Lizard: Cut off the Tail!
Practical Post-Quantum Public-Key Encryption from LWE and LWR

Jung Hee Cheon, Duhyeong Kim, Joohee Lee, and Yongsoo Song

Seoul National University (SNU), Republic of Korea,
{jhcheon,doodoo1204,skfro6360,lucius05}@snu.ac.kr

Abstract. The LWE problem has been widely used in many constructions for post-quantum cryptography due to its strong security reduction from the worst-case of lattice hard problems and its lightweight operations. The PKE schemes based on the LWE problem have a simple and fast decryption, but the encryption phase is rather slow due to large parameter size for the leftover hash lemma or expensive Gaussian samplings.

In this paper, we propose a novel PKE scheme, called Lizard, without relying on either of them. The encryption procedure of Lizard first combines several LWE samples as in the previous LWE-based PKEs, but the following step to re-randomize this combination before adding a plaintext is different: it removes several least significant bits of each component of the computed vector rather than adding an auxiliary error vector. Lizard is IND-CPA secure under the hardness assumptions of the LWE and LWR problems, and its variant achieves IND-CCA security in the quantum random oracle model.

Our approach accelerates encryption speed to a large extent and also reduces the size of ciphertexts, and Lizard is very competitive for applications requiring fast encryption and decryption phases. In our single-core implementation on a laptop, the encryption and decryption of IND-CCA Lizard with 256-bit plaintext space under 128-bit quantum security take 0.014 and 0.027 milliseconds, which are comparable to those of NTRU. To achieve these results, we further take some advantages of sparse small secrets.

Keywords: Post-Quantum Cryptography, Public-Key Encryption, Learning with Rounding, Learning with Errors

1 Introduction

Since the National Institute of Standards and Technology (NIST) launched a project to develop new quantum-resistant cryptography standards [1], post-quantum cryptography has gained a growing attention at this moment. Lattice-based cryptography, one of the most attractive areas of the post-quantum cryptography, has been studied actively over the last decade due to its distinctive advantages on the strong security, fast implementations, and versatility in many applications. In particular, the Learning with Errors (LWE) problem [35] has very attractive features for many usages due to its rigorous reduction from the worst-case of the lattice problems that are regarded to be hard to solve even after the advance of quantum computers.

The LWE problem was first introduced to construct a Public-Key Encryption (PKE) by Regev [35] in 2005. Some well-known variants of Regev's scheme [21, 33] had a drawback requiring too large parameters to be used in practice. It was improved by Lindner and Peikert [27] using a method to insert noises to a combination of LWE samples in the encryption stage. Recently, several post-quantum key exchanges [4, 10, 9, 17, 32] and one more efficient PKE [14] with sparse small secrets have been proposed on the hardness assumptions of the LWE problem and its ring variant. They enjoy fast performances in practice as well as quantum-resistant security, but the noise sampling caused some inefficiency since the distribution of noises has to be close to the discrete Gaussian distribution.

The *learning with rounding* (LWR) problem, introduced by Banerjee, Peikert and Rosen [6], is a de-randomized version of the LWE problem, which generates an instance using the deterministic rounding process into a smaller modulus instead of adding auxiliary errors. Since the sampling of LWR instances does not contain the Gaussian sampling process, it is rather simpler than that of LWE instances. Up to recently, there have been several researches on the hardness of the LWR problem, which address that the LWR problem is at least as hard as the LWE problem when the number of samples is bounded [5, 6, 8].

Our Contributions. We propose a novel PKE scheme, called Lizard, based on LWE and LWR. Lizard has a conceptually simple encryption procedure consisting of subset sum and rounding operations without Gaussian samplings. Through our delicate cryptanalysis against the LWR problem, we show that the parameters of Lizard can be set as tight as those of the Lindner and Peikert's PKE scheme [27], and so our scheme enjoys two advantages of smaller ciphertext and faster encryption speed compared to their scheme.

Taking some advantages of sparse binary secrets, we further show that our PKE scheme Lizard is very practical. We implement Lizard and achieve a comparable performance result to that of NTRU [18, 22, 23] in spite of the better security grounds. We remark that our scheme has stronger security guarantee than NTRU in the sense that our scheme has a provable security from the LWE and LWR problems which have reductions from the standard lattice problems (GapSVP, SIVP), but NTRU does not.[1]

Technical Details. Our PKE scheme consists of Lizard.Setup, Lizard.KeyGen, Lizard.Enc, and Lizard.Dec. In the key generation Lizard.KeyGen, we choose a private key $\mathbf{s}$ and use it to generate several samples of the LWE problem in modulo q. The public key is $(A,\ \mathbf{b} = A\mathbf{s} + \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, where the error term $\mathbf{e}$ is sampled from the discrete Gaussian distribution. To encrypt a plaintext $M \in \mathbb{Z}_t$, we first generate an ephemeral secret vector $\mathbf{r}$ and calculate $(A^T\mathbf{r},\ \langle \mathbf{b}, \mathbf{r}\rangle + (q/t) \cdot M)$. Then, we rescale the vector into a lower modulus $p < q$ using the rounding function defined by

$$\mathbb{Z}_q^{n+1} \to \mathbb{Z}_p^{n+1} \quad , \quad \mathbf{x} \mapsto \lfloor (p/q) \cdot \mathbf{x} \rceil ,$$

for the ciphertext dimension $n+1$. The function $\lfloor \cdot \rceil$ denotes the component-wise rounding of entries to the closest integers.

For the concrete instantiation of our PKE scheme, we take private keys and ephemeral secrets used in encryption procedure from certain small distributions for efficiency. In particular, ephemeral secrets for the encryption procedure are chosen to be binary vectors in $\{0, \pm 1\}^m$ with small Hamming weights. The Hamming weight of ephemeral secret vectors has an effect on the error size after subset sum of the public data, while the secret key size is related to the error caused by rounding into a smaller modulus p. Therefore, the smallness of private keys and ephemeral secrets takes an important role in efficiency of our scheme including encryption speed and ciphertext size.

Cryptanalysis of LWR and Parameter Selection. While various attacks on the LWE problem were proposed, the cryptanalytic hardness of the LWR problem has not been well-understood so far. Considering all known attacks on LWE and LWR, the best attack on the LWR problem is a dual attack combined with Albrecht's combinatorial attack for the sparse secrets [2] in our setup.

[1] The provably secure variant of NTRU [39] is secure with the hardness assumption of ring-LWE, but the ring-LWE problem only has a reduction from a lattice problem with ring structure, not from the standard lattice problems.

Combining the concrete analyses of the correctness condition and the LWR problem, we conclude that the parameters of Lizard, and Lindner and Peikert's PKE are comparable under the same setup of distributions, security level, and decryption failure probability. Consequently, Lizard achieves a better efficiency in ciphertext size and encryption speed compared to Lindner and Peikert's PKE.

We also present our parameter sets for three different security levels based on the best known attacks against LWE and LWR, and the correctness condition, following the methodology of NewHope [4] and Frodo [9]. In particular, we provide the *recommended* parameter set for the long-term security, which remains secure against all known quantum attacks.

Variants of Lizard. Additionally, we describe some useful variants of Lizard: a ring variant RLizard, and an additive homomorphic encryption derived from Lizard. The security of RLizard is based on the hardness of ring-LWE and ring-LWR. Since we use a polynomial instead of a matrix in RLizard, the public key size of RLizard is considerably small compared to that of Lizard. Lizard could also be a post-quantum alternative for additive homomorphic encryptions. The previous schemes [15, 29, 30] appeared to require large parameters [19], or are insecure under the attacks using a quantum computer [37].

Implementation and Comparison. The proposed PKE schemes were implemented in C language and we measured their performances on a Macbook Pro with an Intel core i5 running at 2.9 GHz processor. With 128-bit quantum security, the encryption and decryption of CCA version of Lizard take about 0.014 and 0.027 milliseconds, respectively. The source code of our schemes will be uploaded at `https://github.com/LizardOpenSource/Lizard_c`.

We compare the IND-CCA version of Lizard with NTRU [22, 23] and the recently proposed LWE-based PKE scheme [14] for 128-bit quantum security, which shows comparable results to NTRU in terms of both enc/dec speed and ciphertext size. Moreover, we present implementation results of our IND-CPA scheme Lizard with small plaintext space, and the additive homomorphic encryption derived from Lizard.

