = n - \ell w$. Let χ_r be some distribution on $\mathbb{Z}_q^{\bar{n}}$ such that $H_\infty(\chi_r) > 2\lambda \log q$. We present the trapdoor generation algorithm in Algorithm 3, which takes as input a matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{\lambda \times \bar{n}}$ and an invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{\lambda\times\lambda}$, (If there is no input, it randomly picks $\bar{\mathbf{A}} \leftarrow \mathbb{Z}_q^{\lambda\times\bar{n}}$, and set $\mathbf{H} = \mathbf{I}$.) and outputs a pseudo random matrix $\mathbf{A} \in \mathbb{Z}_q^{\lambda\times n}$ with trapdoor $(\mathbf{R}_1, \ldots, \mathbf{R}_\ell)$.

Theorem 11. *Let $\lambda \in (\omega(\log n), O(n^{<1})]$, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ be a prime such that $q > \omega(\lambda)$, $k = \Theta(\lambda \log \lambda / \log n)$, $\alpha \le \frac{1}{10\sqrt{k}}$, $\ell = \omega(\log n)$, $\kappa = \lceil \log q \rceil$, $w = \lambda\kappa$, $\bar{n} = n - \ell w$. Let χ_r be some distribution on $\mathbb{Z}_q^{\bar{n}}$ such that $H_\infty(\chi_r) > 2\lambda \log q$. For a random matrix $\bar{\mathbf{A}} \leftarrow \mathbb{Z}_q^{\lambda\times\bar{n}}$, and an invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{\lambda\times\lambda}$, let $(\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell)) \leftarrow$ TrapGen$^{n,\lambda,\ell,\chi_r}(\bar{\mathbf{A}}, \mathbf{H})$. Then $\mathbf{A}$ is statistically indistinguishable from uniform.*

Proof. By Lemma 7, we have

$$\mathsf{SD}((\bar{\mathbf{A}}, (\bar{\mathbf{A}}\mathbf{R}_1, \bar{\mathbf{A}}\mathbf{R}_2, \ldots, \bar{\mathbf{A}}\mathbf{R}_\ell)), (\bar{\mathbf{A}}, (\mathcal{U}_q^{\lambda\times w})^{\otimes\ell})) \le 2^{(\lambda w\ell \log q - w\ell H_\infty(\chi_r))/2} = \mathsf{negl}(n).$$

So

$$\mathsf{SD}((\bar{\mathbf{A}}, (\mathbf{H}\mathbf{G}_\lambda - \bar{\mathbf{A}}\mathbf{R}_1, \mathbf{H}\mathbf{G}_\lambda - \bar{\mathbf{A}}\mathbf{R}_2, \ldots, \mathbf{H}\mathbf{G}_\lambda - \bar{\mathbf{A}}\mathbf{R}_\ell)), (\bar{\mathbf{A}}, (\mathcal{U}_q^{\lambda\times w})^{\otimes\ell})) = \mathsf{negl}(n).$$

Then we have $\mathbf{A}$ is statistically indistinguishable from uniform. □

Algorithm 3 Trapdoor Generation

function $\textsc{TrapGen}^{n,\lambda,\ell,\chi_r}(\bar{\mathbf{A}}, \mathbf{H})$
  Sample $r_1, \ldots, r_{w\ell} \leftarrow \chi_r$.
  for $i = 1$ **to** ℓ **do**
    Let $\mathbf{R}_i = (r_{(i-1)w+1} \| \ldots \| r_{iw})$.
    Compute $\underline{\mathbf{A}}_i = \mathbf{H}\mathbf{G}_\lambda - \bar{\mathbf{A}}\mathbf{R}_i$
  end for
  Compute $\mathbf{A} := [\bar{\mathbf{A}} \| \underline{\mathbf{A}}_1 \| \underline{\mathbf{A}}_2 \| \ldots \| \underline{\mathbf{A}}_\ell]$
  return $\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell)$
end function

Inversion Assume $(\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell)) \leftarrow \mathsf{TrapGen}(\bar{\mathbf{A}}, \mathbf{H})$ is generated by the algorithm above. Let $\mathbf{s} \leftarrow \mathbb{Z}_q^\lambda, \mathbf{e} \leftarrow \Psi_{\alpha,q}^n$. For LWE instance $\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$, we show that Algorithm 4 takes as input $\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell), \mathbf{H}, \mathbf{b}$, and outputs the correct $\mathbf{s}$ and $\mathbf{e}$ with $1 - \mathsf{negl}(n)$ probability.

Algorithm 4 Inverting LWE with Trapdoor

function $\textsc{InvertLWE}^{n,\lambda,\ell}(\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell), \mathbf{H}, \mathbf{b})$
  Parse $\mathbf{b} = (\bar{\mathbf{b}}^T, \underline{\mathbf{b}}_1^T, \ldots, \underline{\mathbf{b}}_\ell^T)^T \in \mathbb{Z}_q^{\bar{n}} \times (\mathbb{Z}_q^w)^{\otimes \ell}$
  for $i = 1$ **to** ℓ **do**
    Compute $\hat{\mathbf{b}}_i := \mathbf{R}_i^T\bar{\mathbf{b}} + \underline{\mathbf{b}}_i$
  end for
  Let $(\hat{\mathbf{s}}, \{\hat{\mathbf{e}}_i\}_{1 \le i \le \ell}) \leftarrow \textsc{InvertPrimitiveLWE}^{\lambda,\kappa,\ell}(\{\hat{\mathbf{b}}_i\}_{1 \le i \le \ell})$
  Compute $\mathbf{s} := \mathbf{H}^{-T}\mathbf{s}$.
  Compute $\mathbf{e} := \mathbf{b} - \mathbf{A}^T\mathbf{s}$
  return $\mathbf{s}, \mathbf{e}$
end function

We begin our analysis of Algorithm 4 by parsing $\mathbf{e} = (\bar{\mathbf{e}}^T, \underline{\mathbf{e}}_1^T, \ldots, \underline{\mathbf{e}}_\ell^T)^T \in \mathbb{Z}^{\bar{n}} \times (\mathbb{Z}^w)^{\otimes \ell}$ for clarity. This yields $\bar{\mathbf{b}} = \bar{\mathbf{A}}^T\mathbf{s} + \bar{\mathbf{e}}$ and $\underline{\mathbf{b}}_i = \underline{\mathbf{A}}^T\mathbf{s} + \underline{\mathbf{e}}_i$ for each $1 \le i \le \ell$. Then $\hat{\mathbf{b}}_i := \mathbf{R}_i^T\bar{\mathbf{b}} + \underline{\mathbf{b}}_i = \mathbf{R}_i^T\bar{\mathbf{A}}^T\mathbf{s} + \mathbf{R}_i^T\bar{\mathbf{e}} + \underline{\mathbf{A}}^T\mathbf{s} + \underline{\mathbf{e}}_i = \mathbf{G}_\lambda^T\mathbf{H}^T\mathbf{s} + \mathbf{R}_i^T\bar{\mathbf{e}} + \underline{\mathbf{e}}_i$. Let $\hat{\mathbf{s}} = \mathbf{H}^T\mathbf{s}$, $\hat{\mathbf{e}}_i = \mathbf{R}_i^T\bar{\mathbf{e}} + \underline{\mathbf{e}}_i$, the remaining thing to do is giving a bound for every $\hat{\mathbf{e}}_i$. Concretely, parse $\hat{\mathbf{e}}_i = (\hat{\mathbf{e}}_{i,1}^T \| \ldots \| \hat{\mathbf{e}}_{i,\lambda}^T)^T \in (\mathbb{Z}^\kappa)^{\otimes \lambda}$. For every $h = 1, \ldots, \lambda$, we need to prove $\Pr[\langle \hat{\mathbf{e}}_{i,h}, \tilde{\mathbf{t}}_j \rangle < q/2] > 2/3$, which enables us to recover $\hat{s}(h)$ by Lemma 4 (note that $\hat{\mathbf{s}} = (\hat{s}(1), \ldots, \hat{s}(h))$). Then the secret $\mathbf{s}$ can be calculated by $\mathbf{s} = \mathbf{H}^{-T}\hat{\mathbf{s}}$.

We first give a bound for $\mathbf{R}_i\bar{\mathbf{e}}$. Attentive readers may have realized that the distribution of $\mathbf{R}_i$ (or χ_r) has not yet been specified. Here, we need every $\mathbf{R}_i\bar{\mathbf{e}}$ to have an independent bound conditioned on the choice of $\bar{\mathbf{e}}$. Therefore, instead of letting $\chi_r \sim \Xi^{[\bar{n}:k]}$ as in our PKE construction, we let $\chi_r \sim \tilde{\mathcal{B}}_{k/\bar{n}}^{\bar{n}}$, namely the truncated Bernoulli distribution (see Definition 10). This enables us to bound the projection length of $\mathbf{R}_i\bar{\mathbf{e}}$ on $\tilde{\mathbf{t}}_j$, which is presented in the following lemma.

Lemma 5 (Bound for $\mathbf{R}_i\bar{\mathbf{e}}$). *Let $\lambda \in (\omega(\log n), O(n^{<1}))$, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ be a prime such that $q > \omega(\lambda)$, $\ell = \omega(\log n)$, $\kappa = \lceil \log q \rceil$, $w = \lambda\kappa$, $\bar{n} = n - \ell w$, $k = \Theta(\lambda \log \lambda / \log n)$ s.t. $H_\infty(\tilde{\mathcal{B}}^{\bar{n}}_{k/\bar{n}}) > 2\lambda \log q$. Let $\mathbf{r}_1, \ldots, \mathbf{r}_\kappa \leftarrow \tilde{\mathcal{B}}^{\bar{n}}_{k/\bar{n}}$, and $\mathbf{R} := (\mathbf{r}_1 \| \ldots \| \mathbf{r}_\kappa)$. For any fixed $\mathbf{e}' \in \mathbb{R}^{\bar{n}}$ with the following two properties,*

(i) $|\sum_{i=1}^{\bar{n}} e'(i)| < 0.01\sqrt{\frac{\bar{n}\log\bar{n}}{k}}$,
(ii) $\sum_{i=1}^{\bar{n}} e'(i)^2 < \frac{0.16\bar{n}}{100k}$,

let $\bar{\mathbf{e}} = \lfloor q\mathbf{e}' \rceil$, and we have $\forall 1 \le j \le \kappa$, $\Pr[\langle \mathbf{R}^T\bar{\mathbf{e}}, \tilde{\mathbf{t}}_j \rangle < q/4] > 0.79$.

We postpone the proof of Lemma 5 to Appendix D.

In the following we show when we generate a matrix $\mathbf{A}$ with trapdoor $\mathbf{R}_1, \ldots, \mathbf{R}_\ell$ using $\chi_r \sim \tilde{\mathcal{B}}^{\bar{n}}_{k/\bar{n}}$ in Algorithm 3, we can invert large noise LWE with overwhelming probability by Algorithm 4.

Theorem 12 (Correctness of inversion). *Let $\lambda \in (\omega(\log n), O(n^{<1}))$, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ be a prime such that $q > \omega(\lambda)$, $\ell = \omega(\log n)$, $\kappa = \lceil \log q \rceil$, $w = \lambda\kappa$, $\bar{n} = n - \ell w$, $k = \Theta(\lambda \log \lambda / \log n)$ s.t. $H_\infty(\tilde{\mathcal{B}}^{\bar{n}}_{k/\bar{n}}) > 2\lambda \log q$, $\alpha \le 1/(10\sqrt{k})$. Let $\mathbf{s} \in \mathbb{Z}_q^\lambda, \mathbf{e} \sim \Psi^n_{\alpha,q}$. Assume $(\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell)) \leftarrow \textsc{TrapGen}^{n,\lambda,\ell,\tilde{\mathcal{B}}^{\bar{n}}_{k/\bar{n}}}(\bar{\mathbf{A}}, \mathbf{H})$, and let $\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$. Then*

$$\Pr[\textsc{InvertLWE}^{n,\lambda,\ell}(\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell), \mathbf{H}, \mathbf{b}) = (\mathbf{s}, \mathbf{e})] = 1 - \mathsf{negl}(n).$$

The proof of this theorem is just a combination of previous lemmas (Lemma 12, Lemma 5, Lemma 13 and Lemma 4), and we postpone its proof to Appendix E.

5.4 Construction of iTDF Family

Now, we are ready to construct iTDF family $\mathcal{T}_\lambda$. Let $\lambda \in [\Omega(\log^2 n), O(n^{<1})]$, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ be a prime such that $q > \omega(\lambda)$, $\ell = \omega(\log n)$, $\kappa = \lceil \log q \rceil$, $w = \lambda\kappa$, $\bar{n} = n - \ell w$, $k = \Theta(\lambda \log \lambda / \log n)$ such that $H_\infty(\tilde{\mathcal{B}}^{\bar{n}}_{k/\bar{n}}) > 2\lambda \log q$, $\alpha \le \frac{1}{10\sqrt{k}}$. Let $\mathcal{D} := \mathbb{Z}_q^\lambda \times [-q\alpha\sqrt{\log n}, q\alpha\sqrt{\log n}]^n$, $\mathcal{R} := \mathbb{Z}^n$ be the domain and range of $\mathcal{T}_\lambda$ respectively, and we construct its syntax as follows.

- $\mathsf{Sample}(1^n)$: Randomly sample a matrix $\bar{\mathbf{A}} \leftarrow \mathbb{Z}_q^{\lambda \times \bar{n}}$ and an invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{\lambda \times \lambda}$. Compute $(\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell)) \leftarrow \textsc{TrapGen}^{n,\lambda,\ell,\tilde{\mathcal{B}}^{\bar{n}}_{k/\bar{n}}}(\bar{\mathbf{A}}, \mathbf{H})$. Output $(ek, ik) = (\mathbf{A}, (\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell), \mathbf{H}))$.
- $\mathsf{Eval}(ek, (\mathbf{s}, \mathbf{e}))$ takes as input the evaluation key $ek = \mathbf{A}$ and $(\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^\lambda \times [-q\alpha\sqrt{\log n}, q\alpha\sqrt{\log n}]$, and outputs $\mathbf{y} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$.
- $\mathsf{Invert}(ik, \mathbf{y})$ takes as input an image $\mathbf{y} \in \mathbb{Z}^n$ and the inversion key $ik = (\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell), \mathbf{H})$. It computes $(\mathbf{s}, \mathbf{e}) \leftarrow \textsc{InvertLWE}^{n,\lambda,\ell}(\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell), \mathbf{H}, \mathbf{A}^T\mathbf{s} + \mathbf{e})$, and outputs $(\mathbf{s}, \mathbf{e})$ (or output $\perp$ if $\textsc{InvertLWE}$ returns $\perp$).

Theorem 13 (Injectivity). *Let $\lambda \in [\Omega(\log^2 n), O(n^{<1})]$, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ be a prime such that $q > \omega(\lambda)$, $\ell = \omega(\log n)$, $\kappa = \lceil \log q \rceil$, $w = \lambda\kappa$, $\bar{n} = n - \ell w$,*

$k = \Theta(\lambda \log \lambda / \log n)$ *such that* $H_\infty(\tilde{\mathcal{B}}^{\bar{n}}_{k/\bar{n}}) > 2\lambda \log q$, $\alpha \le \frac{1}{10\sqrt{k}}$. *Then*

$$\Pr_{(ek,ik)\leftarrow \mathsf{Sample}(1^n)}\Big[\exists \mathbf{s}_1, \mathbf{s}_2 \in \mathbb{Z}_q^\lambda, \mathbf{e}_1, \mathbf{e}_2 \in [-q\alpha\sqrt{\log n}, q\alpha\sqrt{\log n}]^n \text{ s.t. } (\mathbf{s}_1, \mathbf{e}_1) \neq (\mathbf{s}_2, \mathbf{e}_2) \wedge \mathsf{Eval}(ek, (\mathbf{s}_1, \mathbf{e}_1)) = \mathsf{Eval}(ek, (\mathbf{s}_2, \mathbf{e}_2))\Big] = \mathsf{negl}(n). \tag{1}$$