Organization. The rest of the paper is organized as follows. In Section 2, we summarize some notations used in this paper, and introduce LWE and LWR. We describe our public-key encryption scheme Lizard based on both LWE and LWR in Section 3, and provide the concrete analysis and parameters of our scheme in Section 4. In Section 5, we propose some variants of Lizard. Finally, we provide implementation results, and compare the performance of our schemes with other lattice-based schemes in Section 6.

2 Preliminaries

2.1 Notation

All logarithms are base 2 unless otherwise indicated. For a positive integer q, we use $\mathbb{Z} \cap (-q/2, q/2]$ as a representative of $\mathbb{Z}_q$. For a real number r, $\lfloor r \rceil$ denotes the nearest integer to r, rounding upwards in case of a tie. We denote vectors in bold, *e.g.,* $\mathbf{a}$, and every vector in this paper is a column vector. The norm $\|\cdot\|$ is always 2-norm in this paper. We denote by $\langle \cdot, \cdot \rangle$ the usual dot product of two vectors. We use $x \leftarrow D$ to denote the sampling x according to the distribution D. It denotes the uniform sampling when D is a finite set. For an integer $n \geq 1$, D^n denotes the product of i.i.d. random variables $D_i \sim D$. We let λ denote the security parameter throughout the paper: all known valid attacks against the cryptographic scheme under scope should take $\Omega(2^\lambda)$ bit operations. A function $\mathsf{negl} : \mathbb{N} \rightarrow \mathbb{R}^+$ is negligible if for every positive polynomial $p(\lambda)$ there exists $\lambda_0 \in \mathbb{N}$ such that $\mathsf{negl}(\lambda) < 1/p(\lambda)$ for all $\lambda > \lambda_0$. For two matrices A and B with the same number of rows, $(A \| B)$ denotes their row concatenation, *i.e.,* for $A \in \mathbb{Z}^{m \times n_1}$ and $B \in \mathbb{Z}^{m \times n_2}$, the $m \times (n_1 + n_2)$ matrix $C = (A \parallel B)$ is defined as $c_{ij} = \begin{cases} a_{i,j} & 1 \leq j \leq n_1 \\ b_{i,(j-n_1)} & n_1 < j \leq n_1 + n_2 \end{cases}$.

2.2 Distributions

For a positive integer q, we define $\mathcal{U}_q$ by the uniform distribution over $\mathbb{Z}_q$. For a real $\sigma > 0$, the discrete Gaussian distribution of parameter σ, denoted by $\mathcal{D}G_\sigma$, is a probability distribution with support $\mathbb{Z}$ that assigns a probability proportional to $\exp(-\pi x^2/\sigma^2)$ to each $x \in \mathbb{Z}$. Note that the variance of $\mathcal{D}G_\sigma$ is very close to $\sigma^2/2\pi$ unless σ is very small.

For an integer $0 \leq h \leq n$, the distribution $\mathcal{H}WT_n(h)$ samples a vector uniformly from $\{0, \pm 1\}^n$, under the condition that it has exactly h nonzero entries.

For a real number $0 < \rho < 1$, the distribution $\mathcal{Z}O_n(\rho)$ samples a vector $\mathbf{v}$ from $\{0, \pm 1\}^n$ where each component v_i of the vector $\mathbf{v}$ is chosen satisfying $\Pr[v_i = 0] = 1 - \rho$ and $\Pr[v_i = 1] = \rho/2 = \Pr[v_i = -1]$.

2.3 Learning with Errors

Since Regev [35] introduced the *learning with errors* (LWE) problem, a lot of cryptosystems based on this problem have been proposed relying on its versatility. For an n-dimensional vector $\mathbf{s} \in \mathbb{Z}^n$ and an error distribution χ over $\mathbb{Z}$, the LWE distribution $A^{\mathsf{LWE}}_{n,q,\chi}(\mathbf{s})$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is obtained by choosing a vector $\mathbf{a}$ uniformly and randomly from $\mathbb{Z}_q^n$ and an error e from χ, and outputting

$$(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$

The search LWE problem is to find $\mathbf{s} \in \mathbb{Z}_q$ for given arbitrarily many independent samples $(\mathbf{a}_i, b_i)$ from $A^{\mathsf{LWE}}_{n,q,\chi}(\mathbf{s})$. The decision LWE, denoted by $\mathsf{LWE}_{n,q,\chi}(\mathcal{D})$, aims to distinguish the distribution $A^{\mathsf{LWE}}_{n,q,\chi}(\mathbf{s})$ from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ with non-negligible advantage, for a fixed $\mathbf{s} \leftarrow \mathcal{D}$. When the number of samples are limited by m, we denote the problem by $\mathsf{LWE}_{n,m,q,\chi}(\mathcal{D})$.

In this paper, we only consider the discrete Gaussian $\chi = \mathcal{D}G_{\alpha q}$ as an error distribution where α is the error rate in $(0, 1)$, so α will substitute the distribution χ in description of LWE problem, say $\mathsf{LWE}_{n,m,q,a}(\mathcal{D})$. The LWE problem is self-reducible, so we usually omit the key distribution $\mathcal{D}$ when it is a uniform distribution over $\mathbb{Z}_q^n$.

The hardness of the decision LWE problem is guaranteed by the worst case hardness of the standard lattice problems: the decision version of the *shortest vector problem* (GapSVP), and the *shortest independent vectors problem* (SIVP). After Regev [35] presented the quantum reduction from those lattice problems to the LWE problem, Peikert et al. [12, 31] improved the reduction to a classical version for significantly worse parameter; the dimension should be the size of $n \log q$. In this case, note that the reduction holds only for the GapSVP, not SIVP.

After the works on the connection between the LWE problem and some lattice problems, some variants of LWE, of which the secret distributions are modified from the uniform distribution, were proposed. In [12], Brakerski et al. proved that the LWE problem with binary secret is at least as hard as the original LWE problem. Following the approach of [12], Cheon et al. [14] proved the hardness of the LWE problem with sparse secret, *i.e.*, the number of non-zero components of the secret vector is a constant.

As results of Theorem 4 in [14], the hardness of the LWE problems with (sparse) small secret, $\mathsf{LWE}_{n,m,q,\beta}(\mathcal{H}WT_n(h))$ and $\mathsf{LWE}_{n,m,q,\beta}(\mathcal{Z}O_n(\rho))$, are guaranteed by the following theorem.

Theorem 1. *(Informal) For positive integers m, n, k, q, h, $0 < \alpha, \beta < 1$ and $0 < \rho < 1$, following statements hold:*

1. *If* $\log({}_nC_h) + h > k\log q$ *and* $\beta > \alpha\sqrt{10h}$*, then the* $\mathsf{LWE}_{n,m,q,\beta}(\mathcal{HWT}_n(h))$ *problem is at least as hard as the* $\mathsf{LWE}_{k,m,q,\alpha}$ *problem.*
2. *If* $\left((1-\rho)\log\left(\frac{1}{1-\rho}\right) + \rho\log\frac{2}{\rho}\right)n > k\log q$ *and* $\beta > \alpha\sqrt{10n}$*, the* $\mathsf{LWE}_{n,m,q,\beta}(\mathcal{ZO}_n(\rho))$ *problem is at least as hard as the* $\mathsf{LWE}_{k,m,q,\alpha}$ *problem.*

In [11, 33, 34], to pack a string of plaintexts in a ciphertext, LWE with single secret was generalized to LWE with multiple secrets. An instance of multi-secret LWE is $(\mathbf{a}, \langle\mathbf{a},\mathbf{s}_1\rangle + \mathbf{e}_1, ..., \langle\mathbf{a},\mathbf{s}_k\rangle + \mathbf{e}_k)$ where $\mathbf{s}_1, ..., \mathbf{s}_k$ are secret vectors and $\mathbf{e}_1, ..., \mathbf{e}_k$ are independently chosen error vectors. Using the hybrid argument, multi-secret LWE is proved to be at least as hard as LWE with single secret.