Proof. We have

$$\begin{aligned}
&\text{the left hand side of Eqn. (1)}\\
=&\mathsf{negl}(n) + \Pr_{\mathbf{A}\leftarrow\mathbb{Z}_q^{\lambda\times n}}\Big[\exists \mathbf{s}_1, \mathbf{s}_2 \in \mathbb{Z}_q^\lambda, \mathbf{e}_1, \mathbf{e}_2 \in [-q\alpha\sqrt{\log n}, q\alpha\sqrt{\log n}]^n \text{ s.t.}\\
&(\mathbf{s}_1, \mathbf{e}_1) \neq (\mathbf{s}_2, \mathbf{e}_2) \wedge \mathbf{A}^T\mathbf{s}_1 + \mathbf{e}_1 = \mathbf{A}^T\mathbf{s}_2 + \mathbf{e}_2\Big]\\
<&\mathsf{negl}(n) + \Pr_{\mathbf{A}\leftarrow\mathbb{Z}_q^{\lambda\times n}}\Big[\exists \mathbf{s}_1, \mathbf{s}_2 \in \mathbb{Z}_q^\lambda, \mathbf{e}_1, \mathbf{e}_2 \in [-q/8, q/8]^n \text{ s.t. } (\mathbf{s}_1, \mathbf{e}_1) \neq (\mathbf{s}_2, \mathbf{e}_2) \wedge \mathbf{A}^T\mathbf{s}_1 + \mathbf{e}_1 = \mathbf{A}^T\mathbf{s}_2 + \mathbf{e}_2\Big]\\
=&\mathsf{negl}(n) + \Pr_{\mathbf{A}\leftarrow\mathbb{Z}_q^{\lambda\times n}}\Big[\exists \mathbf{s}_1, \mathbf{s}_2 \in \mathbb{Z}_q^\lambda, \mathbf{e}_1, \mathbf{e}_2 \in [-q/8, q/8]^n \text{ s.t. } \mathbf{s}_1 \neq \mathbf{s}_2 \wedge \mathbf{A}^T\mathbf{s}_1 + \mathbf{e}_1 = \mathbf{A}^T\mathbf{s}_2 + \mathbf{e}_2\Big]\\
=&\mathsf{negl}(n) + \Pr_{\mathbf{A}\leftarrow\mathbb{Z}_q^{\lambda\times n}}\Big[\exists \mathbf{s} \in \mathbb{Z}_q^\lambda, \mathbf{e} \in [-q/4, q/4]^n \text{ s.t. } \mathbf{s} \neq \mathbf{0} \wedge \mathbf{A}^T\mathbf{s} = \mathbf{e}\Big]\\
<&\mathsf{negl}(n) + q^{-0.16\lambda} = \mathsf{negl}(n).
\end{aligned}$$

Here, the first "=" uses Theorem 11 ($ek = \mathbf{A}$ is statistically indistinguishable from uniform); the "<" follows from $\alpha q\sqrt{\log n} \le q\sqrt{\log n}/(10\sqrt{k}) = \Theta(q \log n/\sqrt{\lambda \log \lambda}) = O(q/\sqrt{\log\log n}) < q/8$; the second "=" is because when $\mathbf{s}_1 = \mathbf{s}_2$ and $\mathbf{e}_1 \neq \mathbf{e}_2$, the equation $A^T\mathbf{s}_1 + \mathbf{e}_1 = \mathbf{A}^T\mathbf{s}_2 + \mathbf{e}_2$ does not hold; the third "=" results from replacing $\mathbf{s}_1 - \mathbf{s}_2$ by $\mathbf{s}$ and $\mathbf{e}_1 - \mathbf{e}_2$ by $\mathbf{e}$; and the "$\le$" uses Lemma 6. □

Theorem 14 ($(\mathcal{U}_q^\lambda \times \tilde{\Psi}^n_{\alpha,q})$-invertibility). *Let* $\lambda \in (\omega(\log n), O(n^{<1})]$, $q = q(\lambda) \in \mathsf{poly}(\lambda)$ *such that* $q > \omega(\lambda)$, $\ell = \omega(\log n)$, $\kappa = \lceil \log q \rceil$, $w = \lambda\kappa$, $\bar{n} = n - \ell w$, $k = \Theta(\lambda \log \lambda / \log n)$ *such that* $H_\infty(\tilde{\mathcal{B}}^{\bar{n}}_{k/\bar{n}}) > 2\lambda \log q$, $\alpha \le \frac{1}{10\sqrt{k}}$. *Let* $(ek, ik) \leftarrow \mathsf{Sample}(1^n), \mathbf{s} \leftarrow \mathcal{U}_q^\lambda, \mathbf{e} \leftarrow \tilde{\Psi}^n_{\alpha,q}$. *It holds that*

(i)

$$\Pr_{\substack{(ek,ik)\leftarrow\mathsf{Sample}(1^n)\\ \mathbf{s}\leftarrow\mathcal{U}_q^\lambda, \mathbf{e}\leftarrow\tilde{\Psi}^n_{\alpha,q}}}[\mathsf{Invert}(ik, \mathsf{Eval}(ek, (\mathbf{s}, \mathbf{e}))) = (\mathbf{s}, \mathbf{e})] = 1 - \mathsf{negl}(n).$$

(ii) if search $\mathsf{LWE}(\lambda, q, \alpha)$ *is hard, then for any PPT adversary* $\mathcal{A}$

$$\Pr_{\substack{(ek,ik)\leftarrow\mathsf{Sample}(1^n)\\ \mathbf{s}\leftarrow\mathcal{U}_q^\lambda, \mathbf{e}\leftarrow\tilde{\Psi}^n_{\alpha,q}}}[\mathcal{A}(ek, \mathsf{Eval}(ek, (\mathbf{s}, \mathbf{e}))) = (\mathbf{s}, \mathbf{e})] = \mathsf{negl}(n).$$

Proof.

(i) Let $(\mathbf{A}, (\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell), \mathbf{H})) = (ek, ik) \leftarrow \mathsf{Sample}(1^n)$. We have

$$\begin{aligned}
&\Pr_{\mathbf{s}\leftarrow\mathcal{U}_q^\lambda, \mathbf{e}\leftarrow\tilde{\Psi}_{\alpha,q}^n}[\mathsf{Invert}(ik, \mathsf{Eval}(ek, (\mathbf{s}, \mathbf{e}))) = (\mathbf{s}, \mathbf{e})] \\
\geq &\Pr_{\mathbf{s}\leftarrow\mathcal{U}_q^\lambda, \mathbf{e}\leftarrow\Psi_{\alpha,q}^n}[\mathsf{Invert}(ik, \mathsf{Eval}(ek, (\mathbf{s}, \mathbf{e}))) = (\mathbf{s}, \mathbf{e})] - \mathsf{negl}(n) \\
= &\Pr_{\mathbf{s}\leftarrow\mathcal{U}_q^\lambda, \mathbf{e}\leftarrow\Psi_{\alpha,q}^n}[\textsc{InvertLWE}^{n,\lambda,\ell}(\mathbf{A}, (\mathbf{R}_1, \ldots, \mathbf{R}_\ell), \mathbf{H}, \mathbf{b}) = (\mathbf{s}, \mathbf{e})] - \mathsf{negl}(n) \\
= &1 - \mathsf{negl}(n),
\end{aligned}$$

where the first inequality is by definition of $\tilde{\Psi}_{\alpha,q}^n$, and the last equality is by Theorem 12.

(ii)

$$\begin{aligned}
&\Pr_{\substack{(ek,ik)\leftarrow\mathsf{Sample}(1^n) \\ \mathbf{s}\leftarrow\mathcal{U}_q^\lambda, \mathbf{e}\leftarrow\tilde{\Psi}_{\alpha,q}^n}}[\mathcal{A}(ek, \mathsf{Eval}(ek, (\mathbf{s}, \mathbf{e}))) = (\mathbf{s}, \mathbf{e})] \\
\leq &\Pr_{\substack{(ek,ik)\leftarrow\mathsf{Sample}(1^n) \\ \mathbf{s}\leftarrow\mathcal{U}_q^\lambda, \mathbf{e}\leftarrow\Psi_{\alpha,q}^n}}[\mathcal{A}(ek, \mathsf{Eval}(ek, (\mathbf{s}, \mathbf{e}))) = (\mathbf{s}, \mathbf{e})] + \mathsf{negl}(n) \\
\leq &\Pr_{\substack{\mathbf{A}\leftarrow\mathbb{Z}_q^{\lambda\times n} \\ \mathbf{s}\leftarrow\mathcal{U}_q^\lambda, \mathbf{e}\leftarrow\Psi_{\alpha,q}^n}}[\mathcal{A}(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e}) = (\mathbf{s}, \mathbf{e})] + \mathsf{negl}(n) \leq \mathsf{negl}(n),
\end{aligned}$$

where the first inequality is by definition of $\tilde{\Psi}_{\alpha,q}^n$, the second inequality is by Theorem 11 ($\mathbf{A}$ is statistically close to random), and the last inequality is by hardness of search $\mathsf{LWE}(\lambda, q, \alpha)$.

□

6 Public-Key Encryption from LPN with Constant Noise

Yu, Zhang [YZ16] proposed a PKE scheme based on $(2^{\omega(n^{1/2})}, 2^{-\omega(n^{1/2})})$-hardness of $\mathsf{LPN}(n, \mu)$, where μ is a constant. In this section, we propose a simpler PKE scheme with equivalent security. This construction is very similar to that in Subsection 3.1.

6.1 Single-Bit LPN-PKE Scheme with Weak Correctness

Construction Let $\lambda = H_\infty(\Xi^{[n;k]})/2 = \Theta(\log^2 n)$, where $k = \log n$. Let $\mu \in (0, \frac{1}{2})$ be a constant.

- The message space is $\mathcal{M} = \{0, 1\}$.
- $\mathsf{KeyGen}(1^n)$: Given security parameter 1^n, it samples $\mathbf{A} \leftarrow \{0, 1\}^{\lambda\times n}$, as well as $\mathbf{s} \leftarrow \{0, 1\}^\lambda$, $\mathbf{e} \leftarrow \mathcal{B}_\mu^n$. Then it computes $\mathbf{b} := \mathbf{A}^T\mathbf{s} + \mathbf{e}$, and sets $(pk, sk) := ((\mathbf{A}, \mathbf{b}), \mathbf{s})$.

- $\mathsf{Enc}(pk, \mathfrak{m})$: Given the public key $pk = (\mathbf{A}, \mathbf{b})$ and the message $\mathfrak{m} \in \mathcal{M}$, it samples $\mathbf{r} \leftarrow \Xi^{[n;k]}$, and outputs $\mathfrak{c} := (\mathbf{c}_1, c_2)$ as ciphertext, where $\mathbf{c}_1 = \mathbf{Ar}$ and $c_2 = \mathbf{r}^T\mathbf{b} + \mathfrak{m}$.
- $\mathsf{Dec}(sk, \mathfrak{c})$: Given the secret key $sk = \mathbf{s}$ and the ciphertext $\mathfrak{c}$, parse $\mathfrak{c}$ into $(\mathbf{c}_1, c_2)$, output $d = c_2 - \mathbf{c}_1^T\mathbf{s}$.

Theorem 15 (Weak correctness). *Let $\lambda = \Theta(\log^2 n) = H_\infty(\Xi^{[n;k]})/2$, $\mu \in (0, 1/2)$ be a constant. Π^{LPN} has $(\frac{1}{2} + \frac{1}{\mathsf{poly}(n)})$-correctness.*

Remark 2. To make the scheme strongly correct, we cannot apply the method in subsection 3.2, i.e., generating a public key with ℓ secret keys, and encrypting the message for ℓ times. This is because the correctness of Π^{LPN} is only $\frac{1}{2} + \frac{1}{\mathsf{poly}(n)}$, which needs $\mathsf{poly}(n)$ times of repetition to reach $1 - \mathsf{negl}(n)$. However, we need $\ell \leq O(\lambda) = O(\log^2 n)$ to bound the min-entropy of $\Xi^{[n;k]}$. Therefore, to achieve strong correctness, we should apply a parallel repetition to not only the secret key but also the public key.

Proof. $d = c_2 - \mathbf{c}_1^T\mathbf{s} = \mathbf{r}^T\mathbf{e} + \mathfrak{m}$. Let $e' := \mathbf{r}^T\mathbf{e}$, then $e \sim (\mathcal{B}_\mu)^{\star k}$. By Lemma 8, $\Pr[e' = 1] = \frac{1}{2}(1 - (1 - 2\mu)^k) = \frac{1}{2} - \frac{1}{\mathsf{poly}(n)}$. So $\Pr[d = \mathfrak{m}] = \frac{1}{2} + \frac{1}{\mathsf{poly}(n)}$. □

Theorem 16 (CPA security). *Let $\lambda = \Theta(\log^2 n) = H_\infty(\Xi^{[n;k]})/2$, $\mu \in (0, 1/2)$ be a constant. Assume $\mathsf{LPN}(\lambda, \mu)$ is hard, then Π^{LPN} is IND-CPA secure.*

Remark 3. Similar to Lemma 2, we have the hardness of $\mathsf{LPN}(\lambda, \mu)$ can be implied by $(2^{\omega(n^{1/2})}, 2^{-\omega(n^{1/2})})$-hardness of $\mathsf{LPN}(n, \mu)$.

The proof of this theorem is almost the same as that of Theorem 6, and we postpone it to the full version.

Acknowledgments

We thank Yu Yu for discussions and anonymous reviewers for their valuable comments. L.J. and Y.C. are supported by Shanghai Qi Zhi Institute Innovation Program SQZ202405 and Tsinghua University start-up funding.

References

ABB+23. Divesh Aggarwal, Huck Bennett, Zvika Brakerski, Alexander Golovnev, Rajendra Kumar, Zeyong Li, Spencer Peters, Noah Stephens-Davidowitz, and Vinod Vaikuntanathan. Lattice problems beyond polynomial time. In *STOC*, pages 1516–1526. ACM, 2023. 2

Bab86. László Babai. On lovász' lattice reduction and the nearest lattice point problem. *Comb.*, 6(1):1–13, 1986. 3, 5, 10

BKP23. James Bartusek, Dakshita Khurana, and Alexander Poremba. Publicly-verifiable deletion via target-collapsing functions. In *CRYPTO (5)*, volume 14085 of *Lecture Notes in Computer Science*, pages 99–128. Springer, 2023. 6

BV11. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, pages 97–106. IEEE Computer Society, 2011. 2

CHL+25. Yilei Chen, Zihan Hu, Qipeng Liu, Han Luo, and Yaxin Tu. LWE with quantum amplitudes: Algorithm, hardness, and oblivious sampling. In *CRYPTO*, 2025. 22

GKM+00. Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *FOCS*, pages 325–335. IEEE Computer Society, 2000. 6

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. ACM, 2008. 2

GSW13. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO (1)*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2013. 2

GVW13. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *STOC*, pages 545–554. ACM, 2013. 2

ILL89. Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *STOC*, pages 12–24. ACM, 1989. 25

Jus72. Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Trans. Inf. Theory*, 18(5):652–656, 1972. 9

LM00. Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of statistics*, pages 1302–1338, 2000. 25

MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, 2012. 2, 3, 5, 12, 13, 14

PRS17. Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In *STOC*, pages 461–473. ACM, 2017. 2

PVW08. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, 2008. 9

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93. ACM, 2005. 2, 4, 8, 9, 22

Vai20. Vinod Vaikuntanathan. CS 294-168 Lattices, Learning with Errors and Post-Quantum Cryptography: Lecture 1, 2020. https://people.csail.mit.edu/vinodv/CS294/lecture1.pdf. 2

YZ16. Yu Yu and Jiang Zhang. Cryptography with auxiliary input and trapdoor from constant-noise LPN. In *CRYPTO (1)*, volume 9814 of *Lecture Notes in Computer Science*, pages 214–243. Springer, 2016. 3, 19, 27

A Preliminary (Extended)

A.1 Linear Algebra and Lattices

For any ordered set $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\} \in \mathbb{R}^n$, denote its Gram-Schmidt orthogonalization as $\tilde{\mathbf{V}} = \{\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_k\}$, where $\tilde{\mathbf{v}}_i$ is the component of $\mathbf{v}_i$ orthogonal to

$\mathsf{span}(\mathbf{v}_1, \ldots, \mathbf{v}_{i-1})$ for all $i = 1, \ldots, k$. For any basis $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ of $\mathbb{R}^n$, its dual basis is defined as $\mathbf{V}^* = \mathbf{V}^{-T}$. If we orthogonalize $\mathbf{V}$ and $\mathbf{V}^*$ in forward and reverse order respectively, then we have $\tilde{\mathbf{v}}_i^* = \tilde{\mathbf{v}}_i / \|\tilde{\mathbf{v}}_i\|^2$. Particularly, $\|\tilde{\mathbf{v}}_i^*\| = 1/\|\tilde{\mathbf{v}}_i\|$.

An n-dimensional lattice L of rank $k \leq n$ is a discrete additive subgroup of $\mathbb{R}^n$. Given k linearly independent basis vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k \in \mathbb{R}^n\}$, the lattice generated by $\mathbf{B}$ is $L(\mathbf{B}) = L(\mathbf{b}_1, \ldots, \mathbf{b}_k) = \{\sum_{i=1}^k x_i \cdot \mathbf{b}_i : x_i \in \mathbb{Z}\}$. By default, we work with full-rank lattices unless explicitly mentioned. The dual of lattice $L \subseteq \mathbb{R}^n$ is defined as $L^* := \{\mathbf{y} \in \mathbb{R}^n : \langle \mathbf{y}, \mathbf{x} \rangle \in \mathbb{Z} \text{ for all } \mathbf{x} \in L\}$.

If $\mathbf{B}$ is a basis of a full-rank lattice L, then $\mathbf{B}^* = \mathbf{B}^{-T}$ is a basis of L^*. For an arbitrary matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, define the following q-ary lattices

$$\begin{aligned} \Lambda^{\perp}(\mathbf{A}) &:= \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A}\mathbf{z} = 0 \bmod q\} \\ \Lambda(\mathbf{A}^T) &:= \{\mathbf{z} \in \mathbb{Z}^m : \exists \mathbf{s} \in \mathbb{Z}_q^n \text{ s.t. } \mathbf{z} = \mathbf{A}^T\mathbf{s} \bmod q\}. \end{aligned} \tag{2}$$

It is easy to check that $q \cdot \Lambda^{\perp}(\mathbf{A})^* = \Lambda(\mathbf{A}^T)$, that is, $\Lambda^{\perp}(\mathbf{A})^*$ and $\Lambda(\mathbf{A}^T)$ are dual lattices up to a scaling factor of q.