2.4 Learning with Rounding

The LWR problem was firstly introduced by Banerjee et al. [6] to improve the efficiency of pseudorandom generator (PRG) based on the LWE problem. Unlikely to the LWE problem, errors in the LWR problem are deterministic so that the problem is so-called a "derandomized" version of the LWE problem. To hide secret information, the LWR problem uses a rounding by a modulus p instead of inserting errors. Then, the deterministic error is created by scaling down from $\mathbb{Z}_q$ to $\mathbb{Z}_p$.

For an n-dimensional vector $\mathbf{s}$ over $\mathbb{Z}_q$, the LWR distribution $A^{\mathsf{LWR}}_{n,q,p}(\mathbf{s})$ over $\mathbb{Z}_q^n \times \mathbb{Z}_p$ is obtained by choosing a vector $\mathbf{a}$ from $\mathbb{Z}_q^n$ uniform randomly, and returning

$$\left(\mathbf{a}, \left\lfloor \frac{p}{q} \cdot (\langle\mathbf{a},\mathbf{s}\rangle \bmod q)\right\rceil\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_p.$$

As in the LWE problem, $A^{\mathsf{LWR}}_{n,m,q,p}(\mathbf{s})$ denotes the distribution of m samples from $A^{\mathsf{LWR}}_{n,q,p}(\mathbf{s})$; that is contained in $\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_p^m$. The search LWR problem are defined respectively as finding secret $\mathbf{s}$ just as same as the search version of LWE problem. In contrary, the decision $\mathsf{LWR}_{n,m,q,p}(\mathcal{D})$ problem aims to distinguish the distribution $A^{\mathsf{LWR}}_{n,q,p}(\mathbf{s})$ from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_p$ with m instances for a fixed $\mathbf{s} \leftarrow \mathcal{D}$.

In [6], Banerjee et al. proved that there is an efficient reduction from the LWE problem to the LWR problem for a modulus q of super-polynomial size. Later, the follow-up works by Alwen et al. [5] and Bogdanov et al. [8] improved the reduction by eliminating the restriction on modulus size and adding a condition of the bound of the number of samples. In particular, the reduction by Bogdanov et al. works when $2mBp/q$ is a constant, where B is a bound of errors in the LWE problem, m is the number of samples in both problems, and p is the rounding modulus in the LWR problem. That is, the rounding modulus p is proportional to $1/m$ for fixed q and B. Since the reduction from LWE to LWR is independent of the secret distribution, the hardness of $\mathsf{LWR}_{n,m,q,p}(\mathcal{HWT}_n(h))$ and $\mathsf{LWR}_{n,m,q,p}(\mathcal{ZO}_n(\rho))$ is obtained from that of the LWE problems with corresponding secret distributions.

2.5 Ring variants of LWE and LWR

In [28], Lyubashevsky et al. deal with the LWE problem over rings, namely ring-LWE. For positive integers n and q, and an irreducible polynomial $g(x) \in \mathbb{Z}[x]$ of degree n, we define the ring $R = \mathbb{Z}[x]/(g(x))$ and its quotient ring modulo q, $R_q = \mathbb{Z}_q[x]/(g(x))$. The ring-LWE problem is to distinguish between the uniform distribution and the distribution of $(a, a\cdot s + e) \in R_q^2$ where a is uniform randomly chosen polynomial, e is chosen from a error distribution, and s is a secret polynomial.

Due to the efficiency and compactness of ring-LWE, many lattice-based cryptosystems are constructed as *ring-*LWE *based*, rather than LWE-based. Similarly to LWE, the ring-LWE problem over the ring R is at least as hard as the search version of approximate SVP over the ideal lattices of R, in the sense of quantum reduction.

The ring variant of LWR is introduced in [6, 8] as an analogue of LWR. In the ring-LWR problem, the vectors chosen from $\mathbb{Z}_q^n$ are substituted by polynomials in R_q, *i.e.*, the ring-LWR instance for a secret polynomial $s \in R_q$ is

$$\left(a, \left\lfloor \frac{p}{q} \cdot a \cdot s \right\rceil\right) \in R_q \times R_p.$$

where $\lfloor (p/q) \cdot a \cdot s \rceil$ is obtained by applying the rounding function to each coefficient of $(p/q) \cdot a \cdot s$. The search and decision ring-LWR problems are defined the same way as the LWR problem, but over rings.

In [6], Banerjee et al. proved that decision ring-LWR is at least as hard as decision ring-LWE for sufficiently large modulus. Later, reduction from search ring-LWE to search ring-LWR was constructed in overall scope of the modulus [6] when the number of samples is bounded.

3 (LWE+LWR)-based Public-key Encryption Scheme

In this section, we present a (probabilistic) public-key encryption scheme Lizard based on both the LWE and LWR problems with provable security. Our construction has several advantages: one is that we could compress the ciphertext size by scaling it down from $\mathbb{Z}_q$ to $\mathbb{Z}_p$ where p is the rounding modulus, and the other is that we speed up the encryption algorithm by eliminating the Gaussian sampling process.

3.1 The Construction of Lizard

We now describe our public-key encryption scheme based on both the LWE and LWR problems. The public key consists of m number of n dimensional LWE samples, and encryptions of zero form $(n + \ell)$ samples of m dimensional LWR where ℓ is the dimension of plaintext vectors. The scheme is described as follows:

- $\mathsf{Lizard.Setup}(1^\lambda)$: Choose positive integers m, n, q, p, t and ℓ. Choose private key distribution $\mathcal{D}_s$ over $\mathbb{Z}^n$, ephemeral secret distribution $\mathcal{D}_r$ over $\mathbb{Z}^m$, and parameter σ for discrete Gaussian distribution $\mathcal{D}G_\sigma$. Output $params \leftarrow (m, n, q, p, t, \ell, \mathcal{D}_s, \mathcal{D}_r, \sigma)$.
- $\mathsf{Lizard.KeyGen}(params)$: Generate a random matrix $A \leftarrow \mathbb{Z}_q^{m \times n}$. Choose a secret matrix $S = (\mathbf{s}_1 \| \cdots \| \mathbf{s}_\ell)$ by sampling column vectors $\mathbf{s}_i \in \mathbb{Z}^n$ independently from the distribution $\mathcal{D}_s$. Generate an error matrix $E = (\mathbf{e}_1 \| \cdots \| \mathbf{e}_\ell)$ from $\mathcal{D}G_\sigma^{m \times \ell}$ and let $B \leftarrow AS + E \in \mathbb{Z}_q^{m \times \ell}$ where the operations are held in modular q. Output the public key $\mathsf{pk} \leftarrow (A \| B) \in \mathbb{Z}_q^{m \times (n+\ell)}$ and the secret key $\mathsf{sk} \leftarrow S \in \mathbb{Z}^{n \times \ell}$.
- $\mathsf{Lizard.Enc}_{\mathsf{pk}}(\mathbf{m})$: For a plaintext $\mathbf{m} = (m_i)_{1 \le i \le \ell} \in \mathbb{Z}_t^\ell$, choose an m dimensional vector $\mathbf{r} \in \mathbb{Z}^m$ from the distribution $\mathcal{D}_r$. Compute the vectors $\mathbf{c}_1' \leftarrow A^T \mathbf{r}$ and $\mathbf{c}_2' \leftarrow B^T \mathbf{r}$ over $\mathbb{Z}_q$, and output the vector
$$\mathbf{c} \leftarrow (\mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_p^{n+\ell}$$
where $\mathbf{c}_1 \leftarrow \lfloor (p/q) \cdot \mathbf{c}_1' \rceil \in \mathbb{Z}_p^n$ and $\mathbf{c}_2 \leftarrow \lfloor (p/t) \cdot \mathbf{m} + (p/q) \cdot \mathbf{c}_2' \rceil \in \mathbb{Z}_p^\ell$.
- $\mathsf{Lizard.Dec}_{\mathsf{sk}}(\mathbf{c})$: For a ciphertext $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_p^{n+\ell}$, compute and output the vector
$$\mathbf{m}' \leftarrow \left\lfloor \frac{t}{p} (\mathbf{c}_2 - S^T \mathbf{c}_1) \right\rceil \pmod{t}.$$

We will assume that $t \mid p \mid q$ in the rest of paper. This restriction simplifies the encryption procedure (*e.g.,* $(p/t) \cdot \mathbf{m}$ is a vector of integers) and makes the implementation of the rounding procedure $x \mapsto \lfloor (p/q) \cdot x \rceil$ faster. However, our scheme still works correctly for parameters not satisfying this condition.

3.2 Correctness and Security

The following lemma shows a required condition of parameter setup to ensure the correctness of our PKE scheme.

Lemma 1 (Correctness). *The PKE scheme Lizard works correctly as long as the following inequality holds for the security parameter λ:*

$$\Pr\left[|\langle \mathbf{e}, \mathbf{r} \rangle + \langle \mathbf{s}, \mathbf{f} \rangle| \geq \frac{q}{2t} - \frac{q}{2p} : \mathbf{e} \leftarrow \mathcal{D}G_{\sigma}^{m}, \mathbf{r} \leftarrow \mathcal{D}_r, \mathbf{s} \leftarrow \mathcal{D}_s, \mathbf{f} \leftarrow \mathbb{Z}_{q/p}^{n} \right] < \mathsf{negl}(\lambda).$$

Proof. Let $\mathbf{r} \in \mathbb{Z}^m$ be a vector sampled from $\mathcal{D}_r$ in our encryption procedure, and let $\mathbf{c}' = (\mathbf{c}'_1, \mathbf{c}'_2) \leftarrow (A^T\mathbf{r}, B^T\mathbf{r}) \in \mathbb{Z}_q^{n+\ell}$. The output ciphertext is $\mathbf{c} \leftarrow (\mathbf{c}_1 = \lfloor (p/q) \cdot \mathbf{c}'_1 \rceil, \mathbf{c}_2 = \lfloor (p/t) \cdot \mathbf{m} + (p/q) \cdot \mathbf{c}'_2 \rceil)$.

Let $\mathbf{f}_1 \leftarrow \mathbf{c}'_1 \pmod{q/p} \in \mathbb{Z}_{q/p}^n$ and $\mathbf{f}_2 \leftarrow \mathbf{c}'_2 \pmod{q/p} \in \mathbb{Z}_{q/p}^{\ell}$ be the vectors satisfying $(q/p) \cdot \mathbf{c}_1 = \mathbf{c}'_1 - \mathbf{f}_1$ and $(q/p) \cdot (\mathbf{c}_2 - (p/t) \cdot \mathbf{m}) = \mathbf{c}'_2 - \mathbf{f}_2$. Note that $\mathbf{f}_1 = A^T\mathbf{r} \pmod{q/p}$ is uniformly and randomly distributed over $\mathbb{Z}_{q/p}^n$ independently from the choice of $\mathbf{r}$, $\mathbf{e}$, and $\mathbf{s}$. Then for any $1 \leq i \leq \ell$, the i-th component of $\mathbf{c}_2 - S^T\mathbf{c}_1 \in \mathbb{Z}_q^{\ell}$ is

$$\begin{aligned}(\mathbf{c}_2 - S^T\mathbf{c}_1)[i] &= (p/t) \cdot m_i + (p/q) \cdot (\mathbf{c}'_2 - S^T\mathbf{c}'_1)[i] - (p/q) \cdot (\mathbf{f}_2[i] - \langle \mathbf{s}_i, \mathbf{f}_1 \rangle) \\ &= (p/t) \cdot m_i + (p/q) \cdot (\langle \mathbf{e}_i, \mathbf{r} \rangle + \langle \mathbf{s}_i, \mathbf{f}_1 \rangle) - (p/q) \cdot \mathbf{f}_2[i] \\ &= (p/t) \cdot m_i + \lfloor (p/q) \cdot (\langle \mathbf{e}_i, \mathbf{r} \rangle + \langle \mathbf{s}_i, \mathbf{f}_1 \rangle) \rceil\end{aligned}$$

since $\mathbf{f}_2 = (AS + E)^T\mathbf{r} = S^T\mathbf{f}_1 + E^T\mathbf{r} \pmod{q/p}$. Therefore, the correctness of our scheme is guaranteed if the encryption error is bounded by $p/2t$, or equivalently, $|\langle \mathbf{e}_i, \mathbf{r} \rangle + \langle \mathbf{s}_i, \mathbf{f}_1 \rangle| < q/2t - q/2p$ with an overwhelming probability. □

We argue that the proposed encryption scheme is *IND-CPA secure* under the hardness assumptions of the LWE problem and the LWR problem. The following theorem gives an explicit proof of our argument on security.

Theorem 2 (Security). *The PKE scheme Lizard is IND-CPA secure under the hardness assumption of* $\mathsf{LWE}_{n,m,q,\mathcal{D}G_\sigma}(\mathcal{D}_s)$ *and* $\mathsf{LWR}_{m,n+\ell,q,p}(\mathcal{D}_r)$.

Proof. An encryption of $\mathbf{m}$ can be generated by adding $(p/t) \cdot \mathbf{m}$ to an encryption of zero. Hence, it is enough to show that the pair of public information $\mathsf{pk} = (A\|B) \leftarrow \mathsf{Lizard.KeyGen}(params)$ and encryption of zero $\mathbf{c} \leftarrow \mathsf{Lizard.Enc}_{\mathsf{pk}}(\mathbf{0})$ is computationally indistinguishable from the uniform distribution over $\mathbb{Z}_q^{m \times (n+\ell)} \times \mathbb{Z}_q^{n+\ell}$ for a parameter set $params \leftarrow \mathsf{Lizard.Setup}(1^\lambda)$.

- $\mathcal{D}_0 = \{(\mathsf{pk}, \mathbf{c}) : \mathsf{pk} \leftarrow \mathsf{Lizard.KeyGen}(params), \mathbf{c} \leftarrow \mathsf{Lizard.Enc}_{\mathsf{pk}}(\mathbf{0})\}$.
- $\mathcal{D}_1 = \{(\mathsf{pk}, \mathbf{c}) : \mathsf{pk} \leftarrow \mathbb{Z}_q^{m \times (n+\ell)}, \mathbf{c} \leftarrow \mathsf{Lizard.Enc}_{\mathsf{pk}}(\mathbf{0})\}$.
- $\mathcal{D}_2 = \{(\mathsf{pk}, \mathbf{c}) : \mathsf{pk} \leftarrow \mathbb{Z}_q^{m \times (n+\ell)}, \mathbf{c} \leftarrow \mathbb{Z}_p^{n+\ell}\}$.

The public key $\mathsf{pk} = (A\|B) \leftarrow \mathsf{Lizard.KeyGen}(params)$ is generated by sampling m instances of LWE problem with ℓ independent secret vectors $\mathbf{s}_1, \ldots, \mathbf{s}_\ell \leftarrow \mathcal{D}_s$. In addition, the multi-secret LWE problem is no easier than ordinary LWE problem as noted in Section 2.3. Hence, distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ are computationally indistinguishable under the $\mathsf{LWE}_{n,m,q,\mathcal{D}G_\sigma}(\mathcal{D}_s)$ assumption.

Now assume that pk is uniform random over $\mathbb{Z}_q^{m \times (n+\ell)}$. Then pk and $\mathbf{c} \leftarrow \mathsf{Lizard.Enc}_{\mathsf{pk}}(\mathbf{0})$ together form $(n + \ell)$ instances of the m dimensional LWR problem with secret $\mathbf{r} \leftarrow \mathcal{D}_r$. Therefore, distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ are computationally indistinguishable under the $\mathsf{LWR}_{m,n+\ell,q,p}(\mathcal{D}_r)$ assumption.

As a result, distributions $\mathcal{D}_0$ and $\mathcal{D}_2$ are computationally indistinguishable under the hardness assumption of $\mathsf{LWE}_{n,m,q,\mathcal{D}G_\sigma}(\mathcal{D}_s)$ and $\mathsf{LWR}_{m,n+\ell,q,p}(\mathcal{D}_r)$, which denotes the IND-CPA security of the PKE scheme. □

Remark 1. Our IND-CPA PKE scheme Lizard can be naturally converted into two IND-CCA versions: one in the random oracle model using the Fujisaki-Okamoto conversion [20], and the other in the quantum random oracle model using the Targhi-Unruh conversion [40]. In the rest of paper, we denote the CCA version of Lizard by CCALizard. The scheme description of CCALizard is in Appendix A.