Lemma 6 ([CHL+25, Lemma 16]). *Let $q \geq 2, m \geq 2n \log q$, then for all but at most $q^{-0.16n}$ fraction of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we have $\lambda_1^{\infty}(\Lambda(\mathbf{A}^T)) \geq q/4$, where λ_1^{∞} is the first successive minimum in distance measured in the ℓ_{∞}-norm.*

A.2 Gaussians

In this paper, we focus on 1-dimensional Gaussians. For any real $r > 0$, define the Gaussian function on $\mathbb{R}$ with width parameter r as: $\forall x \in \mathbb{R},\ \rho_r(x) := e^{-\pi x^2/r^2}$. And we define the continuous Gaussian distribution D_r as: $\forall x \in \mathbb{R},\ D_r(x) := \rho_r(x)/r$. For any real $\alpha > 0$ and integer $q > 0$, we use the following version of discrete Gaussian distribution: $\forall i \in \mathbb{Z},\ \Psi_{\alpha,q}(i) := \int_{x=i-\frac{1}{2}}^{i+\frac{1}{2}} D_{\alpha q}(x)dx$. That is, $\Psi_{\alpha,q} \sim \lfloor D_{\alpha q} \rceil$.

A.3 Learning with Errors

In this paper, we focus on the following variant of LWE, which is put forward in [Reg05].

Definition 1 (Learning with errors). *Let $q = q(n)$ be a prime modulus. $m = m(n), \alpha = \alpha(n), t = t(n), \epsilon = \epsilon(n)$. We say the search $\mathsf{LWE}(n, q, \alpha)$ is (t, ϵ)-hard if for every inverter $\mathcal{A}$ of running time t,*

$$\Pr_{\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e} \leftarrow \Psi_{\alpha,q}^m} [\mathcal{A}(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e}) = \mathbf{s}] < \epsilon.$$

We say the decision $\mathsf{LWE}(n, q, \alpha)$ is (t, ϵ)-hard if for every distinguisher $\mathcal{D}$ of running time t,

$$\left| \Pr_{\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e} \leftarrow \Psi_{\alpha,q}^m} [\mathcal{D}(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e}) = 1] - \Pr_{\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}} [\mathcal{D}(\mathbf{A}, \mathcal{U}_q^m) = 1] \right| < \epsilon.$$

When $t = n^{\omega(1)}$ and $\epsilon = n^{-\omega(1)}$, we simply say the search (decision) LWE problem is hard.

A.4 Learning Parity with Noise

The learning parity with noise (LPN) problem, which can be seen as an analog of LWE, is defined as follows.

Definition 2 (Learning parity with noise). *Let $m = m(n), 0 < \mu < 1/2, t = t(n), \epsilon = \epsilon(n)$. We say the search $\mathsf{LPN}(n, \mu)$ is (t, ϵ)-hard if for every inverter $\mathcal{A}$ of running time t,*

$$\Pr_{\mathbf{A}\leftarrow\{0,1\}^{n\times m},\mathbf{s}\leftarrow\{0,1\}^n,\mathbf{e}\leftarrow\mathcal{B}_\mu^m}[\mathcal{A}(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e}) = \mathbf{s}] < \epsilon.$$

We say the decision $\mathsf{LPN}(n, \mu)$ is (t, ϵ)-hard if for every distinguisher $\mathcal{D}$ of running time t,

$$\left|\Pr_{\mathbf{A}\leftarrow\{0,1\}^{n\times m},\mathbf{s}\leftarrow\{0,1\}^n,\mathbf{e}\leftarrow\mathcal{B}_\mu^m}[\mathcal{D}(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e}) = 1] - \Pr_{\mathbf{A}\leftarrow\{0,1\}^{n\times m}}[\mathcal{D}(\mathbf{A},\mathcal{U}^m) = 1]\right| < \epsilon.$$

When $t = n^{\omega(1)}$ and $\epsilon = n^{-\omega(1)}$, we simply say the search (decision) LPN problem is hard.

A.5 Public-Key Encryption Schemes

Definition 3 (Public-key encryption schemes). *A public-key encryption scheme is a tuple of PPT algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space $\mathcal{M}$ such that*

- ***Key generation:*** $\mathsf{KeyGen}(1^n)$ *takes as input a security parameter 1^n, and outputs a pair of public and private keys (pk, sk).*
- ***Encryption:*** $\mathsf{Enc}(pk, \mathfrak{m})$ *takes a public key pk and a message $\mathfrak{m} \in \mathcal{M}$ as input, and outputs a ciphertext $\mathfrak{c}$.*
- ***Decryption:*** $\mathsf{Dec}(sk, \mathfrak{c})$ *takes as input a secret key sk and ciphertext $\mathfrak{c}$, and deterministically outputs a message $\mathfrak{m}$.*

Definition 4 (δ-correctness). *For $0 < \delta \leq 1$, we say that a public-key encryption scheme is δ-correct if for every $\mathfrak{m} \in \mathcal{M}$, it holds that*

$$\Pr_{(pk,sk)\leftarrow\mathsf{KeyGen}(1^n)}[\mathsf{Dec}(sk, \mathsf{Enc}(pk, \mathfrak{m})) = \mathfrak{m}] \geq \delta.$$

When $\delta > 1 - \mathsf{neg}(n)$, we say the scheme is (strongly) correct. Otherwise, we say it is weakly correct if for any $\mathfrak{m}' \in \mathcal{M}$ such that $\mathfrak{m}' \neq \mathfrak{m}$, it holds that

$$\Pr_{(pk,sk)\leftarrow\mathsf{KeyGen}(1^n)}[\mathsf{Dec}(sk, \mathsf{Enc}(pk, \mathfrak{m})) = \mathfrak{m}] - \Pr_{(pk,sk)\leftarrow\mathsf{KeyGen}(1^n)}[\mathsf{Dec}(sk, \mathsf{Enc}(pk, \mathfrak{m})) = \mathfrak{m}'] \geq \frac{1}{\mathsf{poly}(n)}.$$

Definition 5 (IND-CPA security). *A public-key encryption scheme Π is IND-CPA secure if for any PPT adversaries $\mathcal{A}$,*

$$\Pr[\mathsf{Exp}^{\mathsf{CPA}}_{\mathcal{A},\Pi}(1^n) = 1] = \frac{1}{2} + \mathsf{negl}(n),$$

where $\mathsf{Exp}^{\mathsf{CPA}}_{\mathcal{A},\Pi}(1^n)$ is the IND-CPA indistinguishability experiment defined as below:

- $\mathsf{KeyGen}(1^n)$ *is run to obtain* (pk, sk).
- *The adversary $\mathcal{A}$ takes 1^n and pk as input, and then outputs a pair of messages $\mathfrak{m}_0, \mathfrak{m}_1$ with the same length.*
- *A random bit $\beta \leftarrow \{0,1\}$ is chosen, and then a challenge ciphertext $\mathfrak{c} \leftarrow \mathsf{Enc}_{pk}(\mathfrak{m}_\beta)$ is computed and given to $\mathcal{A}$.*
- *$\mathcal{A}$ guesses a bit β'.*
- *The output of the experiment is defined to be 1 if $\beta' = \beta$ (indicating $\mathcal{A}$ succeeds), and 0 otherwise (indicating $\mathcal{A}$ fails).*

A.6 Injective Trapdoor Functions

Definition 6 (Trapdoor functions). *An (injective) trapdoor function family is a tuple of PPT algorithms* $(\mathsf{Sample}, \mathsf{Eval}, \mathsf{Invert})$ *with domain $\mathcal{D}$ and range $\mathcal{R}$ such that*

- $\mathsf{Sample}(1^n)$ *outputs* (ek, ik)*, where ek is an evaluation key and ik is an inversion key (trapdoor).*
- $\mathsf{Eval}(ek, x)$ *takes as input a string $x \in \mathcal{D}$ and an evaluation key ek, and outputs a string $y \in \mathcal{R}$.*
- $\mathsf{Invert}(ik, y)$ *takes a string $y \in \mathcal{R}$ and an inversion key ik as inputs, outputting either a string $x \in \mathcal{D}$ or a special symbol $\perp$ to denote failure.*

Definition 7 (Injectivity). *For any injective trapdoor function (iTDF) family $\mathcal{T} : \mathcal{D} \to \mathcal{R}$ consisting of* $(\mathsf{Sample}, \mathsf{Eval}, \mathsf{Invert})$*, it holds that*

$$\Pr_{(ek,ik)\leftarrow\mathsf{Sample}(1^n)}[\exists x_1, x_2 \in \mathcal{D} \text{ s.t. } (x_1 \neq x_2 \wedge \mathsf{Eval}(ek, x_1) = \mathsf{Eval}(ek, x_2))] = \mathsf{negl}(n).$$

Definition 8 (Φ-invertibility). *For some distribution Φ on $\mathcal{D}$, an (injective) trapdoor function is easy to invert only with the trapdoor. More concretely, an iTDF family $\mathcal{T} : \mathcal{D} \to \mathcal{R}$ consisting of* $(\mathsf{Sample}, \mathsf{Eval}, \mathsf{Invert})$ *should be*

- *easy to invert with a trapdoor, i.e.,*

$$\Pr_{(ek,ik)\leftarrow\mathsf{Sample}(1^n), x\leftarrow\Phi}[\mathsf{Invert}(ik, \mathsf{Eval}(ek, x)) = x] = 1 - \mathsf{negl}(n).$$

- *hard to invert without a trapdoor, i.e., for any PPT adversary $\mathcal{A}$*

$$\Pr_{(ek,ik)\leftarrow\mathsf{Sample}(1^n), x\leftarrow\Phi}[\mathcal{A}(ek, \mathsf{Eval}(ek, x)) = x] = \mathsf{negl}(n).$$

A.7 Leftover Hash Lemma

We first present the definition of 2-universal hash function family as follows.