3.3 Advantages of (LWE+LWR)-based PKE scheme

In this subsection, we compare Lizard with the previous LWE-based PKE schemes, Regev's scheme (Regev) [35] and Lindner-Peikert's scheme (LP) [27], and show that our scheme has some advantages in performance under a reasonable cryptanalytic assumption about the LWR problem. Instead of the specific descriptions of previous schemes, we will consider a generalized version of the Regev and LP schemes with undetermined small distributions $\mathcal{D}_s$ of secret vector and $\mathcal{D}_r$ of ephemeral vector for encryption.

All three schemes assume the hardness of the LWE problem to guarantee the computational randomness of public information

$$\mathsf{pk} \leftarrow (A \| B = AS + E) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times \ell},$$

where A is a matrix uniformly and randomly chosen from $\mathbb{Z}_q^{m \times n}$, $S = (\mathbf{s}_1 \| \cdots \| \mathbf{s}_\ell)$ is a secret matrix sampled from $\mathcal{D}_s^\ell$, and E is an error matrix sampled from $\mathcal{DG}_\sigma^{m \times \ell}$. This matrix is computationally indistinguishable from a uniform matrix over $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times \ell}$ under $\mathsf{LWE}_{n,q,\mathcal{DG}_\sigma}(\mathcal{D}_s)$ assumption. The main difference of these schemes is shown in the encryption procedure of plaintext $\mathbf{m} \in \mathbb{Z}_t^\ell$.

- $\mathsf{Regev.Enc}_{\mathsf{pk}}(\mathbf{m})$: Choose an m dimensional vector $\mathbf{r} \in \mathbb{Z}^m$ from the distribution $\mathcal{D}_r$. Output the vector $\mathbf{c} \leftarrow (\mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_q^{n+\ell}$ where $\mathbf{c}_1 \leftarrow A^T\mathbf{r}$ and $\mathbf{c}_2 \leftarrow B^T\mathbf{r} + (q/t) \cdot \mathbf{m}$.
- $\mathsf{LP.Enc}_{\mathsf{pk}}(\mathbf{m})$: Choose an m dimensional vector $\mathbf{r} \in \mathbb{Z}^m$ from the distribution $\mathcal{D}_r$ and error vectors $\mathbf{f}_1 \leftarrow \mathcal{DG}_{\sigma'}^n$ and $\mathbf{f}_2 \leftarrow \mathcal{DG}_{\sigma'}^\ell$. Output the vector $\mathbf{c} \leftarrow (\mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_q^{n+\ell}$ where $\mathbf{c}_1 \leftarrow A^T\mathbf{r} - \mathbf{f}_1$ and $\mathbf{c}_2 \leftarrow B^T\mathbf{r} + (q/t) \cdot \mathbf{m} + \mathbf{f}_2$.
- $\mathsf{Lizard.Enc}_{\mathsf{pk}}(\mathbf{m})$: Choose an m dimensional vector $\mathbf{r} \in \mathbb{Z}^m$ from the distribution $\mathcal{D}_r$. Compute the vectors $\mathbf{c}_1' \leftarrow A^T\mathbf{r}$ and $\mathbf{c}_2' \leftarrow B^T\mathbf{r}$ over $\mathbb{Z}_q$, and output the vector $\mathbf{c} \leftarrow (\mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_p^{n+\ell}$ where $\mathbf{c}_1 \leftarrow \lfloor (p/q) \cdot \mathbf{c}_1' \rceil \in \mathbb{Z}_p^n$ and $\mathbf{c}_2 \leftarrow \lfloor (p/q) \cdot \mathbf{c}_2' + (p/t) \cdot \mathbf{m} \rceil \in \mathbb{Z}_p^\ell$.

The Regev scheme applies the leftover hash lemma (LHL) to guarantee the randomness of $(\mathsf{pk}, \mathsf{Lizard.Enc}_{\mathsf{pk}}(\mathbf{m}))$. However, this information-theoretic approach requires huge parameter $m = \Omega((n+\ell)\log q) + \omega(\log \lambda)$ for sufficiently large entropy of $\mathbf{r}$, so the Regev scheme is far less efficient than other two schemes in public key size and encryption speed. In the case of the LP scheme, an encryption of zero forms $(n+\ell)$-number of LWE samples with public information pk. Hence, the conditional distribution of $\mathsf{LP.Enc}_{\mathsf{pk}}(\mathbf{m})$ for given pk is computationally indistinguishable from the uniform distribution $\mathbb{Z}_q^{n+\ell}$ under the $\mathsf{LWE}_{m,n+\ell,q,\mathcal{DG}_{\sigma'}}(\mathcal{D}_r)$ assumption. As described in the previous subsection, Lizard has a similar security proof with LP, but the $\mathsf{LWR}_{m,n+\ell,q,p}(\mathcal{D}_r)$ assumption is used instead of $\mathsf{LWE}_{m,n+\ell,q,\mathcal{DG}_{\sigma'}}(\mathcal{D}_r)$. In summary, Lizard can be viewed as a (LWE + LWR)-based scheme while Regev and LP are represented as (LWE + LHL)-based and (LWE + LWE)-based schemes, respectively.

Now let us consider the required conditions for correctness of schemes. All three schemes has the same decryption structure: for a ciphertext $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$, compute $\mathbf{c}_2 - S^T\mathbf{c}_1$ and extract its most significant bits. In our scheme, an encryption error can be represented as $\lfloor (p/q) \cdot (\langle \mathbf{e}_i, \mathbf{r} \rangle + \langle \mathbf{s}_i, \mathbf{f}_1 \rangle) \rceil$ where $\mathbf{s}_i$ is i-th secret vector, $\mathbf{e}_i$ is an error vector sampled from the discrete Gaussian distribution, $\mathbf{r}$ is a randomly chosen small vector for encryption, and $\mathbf{f}_1$ is a random vector in $\mathbb{Z}_{q/p}^n$ defined in the proof of Lemma 1. This error term should be bounded by $p/2t$ for the correctness of the scheme. Meanwhile, an error term of the Regev scheme can be simply described by $\langle \mathbf{e}_i, \mathbf{r}\rangle$ since an encryption of zero is generated by multiplying a small vector $\mathbf{r}$ to public key; however, this value is comparably larger than other two PKE schemes because of its huge dimension. Finally, in the case of the LP scheme, an encryption $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_q^{n+\ell}$ of $\mathbf{m}$ satisfies $(\mathbf{c}_2 - S^T\mathbf{c}_1)[i] = (q/t)\cdot m_i + \langle \mathbf{e}_i, \mathbf{r}\rangle + \langle \mathbf{s}_i, \mathbf{f}_1\rangle + \mathbf{f}_2[i]$, so its encryption error is expressed as $\langle \mathbf{e}_i, \mathbf{r}\rangle + \langle \mathbf{s}_i, \mathbf{f}_1\rangle + \mathbf{f}_2[i]$. This encryption error should be bounded by $q/2t$ for the correctness of the scheme. The hardness assumption problems and correctness conditions are summarized as follows.