Definition 9. *A family of hash functions* $\mathcal{H} := \{h_k : \mathcal{X} \to \{0,1\}^l, k \in \mathcal{S}\}$ *is 2-universal if for any* $x_1, x_2 \in \mathcal{X}$ *such that* $x_1 \neq x_2$*, it holds that* $\Pr_{k \leftarrow \mathcal{S}}[h_k(x_1) = h_k(x_2)] \leq 2^{-l}$.

We present the leftover hash lemma [ILL89] as follows.

Lemma 7 (Leftover hash lemma). *Let* $X \in \mathcal{X}$ *be a random variable such that* $H_\infty(X) \geq k$*. Let* $h : \{0,1\}^s \times \mathcal{X} \to \{0,1\}^l$ *be any 2-universal hash function. If* $V \leftarrow \{0,1\}^s$*, then it holds that* $\mathsf{SD}((V, h(V,X)), (V, \mathcal{U}^l)) \leq 2^{\frac{l-k}{2}}$.

A.8 Probabilistic Bounds

In this subsection, we present some mathematical bounds.

Lemma 8 (Piling-up lemma). *For* $0 < \mu < 1/2$*, let* $X_1, \ldots, X_\ell$ *be independent random variables sampled from* $\mathcal{B}_\mu$*. Then,* $\bigoplus_{i=1}^{\ell} X_i \sim \mathcal{B}_\sigma$*, where* $\sigma = \frac{1}{2}(1 - (1-2\mu)^\ell)$.

Lemma 9 (Laurent-Massart bounds [LM00, Lemma 1]). *The chi-squared distribution with* k *degrees of freedom, denoted as* $\chi^2(k)$*, is the distribution of a sum of* k *independent standard normal variables. Let* $X \sim \chi^2(k)$*. Then, for any* $x > 0$*,* $\Pr[X - k > 2\sqrt{kx} + 2x] \leq e^{-x}$.

Lemma 10. *When* $k = O(m^{<1})$*,* $H_\infty(\Xi^{[m;k]}) = \Theta(k \log m)$.

Proof. By the definition of min-entropy, we get

$$H_\infty(\Xi^{[m;k]}) = \log \binom{m}{k} = \sum_{i=1}^{k} \log \frac{i+m-k}{i} \in (k \log(m/k), k \log(m-k+1)).$$

Therefore, $H_\infty(\Xi^{[m;k]}) = \Theta(k \log m)$. □

Lemma 11. *For any* $a > 0$*,* $\Pr_{x \sim D_\alpha}[|x| \geq a] < \exp\left(-\frac{a^2\pi}{2\alpha^2}\right)$.

Proof. Consider

$$\Pr[x \geq a] = \int_{u=a}^{\infty} \frac{1}{\alpha} \exp\left(-\pi \left(\frac{u}{\alpha}\right)^2\right) \mathrm{d}u = \frac{1}{\alpha} \sqrt{\int_{u=a}^{\infty} \int_{v=a}^{\infty} \exp\left(-\pi \frac{u^2+v^2}{\alpha^2}\right) \mathrm{d}u \mathrm{d}v}.$$

Make a polar coordinate substitution, that is, let $u := \alpha\rho\cos\theta, v := \alpha\rho\sin\theta$, and we obtain

$$\Pr[x \geq a] < \frac{1}{\alpha} \sqrt{\int_{\rho=a/\alpha}^{\infty} \int_{\theta=0}^{\pi/2} \exp\left(-\pi\rho^2\right) \alpha^2 \rho \mathrm{d}\rho \mathrm{d}\theta} = \sqrt{\frac{\pi}{2} \int_{\rho=a/\alpha}^{\infty} \exp\left(-\pi\rho^2\right) \rho \mathrm{d}\rho}.$$

Further let $t := \pi\rho^2$, and we get

$$\Pr[x \geq a] < \sqrt{\frac{\pi}{2}\int_{t=\pi a^2/\alpha^2}^{\infty} \frac{1}{2\pi}\exp(-t)\mathrm{d}t} = \sqrt{\frac{1}{4}\exp\left(-\frac{\pi a^2}{\alpha^2}\right)} = \frac{1}{2}\exp\left(-\frac{\pi a^2}{2\alpha^2}\right).$$

which implies $\Pr[|x| \geq a] < \exp\left(-\frac{\pi a^2}{2\alpha^2}\right)$. □

Lemma 12. *For $\alpha > 0$, let $e_1, e_2, \ldots, e_n \sim D_\alpha$, the following two inequalities hold with overwhelming probability.*

(i) $|\sum_{i=1}^n e_i| < 0.1\alpha\sqrt{n\log n}$.
(ii) $\sum_{i=1}^n e_i^2 < 0.16\alpha^2 n$.

Proof.

(i) Since $\sum_{i=1}^n e_i \sim D_{\sqrt{n}\alpha}$, the conclusion can be directly deduced by applying Lemma 11.
(ii) By Lemma 9, we have $\Pr[\sum_{i=1}^n (\frac{\sqrt{2\pi}}{\alpha}e_i)^2 < 1.001n] = \mathsf{negl}(n)$, which implies $\sum_{i=1}^n e_i^2 < 0.16\alpha^2 n$ with overwhelming probability.

□

Lemma 13. *For any positive integers k, q, real constants α, let $\mathbf{e} \leftarrow \Psi_{\alpha,q}^k$, $\mathbf{t} \in \mathbb{R}^k$. If $\|\mathbf{t}\|_1 = o(q)$, then for any two constants γ, γ' such that $0 < \gamma < \gamma' < 1$, it holds that*

$$\Pr\left[|\langle \mathbf{e}, \mathbf{t}\rangle| < \gamma' q\right] > 1 - \exp\left(-\frac{\pi\gamma^2}{2\|\mathbf{t}\|_2^2\alpha^2}\right).$$

Proof. By definition, for every $1 \leq i \leq k$, $e(i)$ is obtained by sampling some $e_i' \sim D_\alpha$, and calculate $e(i) = \lfloor qe_i' \rceil$. Therefore, we may write $\langle \mathbf{e}, \mathbf{t}\rangle = \sum_{i=1}^k t(i)\lfloor qe_i' \rceil$. Since the bias introduced by the $\lfloor\cdot\rceil$ function does not exceed $1/2$, we have $|\langle \mathbf{e}, \mathbf{t}\rangle - \sum_{i=1}^k t(i)qe_i'| < \frac{\|\mathbf{t}\|_1}{2}$. Thus, it suffices to prove:

$$\Pr\left[\left|\sum_{i=1}^k t(i)qe_i'\right| < \gamma' q - \frac{\|\mathbf{t}\|_1}{2}\right] > 1 - \exp\left(-\frac{\pi\gamma^2}{2\|\mathbf{t}\|_2^2\alpha^2}\right).$$

Since $\|\mathbf{t}\|_1 = o(q)$, and γ, γ' are two constants such that $0 < \gamma < \gamma' < 1$, we have $\gamma' q - \frac{\|\mathbf{t}\|_1}{2} > \gamma q$. Therefore, we only need to prove

$$\Pr\left[\left|\sum_{i=1}^k t(i)e_i'\right| < \gamma\right] > 1 - \exp\left(-\frac{\pi\gamma^2}{2\|\mathbf{t}\|_2^2\alpha^2}\right).$$

Due to the properties of the normal distribution, $\sum_{i=1}^k t(i)e_i' \sim D_{\|\mathbf{t}\|_2\alpha}$. According to Lemma 11, the above statement holds. □

Corollary 2. *Let γ, γ' be two constant such that $0 < \gamma < \gamma' < 1$. If $k = o(q)$, then*

$$\Pr_{e \sim (\Psi_{\alpha,q})^{\star k}}[|e| < \gamma' q] > 1 - \exp\left(-\frac{\pi\gamma^2}{2k\alpha^2}\right).$$

Proof. Let $\mathbf{t} = (1, \ldots, 1) \in \mathbb{R}^k$, $\mathbf{x} \leftarrow \Psi_{\alpha,q}^k$. Then e and $\langle \mathbf{t}, \mathbf{x} \rangle$ have the same distribution. Using Lemma 13, we get the conclusion. □

A.9 Truncated Distributions

We first introduce the truncated Bernoulli distribution, which is put forward in [YZ16].