Scheme	Security	Correctness Condition
Regev	$\mathsf{LWE}_{n,m,q,\mathcal{DG}_\sigma}(\mathcal{D}_s)$ + Leftover hash lemma	$\lvert\langle \mathbf{e}_i, \mathbf{r}\rangle\rvert < q/2t$: $\mathbf{e}_i \leftarrow \mathcal{DG}_\sigma^m$, $\mathbf{r} \leftarrow \mathcal{D}_r$
LP	$\mathsf{LWE}_{n,m,q,\mathcal{DG}_\sigma}(\mathcal{D}_s)$ + $\mathsf{LWE}_{m,n+\ell,q,\mathcal{DG}_{\sigma'}}(\mathcal{D}_r)$	$\lvert\langle \mathbf{e}_i, \mathbf{r}\rangle + \langle \mathbf{s}_i, \mathbf{f}_1\rangle + \mathbf{f}_2[i]\rvert < q/2t$: $\mathbf{e}_i \leftarrow \mathcal{DG}_\sigma^m$, $\mathbf{r} \leftarrow \mathcal{D}_r$, $\mathbf{s}_i \leftarrow \mathcal{D}_s$, $\mathbf{f}_1 \leftarrow \mathcal{DG}_{\sigma'}^n$, $\mathbf{f}_2[i] \leftarrow \mathcal{DG}_{\sigma'}$
Lizard	$\mathsf{LWE}_{n,m,q,\mathcal{DG}_\sigma}(\mathcal{D}_s)$ + $\mathsf{LWR}_{m,n+\ell,q,p}(\mathcal{D}_r)$	$\lvert\langle \mathbf{e}_i, \mathbf{r}\rangle + \langle \mathbf{s}_i, \mathbf{f}_1\rangle\rvert < q/2t - q/2p$: $\mathbf{e}_i \leftarrow \mathcal{DG}_\sigma^m$, $\mathbf{r} \leftarrow \mathcal{D}_r$, $\mathbf{s}_i \leftarrow \mathcal{D}_s$, $\mathbf{f}_i \leftarrow \mathbb{Z}_{q/p}^n$

We mainly compare the performances of LP and Lizard that are clearly more efficient than the Regev scheme. Both schemes share the first error term $\langle \mathbf{e}_i, \mathbf{r}\rangle$ of encryption noise. In the next section, we will show that this value is a summation of many independent and identically distributed random variables for various candidate distributions $\mathcal{D}_r$ so that its distribution is close to a normal distribution by the central limit theorem. In the remaining terms, Lizard samples $\mathbf{f}_1$ from uniform distribution $\mathbb{Z}_{q/p}^n$ and has a slightly tighter bound $q/2t - q/2p$, while LP samples $\mathbf{f}_1$ from the discrete Gaussian distribution and has an additional error term $\mathbf{f}_2[i]$. Similar to the first term, $\langle \mathbf{s}_i, \mathbf{f}_1\rangle$ is close to a normal distribution for various candidate distributions of $\mathcal{D}_s$, whose variance depends on $\mathcal{D}_s$ and the variance of entries of $\mathbf{f}_1$. Specifically, if the variance $q^2/12p^2$ of uniform distribution of $\mathbb{Z}_{q/p}$ coincides with the variance $\sigma'^2/2\pi$ of $\mathcal{DG}_{\sigma'}$, then distributions $\langle \mathbf{s}_i, \mathbf{f}_1\rangle$ in Lizard and LP will be statistically close. In this case, the common term $\langle \mathbf{e}_i, \mathbf{r}\rangle + \langle \mathbf{s}_i, \mathbf{f}_1\rangle$ of two schemes will be close to a normal distribution of the same variance σ_{enc}^2. Therefore, the failure probabilities of Lizard and LP are approximately measured by the complementary error function $\Pr[|\langle \mathbf{e}_i, \mathbf{r}\rangle + \langle \mathbf{s}_i, \mathbf{f}_1\rangle| < q/2t - q/2p] \approx \mathrm{erfc}((q/2t - q/2p)/\sqrt{2}\sigma_{enc})$ and $\Pr[|\langle \mathbf{e}_i, \mathbf{r}\rangle + \langle \mathbf{s}_i, \mathbf{f}_1\rangle + \mathbf{f}_2[i]| < q/2t] \approx \mathrm{erfc}((q/2t)/\sqrt{2(\sigma_{enc}^2 + \sigma'^2)})$, respectively. Since $q/2t - q/2p$ is close to $q/2t$ and σ' is very small compared to σ_{enc} in parameter setting, two PKE schemes will have almost the same failure probability. For instance in the case of our recommended parameter set $(t = 2, q = 1024, p = 256, n = 608, \mathcal{D}_s = \mathcal{ZO}_n(1/2), \mathcal{D}_r = \mathcal{HWT}_m(128))$, the failure probability of Lizard and LP is approximately measured by $\mathrm{erfc}((q/2t - q/2p)/\sqrt{2}\sigma_{enc}) \approx 2^{-47.6}$ and $\mathrm{erfc}((q/2t)/\sqrt{2(\sigma_{enc}^2 + \sigma'^2)}) \approx 2^{-48.3}$, respectively.

Moreover, in Section 4.2 about cryptanalytic hardness of the LWR problem, we show that the best known attack complexity against the LWR problem of the moduli q and p is no less than that of the LWE problem with the same dimension, modulus q, and the error distribution $\mathcal{DG}_{\sigma'}$ of the variance $\sigma'^2/2\pi = q^2/12p^2$.

Combining these two results about functionality and security, we derive our conclusion that Lizard achieves a better efficiency compared to LWE-based PKE scheme while guaranteeing the same hardness in cryptanalysis. More precisely, if we set the parameter satisfying $\sigma'^2/2\pi = q^2/12p^2$, then Lizard has simpler and faster encryption phase (rounding instead of Gaussian sampling) and smaller ciphertexts size $(n+\ell)\log p$ than $(n+\ell)\log q$ of the LP scheme while preserving its cryptanalytic security level and decryption failure probability.

	Ciphertext bitsize	Gaussian Sampling in Encryption Phase
LP	$(n+\ell)\log q$	Yes
Lizard	$(n+\ell)\log p$	No

4 Concrete Instantiation and Parameter Selection

4.1 Correctness Conditions for various Distribution Setups

We specify some candidate distributions of $\mathcal{D}_s$ and $\mathcal{D}_r$, and analyze the distribution of encryption noise in this subsection. Recall the following correctness condition of Lemma 1:

$$\Pr\left[|\langle \mathbf{e}, \mathbf{r}\rangle + \langle \mathbf{s}, \mathbf{f}\rangle| \geq \frac{q}{2t} - \frac{q}{2p} : \mathbf{e} \leftarrow \mathcal{DG}_\sigma^m, \mathbf{r} \leftarrow \mathcal{D}_r, \mathbf{s} \leftarrow \mathcal{D}_s, \mathbf{f} \leftarrow \mathbb{Z}_{q/p}^n\right] < \mathsf{negl}(\lambda).$$

First, we suggest three specific candidates distributions of $\mathcal{D}_r$ and analyze the behavior of the first error term $\langle \mathbf{e}, \mathbf{r}\rangle$. In every case, this random variable is very close to a symmetric normal distribution of variance σ_1^2 for some $\sigma_1 > 0$.

1. $\mathcal{D}_r = \mathcal{DG}_{\sigma_r}^m$: Entries of $\mathbf{r}$ are independently sampled from discrete Gaussian distribution $\mathcal{DG}_{\sigma_r}$ of parameter σ_r. Since $\langle \mathbf{e}, \mathbf{r}\rangle = \sum_{i=1}^m e_i r_i$ is the summation of m-number of i.i.d. symmetric random variables of variance $(\sigma^2/2\pi)(\sigma_r^2/2\pi)$, it looks like a symmetric normal distribution of variance $\sigma_1^2 = m(\sigma^2/2\pi)(\sigma_r^2/2\pi)$ by the central limit theorem.
2. $\mathcal{D}_r = \mathcal{ZO}_m(\rho_r)$: Entries of $\mathbf{r}$ are independently sampled from signed binary random variable $\mathcal{ZO}(\rho_r)$ of parameter $\rho_r \in (0,1)$ of which variance is ρ_r. Similar to the first case, $\langle \mathbf{e}, \mathbf{r}\rangle$ is close to a symmetric normal distribution of variance $\sigma_1^2 = m\rho_r(\sigma^2/2\pi)$.
3. $\mathcal{D}_r = \mathcal{HWT}_m(h_r)$: A vector $\mathbf{r} \in \{0, \pm 1\}^m$ is a random binary vector of Hamming weight h_r. In this case, $\langle \mathbf{e}, \mathbf{r}\rangle$ is the summation of h_r-number of i.i.d. discrete Gaussian distribution of parameter σ. This random variable is close to the discrete Gaussian distribution of parameter $\sqrt{h_r}\sigma$, or equivalently, a symmetric normal distribution of variance $\sigma_1^2 = h_r(\sigma^2/2\pi)$.

We also give three candidates for $\mathcal{D}_s$ and analyze the behavior of second error term $\langle \mathbf{s}, \mathbf{f}\rangle = \sum_{i=1}^n \mathbf{s}[i] \cdot \mathbf{f}[i]$ similarly. It will follow a normal distribution of variance σ_2^2 for some $\sigma_2 > 0$.

1. $\mathcal{D}_s = \mathcal{DG}_{\sigma_s}^n$: Entries of $\mathbf{s}$ are independently sampled from the discrete Gaussian distribution $\mathcal{DG}_{\sigma_s}$ of parameter σ_s. Since $\sum_{i=1}^n \mathbf{s}[i] \cdot \mathbf{f}[i]$ is a summation of n-number of i.i.d. symmetric random variables of variance $(\sigma_s^2/2\pi)(q^2/12p^2)$, it looks like a normal distribution of variance $\sigma_2^2 = n(\sigma_s^2/2\pi)(q^2/12p^2)$ by the central limit theorem.

2. $\mathcal{D}_s = \mathcal{ZO}_n(\rho_s)$: Entries of $\mathbf{s}$ are independently sampled from signed binary random variable $\mathcal{ZO}(\rho_s)$ of parameter $\rho_s \in (0,1)$ of which variance is ρ_s. Then $\sum_{i=1}^{n} \mathbf{s}[i] \cdot \mathbf{f}[i]$ is close to the symmetric normal distribution of variance $\sigma_2^2 = n\rho_s(q^2/12p^2)$ by the central limit theorem.
3. $\mathcal{D}_s = \mathcal{HWT}_n(h_s)$: A vector $\mathbf{s}$ is chosen uniformly and randomly from the set of vectors $\{0, \pm 1\}^n$ of Hamming weight h_s. In this case, $\sum_{i=1}^{n} \mathbf{s}[i] \cdot \mathbf{f}[i]$ is the summation of h_s-number of i.i.d. uniform random variables on $\mathbb{Z}_{q/p}$, so is close to the symmetric normal distribution of variance $\sigma_2^2 = h_s(q^2/12p^2)$.

Thus, putting these analysis together, the encryption noise of our scheme behaves as symmetric normal distribution of variance $\sigma_{enc}^2 = \sigma_1^2 + \sigma_2^2$. The correctness of our scheme holds if $(q/2t - q/2p)/\sigma_{enc} = \Omega(\sqrt{\lambda})$, *i.e.* the symmetric normal distribution of variance σ_{enc}^2 is bounded by $q/2t - q/2p$ with an overwhelming probability.

4.2 Concrete Hardness of LWR

In this subsection, we analyze the attack complexity for an LWR instance using lattice basis reduction algorithms, e.g. the BKZ algorithm [13, 36]. We will propose the parameter sets of our CPA-secure encryption scheme according to our analysis on LWR and the state-of-the-art of the LWE attacks at the end of the section. We remark that the attack strategy we describe here to analyze the LWR problem is partly shared in the community working on the lattice cryptography, and independently studied in the recent papers [2, 14] while it has never been applied to analyze the LWR problem.

Assume that we are given $(A, \mathbf{b}) \in \mathbb{Z}_q^{k \times m} \times \mathbb{Z}_p^k$ either from $\mathsf{LWR}_{m,k,q,p}(\mathcal{D}_r)$ or $\mathcal{U}_q^{k \times m} \times \mathcal{U}_p^k$, and want to distinguish whether it is an LWR instance or not. Let us denote an m-dimensional LWR instance of k samples with the secret $\mathbf{r} \in \mathbb{Z}_q^m$ as

$$\left(A,\ \mathbf{b} = \left\lfloor \frac{p}{q} \cdot A\mathbf{r} \right\rceil \right) \in \mathbb{Z}_q^{k \times m} \times \mathbb{Z}_p^k.$$

We first consider the case that $p \mid q$ for simplicity. Letting $\mathbf{f} = -A\mathbf{r} \pmod{q/p}$, we have $(q/p) \cdot \mathbf{b} = A\mathbf{r} + \mathbf{f}$ over $\mathbb{Z}_q$. For input $(A,\ \mathbf{b})$, construct the lattice

$$\Lambda = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}^k \times \mathbb{Z}^m : A^T\mathbf{x} = \mathbf{y} \pmod{q}\}.$$

Assume that $\mathbf{v} = (\mathbf{x}, \mathbf{y})$ is a short vector in the lattice. Then we have

$$\langle \mathbf{x}, (q/p) \cdot \mathbf{b} \rangle = \langle \mathbf{y}, \mathbf{r} \rangle + \langle \mathbf{x}, \mathbf{f} \rangle \pmod{q},$$

which can be reduced to

$$\langle \mathbf{x}, \mathbf{b} \rangle = (p/q) \cdot (\langle \mathbf{y}, \mathbf{r} \rangle + \langle \mathbf{x}, \mathbf{f} \rangle) \pmod{p}. \tag{1}$$

If $(A, \mathbf{b})$ is an LWR instance with secret $\mathbf{r}$ and the vector $\mathbf{v}$ is sufficiently short, then (1) is smaller than p so that it is detectable for an adversary, while $\langle \mathbf{x}, \mathbf{b} \rangle$ is uniformly distributed over $\mathbb{Z}_p$ for the case that the input $(A, \mathbf{b})$ is chosen from uniform. Hence, one can distinguish the LWR instance from uniform by determining $\langle \mathbf{x}, \mathbf{b} \rangle \bmod p$ is small or not.

In order to derive a sharper bound for this attack, it is better to consider the balanced lattice $\Lambda' = \{\mathbf{v}' = (\mathbf{x}, w^{-1}\mathbf{y}) \in \mathbb{Z}^k \times (w^{-1} \cdot \mathbb{Z})^m : (\mathbf{x}, \mathbf{y}) \in \Lambda\}$ using a scaling factor $w > 0$ and find a short vector $\mathbf{v}' \in \Lambda'$. Since each entry of $\mathbf{f}$ behaves as a uniform random variable on $\mathbb{Z}_{q/p}$ of variance $\sigma^2 = q^2/12p^2$, an adversary might choose $w \in \mathbb{R}$ by

$$w^2 = \frac{q^2}{12p^2 \cdot \sigma_r^2},$$

to balance the entries of two terms $\langle \mathbf{y}, \mathbf{r}\rangle$ and $\langle \mathbf{x}, \mathbf{f}\rangle$ in (1) where σ_r^2 denotes the variance of each component of $\mathbf{r}$. For example, the optimal scaling factor for attack is $w = \frac{q}{p} \cdot \sqrt{\frac{m}{12h_r}}$ when the secret $\mathbf{r}$ is chosen from the distribution $\mathcal{HWT}_m(h_r)$. By the central limit theorem, the value (1) follows the Gaussian distribution of standard deviation $\|\mathbf{v}'\|/\sqrt{12}$. Applying the lemmas in [3, 27], the value $\langle \mathbf{x}, \mathbf{b}\rangle$ can be distinguished from the uniform distribution modulo p with advantage $\approx 1/23$ if the inequality $\sqrt{\pi/6} \cdot \|\mathbf{v}'\| < p$ holds.

Now, let us see how short the BKZ output as its input Λ' is. Let $\hat{q} = w^{-1}q$. The lattice Λ' has the dimension $(m+k)$ and the volume $\hat{q}^m$. Running the BKZ algorithm for the lattice Λ', it outputs a short vector $\mathbf{v}' = (\mathbf{x}, w^{-1}\mathbf{y})$ of the size $\|\mathbf{v}'\| \approx \delta^{m+k} \cdot \hat{q}^{\frac{m}{m+k}}$ which can be reduced down to $2^{2\sqrt{m \log \hat{q} \cdot \log \delta}}$ when $m + k = \sqrt{m \log \hat{q} / \log \delta}$. To sum up, the $\mathsf{LWR}_{m,k,q,p}(\mathcal{D}_r)$ problem is secure only if

$$\frac{m \log \hat{q}}{\log^2 \hat{p}} \geq \frac{1}{4 \log \delta}$$

for $\hat{p} = \sqrt{6/\pi} \cdot p$ and $\hat{q} = \sqrt{12}\sigma_r p$, where σ_r^2 is the variance of component of secret vector $\mathbf{r}$.

Example 1. In case that $\mathbf{r}$ is drawn from the distribution $\mathcal{HWT}_m(h_r)$, $\hat{q} = p \cdot \sqrt{\frac{12h_r}{m}}$. If $\mathbf{r}$ is from the distribution $\mathcal{ZO}_n(1/2)$, then $\hat{q} = \sqrt{6} \cdot p$. Albrecht's combinatorial attack [2] for the small or sparse secret can be also applied in these cases so that we propose our parameters according to our attack combined with the combinatorial strategy.