Definition 10. *For any $\mu \in (0,1)$, a sample $\mathbf{x} \sim \tilde{\mathcal{B}}_\mu^n$ is generated in the following procedure.*

(i) $\mathbf{x} \leftarrow \mathcal{B}_\mu^n$.
(ii) *If* $\mathsf{Ham}(\mathbf{x}) \notin [\frac{1}{2}\mu n, \frac{3}{2}\mu n]$, *discard* $\mathbf{x}$ *and go back to (i).*

When $\mu = \omega(1/n)$, we have $\tilde{\mathcal{B}}_\mu^n$ can be efficiently sampled, and $\mathsf{SD}(\tilde{\mathcal{B}}_\mu^n, \mathcal{B}_\mu^n) = \exp(-\Theta(\mu n))$, which can be proved using Chernoff Bound. What's more, for any positive integer $k = O(n^{<1})$, $\tilde{\mathcal{B}}_{k/n}^n$ can be viewed as a combination of $\Xi^{[n;k/2]}, \ldots, \Xi^{[n;3k/2]}$, so we have $H_\infty(\tilde{\mathcal{B}}_{k/n}^n) = \Theta(k \log n)$ by Lemma 10.

Then we define the truncated discrete gaussian distribution as follows.

Definition 11. *For any $\alpha > 0$, a sample $e \leftarrow \tilde{\Psi}_{\alpha,q}$ is generated in the following procedure.*

(i) $e \leftarrow \Psi_{\alpha,q}$.
(ii) *If* $e \notin [-q\alpha\sqrt{\log n}, q\alpha\sqrt{\log n}]$, *discard* e *and go back to (i).*

By Lemma 11, we know $\tilde{\Psi}_{\alpha q}$ can be efficiently sampled, and is statistically indistinguishable from $\Psi_{\alpha q}$.

B The IND-CPA Security of Π_λ^{LWE}

Proof of Theorem 6. Consider any adversary $\mathcal{A}$. We aim to demonstrate that $\mathcal{A}$'s probability of winning the IND-CPA security experiment is $\frac{1}{2} + \mathsf{negl}(n)$. We prove this by game hopping. To be precise, we define a sequence of games, where

(i) The first game is the same as $\mathsf{Exp}^{\mathsf{CPA}}_{\mathcal{A},\Pi_\lambda^{LWE}}(1^n)$
(ii) In the last game, $\mathcal{A}$ gains no advantage in correctly guessing the challenge bit.
(iii) Any two consecutive games are computationally indistinguishable.

Then the security of Π_λ^{LWE} will be established.

Game 0. In this game, the challenger $\mathcal{C}$ simulates exactly $\mathsf{Exp}^{\mathsf{CPA}}_{\mathcal{A},\Pi^{LWE}_{\lambda}}(1^n)$ for the adversary $\mathcal{A}$. Because this is a single-bit PKE, we can set $\mathfrak{m}_0 = 0$ and $\mathfrak{m}_1 = 1$ without loss of generality (thus there is no need for $\mathcal{A}$ to choose the message).

- The challenger $\mathcal{C}$ samples $\mathbf{A} \leftarrow \mathbb{Z}_q^{\lambda \times n}$, as well as $\mathbf{s} \leftarrow \mathbb{Z}_q^{\lambda}$, $\mathbf{e} \leftarrow \Psi_{\alpha,q}^n$, computes $\mathbf{b} := (\mathbf{A}^T\mathbf{s} + \mathbf{e}) \bmod q$, and sends $pk := (\mathbf{A}, \mathbf{b})$ to $\mathcal{A}$.
- $\mathcal{C}$ continues to choose a random bit $\beta \leftarrow \{0,1\}$, and compute $\mathbf{c}_1 = \mathbf{A}\mathbf{r}$, $c_2 = (\mathbf{r}^T\mathbf{b} + \lfloor q/2 \rceil \cdot \beta) \bmod q$. Then it gives the challenge ciphertext $\mathfrak{c} := (\mathbf{c}_1, c_2)$ to $\mathcal{A}$.
- After $\mathcal{A}$ guesses a bit β', $\mathcal{C}$ outputs 1 if $\beta' = \beta$, or outputs 0 otherwise.

Game 1. This game is identical to Game 0 except that the second element of the public key is sampled uniformly from $\mathbb{Z}_q^n$. In other words, the challenger samples $\mathbf{A} \leftarrow \mathbb{Z}_q^{\lambda \times n}$, $\mathbf{b}' \leftarrow \mathbb{Z}_q^n$, and sends $pk = (\mathbf{A}, \mathbf{b}')$ and ciphertext $\mathfrak{c} := (\mathbf{A}\mathbf{r}, (\mathbf{r}^T\mathbf{b}' + \lfloor q/2 \rceil \cdot \beta) \bmod q)$ to $\mathcal{A}$.

By hardness assumption of $\mathsf{LWE}(\lambda, q, \alpha)$, Game 1 is indistinguishable from Game 0.

Game 2. This game is identical to Game 1 except $\mathfrak{c}$ is chosen uniformly randomly. In other words, the challenger samples $\mathbf{A} \leftarrow \mathbb{Z}_q^{\lambda \times n}$, $\mathbf{b}' \leftarrow \mathbb{Z}_q^n$, and sends $pk = (\mathbf{A}, \mathbf{b}')$ and ciphertext $\mathfrak{c} \sim \mathcal{U}_q^n \times \mathcal{U}_q$.

By Lemma 7, $\mathsf{SD}((\mathbf{A}, \mathbf{b}, \mathbf{A}\mathbf{r}, \mathbf{r}^T\mathbf{b}), (\mathbf{A}, \mathbf{b}, \mathcal{U}^{\lambda}, \mathcal{U})) \leq 2^{((\lambda+1)\log q - H_\infty(\mathbf{r}))/2} = \mathsf{negl}(n)$. So Game 2 is indistinguishable from Game 1. What's more, in Game 2, $\mathcal{A}$ receives nothing about β. Thus we have $\Pr[\beta' = \beta] = \frac{1}{2}$. □

C Proof of Lemma 3

Proof. By following the Gram-Schmidt procedure directly, we work out the value of each $\tilde{\mathbf{t}}_i$ as follows. When $1 \leq i < \kappa$,

$$\tilde{\mathbf{t}}_i = \left(\frac{2^i}{\sum_{j=0}^{i-1} 4^j}, 2 \cdot \frac{2^i}{\sum_{j=0}^{i-1} 4^j}, \ldots, 2^{i-1} \cdot \frac{2^i}{\sum_{j=0}^{i-1} 4^j}, -1, 0, \ldots, 0\right)^T.$$

So we have

$$\|\tilde{\mathbf{t}}_i\|_1 = 1 + \sum_{j=0}^{i-1} 2^j \cdot \frac{2^i}{\sum_{j=0}^{i-1} 4^j} = 1 + (2^i - 1) \cdot \frac{2^i}{(4^i - 1)/3} = 1 + \frac{3 \cdot 2^i}{2^i + 1} = \frac{2^{i+2} + 1}{2^i + 1} \in [3, 4),$$

$$\|\tilde{\mathbf{t}}_i\|_2^2 = 1 + \sum_{j=0}^{i-1} 4^{j-1} \cdot \left(\frac{2^i}{\sum_{j=0}^{i-1} 4^j}\right)^2 = 1 + \frac{4^i}{\sum_{j=0}^{i-1} 4^j} = \frac{4^{i+1} - 1}{4^i - 1} \in (4, 5].$$

When $i = \kappa$,

$$\tilde{\mathbf{t}}_\kappa = \left(\frac{q}{\sum_{j=0}^{\kappa-1} 4^j}, 2 \cdot \frac{q}{\sum_{j=0}^{\kappa-1} 4^j}, \ldots, 2^{\kappa-1} \cdot \frac{q}{\sum_{j=0}^{\kappa-1} 4^j}\right)^T.$$

By $q \in [2^{\kappa-1}+1, 2^\kappa - 1]$, we get

$$\|\tilde{\mathbf{t}}_\kappa\|_1 = \sum_{j=0}^{\kappa-1} 2^j \cdot \frac{q}{\sum_{j=0}^{\kappa-1} 4^j} = (2^\kappa - 1) \cdot \frac{q}{(4^\kappa - 1)/3} = \frac{3q}{2^\kappa + 1} \in (\frac{3}{2}, 3),$$

$$\|\tilde{\mathbf{t}}_\kappa\|_2^2 = \sum_{j=0}^{i-1} 4^j \cdot (\frac{q}{\sum_{j=0}^{\kappa-1} 4^j})^2 = \frac{q^2}{\sum_{j=0}^{\kappa-1} 4^j} = \frac{3q^2}{4^\kappa - 1} \in (\frac{3}{4}, 3).$$

□

D Proof of Lemma 5

Proof. Consider $\langle \mathbf{R}^T \bar{\mathbf{e}}, \tilde{\mathbf{t}}_j \rangle = \sum_{x=1}^{\kappa} \tilde{t}_j(x) \langle \mathbf{r}_x, \bar{\mathbf{e}} \rangle$. Define $\xi_x \sim \langle \mathcal{B}_{k/\bar{n}}^{\bar{n}}, \mathbf{e}' \rangle$ for all $1 \le x \le \kappa$.

$$|\mathbb{E}[\xi_x]| = \frac{k}{\bar{n}} |\sum_{i=1}^{\bar{n}} e'(i)| < 0.01 \sqrt{\frac{k \log \bar{n}}{\bar{n}}} < 0.01.$$

$$\mathbb{D}[\xi_x] = \frac{k}{\bar{n}} \cdot (1 - \frac{k}{\bar{n}}) \cdot \sum_{i=1}^{n} e'(i)^2 < \frac{0.16}{100}.$$

Let $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_\kappa)^T$, and define $\zeta_j = \langle \boldsymbol{\xi}, \tilde{\mathbf{t}}_j \rangle = \sum_{x=1}^{\kappa} \tilde{t}_j(x) \xi_x$ for all $1 \le j \le \kappa$. By Lemma 3,

$$|\mathbb{E}[\zeta_j]| = |\sum_{x=1}^{\kappa} \tilde{t}_j(x) \mathbb{E}[\xi_x]| < 0.01 \|\tilde{\mathbf{t}}_j\|_1 < 0.04,$$

$$\mathbb{D}[\zeta_j] = \sum_{j=1}^{\kappa} \tilde{t}_j(x)^2 \mathbb{D}[\xi_x] < \frac{0.16}{100} \|\tilde{\mathbf{t}}_j\|_2^2 < 0.008.$$

By Chebyshev's inequality,

$$\Pr[|\zeta_j - \mathbb{E}[\zeta_j]| \ge 0.2] \le \frac{\mathbb{D}[\xi]}{0.2^2} < 0.2,$$

so

$$\Pr[|\zeta_j| \ge 0.24] < 0.2.$$

Because for all $1 \le x \le \kappa$, $\mathbf{r}_x \sim \tilde{\mathcal{B}}_{k/\bar{n}}^{\bar{n}}$, by Chernoff bound we have

$$\mathsf{SD}(\mathbf{r}_x, \mathcal{B}_{k/\bar{n}}^{\bar{n}}) = \exp(-\Theta(k)).$$

Because the statistical distance will decrease after applying a function, we have

$$\mathsf{SD}(\langle \mathbf{r}_x, \mathbf{e}' \rangle, \xi_x) \le \exp(-\Theta(k)),$$

and then it holds for all $1 \le j \le \kappa$ that

$$\mathsf{SD}(\langle \mathbf{R}^T\mathbf{e}', \tilde{\mathbf{t}}_j\rangle, \zeta_j) \le \sum_{x=1}^{\kappa} \mathsf{SD}(\langle \mathbf{r}_x, \mathbf{e}'\rangle, \xi_x) \le \kappa \exp(-\Theta(k)) = o(1).$$

Because the rounding error of $\lfloor\cdot\rceil$ is at most $\frac{1}{2}$, and $\mathsf{Ham}(\mathbf{r}_x) \le \frac{3}{2}k$, we have

$$|\langle \mathbf{R}^T \cdot q\mathbf{e}', \tilde{\mathbf{t}}_j\rangle - \langle \mathbf{R}^T\bar{\mathbf{e}}, \tilde{\mathbf{t}}_j\rangle| = |\sum_{x=1}^{\kappa} \tilde{t}_j(x)\langle \mathbf{r}_x, q\mathbf{e}' - \lfloor q\mathbf{e}'\rceil\rangle| \le \sum_{x=1}^{\kappa} \tilde{t}_j(x) \cdot \frac{3k}{4} = \frac{3k}{4}\|\tilde{\mathbf{t}}_j\|_1 = o(q).$$

Finally,

$$\begin{aligned}\Pr[|\langle \mathbf{R}^T\bar{\mathbf{e}}, \tilde{\mathbf{t}}_j\rangle| < \frac{q}{4}] &> \Pr[|\langle \mathbf{R}^T \cdot q\mathbf{e}', \tilde{\mathbf{t}}_j\rangle| + o(q) < \frac{q}{4}] \\ &> \Pr[|\langle \mathbf{R}^T\mathbf{e}', \tilde{\mathbf{t}}_j\rangle| < 0.24] \\ &> \Pr[|\zeta_j| < 0.24] \cdot (1 - o(1)) \\ &> 0.79.\end{aligned}$$

□

E Proof of Theorem 12

Proof. Parse $\mathbf{e} = (\bar{\mathbf{e}}^T \| \underline{\mathbf{e}}_1^T \| \dots \| \underline{\mathbf{e}}_\ell^T)^T \in \mathbb{Z}^{\bar{n}} \times (\mathbb{Z}^w)^{\otimes \ell}$. By definition, there exists some $\mathbf{e}' \leftarrow D_\alpha^{\bar{n}}$ such that $\bar{\mathbf{e}} = \lfloor q\mathbf{e}'\rceil$. By Lemma 12, we have

(i) $|\sum_{i=1}^{\bar{n}} e'(i)| < 0.01\sqrt{\frac{\bar{n}\log\bar{n}}{k}}$,
(ii) $\sum_{i=1}^{\bar{n}} e'(i)^2 < \frac{0.16\bar{n}}{100k}$,

For every $1 \le i \le \ell$, parse $\mathbf{R}_i = (\mathbf{R}_{i,1} \| \dots \| \mathbf{R}_{i,\lambda}) \in (\mathbb{Z}_q^{\bar{n}\times\kappa})^{\otimes\lambda}$ and $\underline{\mathbf{e}}_i = (\underline{\mathbf{e}}_{i,1}^T \| \dots \| \underline{\mathbf{e}}_{i,\lambda}^T)^T \in (\mathbb{Z}^\kappa)^{\otimes\lambda}$. By Lemma 5, we have

$$\forall 1 \le h \le \lambda, \forall 1 \le j \le \kappa,\ \Pr[\langle \mathbf{R}_{i,h}^T\bar{\mathbf{e}}, \tilde{\mathbf{t}}_j\rangle < q/4] > 0.79$$

holds independently for every $1 \le i \le \ell$.

Besides, by Lemma 13, we have for every i, j, h that

$$\Pr[|\langle \underline{\mathbf{e}}_{i,h}, \tilde{\mathbf{t}}_j\rangle| < q/4] > 1 - \exp(-\frac{\pi \cdot (1/5)^2}{2\|\tilde{\mathbf{t}}_j\|_2^2 \cdot (1/10\sqrt{k})^2}) = 1 - \mathsf{negl}(n).$$

Let $\hat{\mathbf{e}}_i := (\hat{\mathbf{e}}_{i,1}^T \| \dots \| \hat{\mathbf{e}}_{i,\lambda})^T = \mathbf{R}_i^T\bar{\mathbf{e}} + \underline{\mathbf{e}}_i \in (\mathbb{Z}^\kappa)^{\otimes\lambda}$, then we have $\hat{\mathbf{e}}_{i,h} = \mathbf{R}_{i,h}^T\bar{\mathbf{e}} + \underline{\mathbf{e}}_{i,h}$ for every $1 \le h \le \lambda$. So

$$\Pr[|\langle \hat{\mathbf{e}}_{i,h}, \tilde{\mathbf{t}}_j\rangle| < q/2] > 0.79 \cdot (1 - \mathsf{negl}(n)) > 2/3$$

holds independently for every i, j, h.

Let $\hat{\mathbf{s}} = \mathbf{H}^T\mathbf{s}$, $\hat{\mathbf{b}}_i := \mathbf{R}_i^T\bar{\mathbf{b}} + \underline{\mathbf{b}}_i = \mathbf{G}_\lambda^T\hat{\mathbf{s}} + \hat{\mathbf{e}}_i$. By Lemma 4, we can use Algorithm 2 to get every entry of $\hat{\mathbf{s}}$ one by one. Because $\mathbf{H}$ is invertible, we can use the value of $\hat{\mathbf{s}}$ to compute $\mathbf{s}$, as well as $\mathbf{e} = \mathbf{b} - \mathbf{A}^T\mathbf{s}$. □