Remark 2. To conclude, the attack complexity of the LWR problem of dimension m, modulus q, and the rounding modulus p is the same with that of the LWE problem of the same dimension m, the same modulus q, and an error rate $\alpha = p^{-1} \cdot \sqrt{\pi/6}$.

This agrees with the view that an LWR sample $(\mathbf{a}, b = \lfloor (p/q) \cdot \langle \mathbf{a}, \mathbf{r}\rangle \rceil) \in \mathbb{Z}_q^m \times \mathbb{Z}_p$ can be naturally seen as a kind of an LWE sample by sending back the value b to an element of $\mathbb{Z}_q$, *i.e.,* $b' = (q/p) \cdot b \in \mathbb{Z}_q$ satisfies $b' = \langle \mathbf{a}, \mathbf{r}\rangle + f \pmod q$ for a small error $f = -\langle \mathbf{a}, \mathbf{r}\rangle \pmod{q/p}$. Note that, in this view, the inserted error is deterministically chosen by random part $\mathbf{a}$ and secret $\mathbf{r}$, but it does not affect on the attack complexity.

4.3 The BKZ complexity

In this subsection, we explain how to set the root Hermite factor δ such that the attack complexities for given δ exceed 2^λ, where λ is the security parameter. We follow the strategies to measure the BKZ complexity in NewHope [4] and Frodo [9]. From the perspective in [4, 9], we review the relations among the root Hermite factor δ, the block size b, and the time complexity T for the BKZ algorithm as follows.

- (pessimistic) T can be estimated as 2^{cb} (about $b2^{cb}$ CPU cycles) in our scheme, where c is some constant. This is an approximate lower bound of the complexity for a single SVP calculation using the sieve algorithm [7, 24–26].
- $\delta = ((\pi b)^{1/b} \cdot b/2\pi e)^{1/2(b-1)}$.

From this, if we fix the constant c, we can calculate T from a given δ.

According to the constant c, we consider the three cases, $c = c_C$ for classical security, $c = c_Q$ for quantum security, and $c = c_P$ for very pessimistic view, following the definitions in [4, 9]. We would explain how to set these constants briefly for self-containedness. On the classical view, the constant c has been studied for a long time, reaching $c_C = 0.292$ (See in [7, 25]). Quantum attacks make the constant c decline. The best known constant is achieved by applying Grover's quantum search algorithm to those sieve algorithms [24, 26], resulting in decrease of c to $c_Q = 0.265$. Since all the algorithms require building lists of $(4/3)^{b/2} = 2^{0.2075b}$ vectors, we set $c_P = 0.2075$ as a pessimistic lower bound of the constant c.

Hence, in the each point of view, to make the attack using the BKZ algorithm as in Section 4.2 infeasible for security parameter $\lambda = 128$, we should set the parameters such that the attack is successful only when $\delta_C \leq 1.003922$, $\delta_Q \leq 1.00367$, and $\delta_P \leq 1.00309$, respectively. This can be shown in simple calculations visualized in Table 1.

Table 1. Our views in BKZ complexity; estimated root Hermite factor δ for BKZ running time 2^{128} (in cycles).

View	c	b	δ
Classical	0.292	409	1.003922
Quantum	0.265	450	1.00367
Paranoid	0.2075	573	1.00309

4.4 Proposed Parameters

In this subsection, we propose our parameter sets adjusted to be secure against the LWR attack in Section 4.2 and the state-of-the-art of the LWE attacks: so far, the best known attack of the LWE problem when the secrets are small and the number of samples is limited is given in the recent proposal [2] which applies the BKW style combinatorial approach to the dual attack on LWE. We remark that one can find a guideline for attacking the LWE problem in [3] as well.

Nowadays, many cryptosystems are threatened not only by developments of the classical attack algorithms on the hard problems, but also by the quantum attacks. We suggest parameter options following the criteria in [4, 9] so that we have two sets called Classical and Recommended according to the security estimates against classical and quantum attacks respectively, and one more set called Paranoid for the pessimistic view. We review the criteria briefly for a self-containedness.

Classical Parameters. This parameter set supplies 128-bit security against the classical attacks, but not enough against quantum attacks.

Recommended Parameters. It provides 128-bit security against all known quantum attacks. We recommend to use this parameter for the long-term security.

Paranoid Parameters. This parameter set would remain secure and have 128-bit security against quantum attacks even if a remarkable improvement towards solving SVP arises.

We present the parameter sets for the case that $\mathcal{D}_s = \mathcal{ZO}_n(1/2)$ and $\mathcal{D}_r = \mathcal{HWT}_m(128)$ in Table 2. To satisfy the correctness condition in Lemma 1, we fix the plaintext modulus $t = 2$ and $h_r = 128$.

The Table 3 shows the time complexity for solving $\mathsf{LWE}_{n,m,q,\alpha}(\mathcal{ZO}_n(1/2))$ and $\mathsf{LWR}_{m,n+\ell,q,p}(\mathcal{HWT}_m(128))$ considering the best known attacks. In the table, the column labeled b denotes the required block size of the BKZ algorithm to achieve a root Hermite factor which draws the best known attack successful against $\mathsf{LWE}_{n,m,q,\alpha}(\mathcal{ZO}_n(1/2))$ and $\mathsf{LWR}_{m,n+\ell,q,p}(\mathcal{HWT}_m(128))$ respectively, and the values in the columns labeled C, Q, and P shows the bit size of required time complexity in CPU cycles measured with the constants c_C, c_Q, and c_P, respectively.

Table 2. Suggested parameter sets for 128-bit security; m and n are dimensions of LWR and LWE, respectively. q is a large modulus in LWE and LWR, and p is a rounding modulus in LWR. α is an error rate in LWE.

	m	n	$\log q$	$\log p$	α^{-1}
Classical	840	544	10	8	171
Recommended	960	608	10	8	182
Paranoid	1450	736	10	8	160

Table 3. Attack complexity of LWE and LWR for the best attack on the suggested parameter sets according to our analysis. Numbers in bold type point to the security claim for the particular parameter set. For example, the recommended set provides 128-bit post-quantum security.

Parameter	Problem	b	C	Q	P
Classical	LWE	409	**128**	117	94
	LWR	421	**132**	120	96
Recommended	LWE	450	140	**128**	102
	LWR	456	142	**130**	103
Paranoid	LWE	575	177	162	**128**
	LWR	577	178	162	**129**

5 Variants of Lizard

In this section, we propose some variants of Lizard : its ring variant and (bounded) additive homomorphic encryption scheme.

5.1 Ring variant of Lizard

Our scheme Lizard has a natural analogue based on the harness of Ring-LWE and Ring-LWR problems. Although the security ground of the ring variant of Lizard, which we call RLizard, is weaker than that of Lizard based on LWE and LWR, the ring variant exploits better key sizes, plaintext expansion rate, and Enc/Dec speed.

We bring some notations for the description of our ring-based encryption scheme. For an integer d, let $\Phi_d(X)$ be the d-th cyclotomic polynomial of degree $n = \phi(d)$. We write the cyclotomic ring and its residue ring modulo an integer q by $R = \mathbb{Z}[X]/(\Phi_d(X))$ and $R_q = \mathbb{Z}_q[X]/(\Phi_d(X))$. We identify the vectors of $\mathbb{Z}_q^n$ with the elements of R_q by $(a_0, ..., a_{n-1}) \mapsto \sum_{i=0}^{n-1} a_i X^i$. For any distribution $\mathcal{D}$ over $\mathbb{Z}_q$, sampling a polynomial $\sum_{i=0}^{n-1} a_i X^i \in R_q$ from D^n denotes sampling the coefficient vector $(a_0, ..., a_{n-1})$ from the distribution. For the simplicity of ring operations, we choose a power-of-two degree in the following description.

- $\mathsf{RLizard.Setup}(1^\lambda)$: Choose positive integers t, p, and q. Let $n \in \mathbb{Z}$ be a power of 2 and $\Phi(X) = X^n + 1$ be the $2n$-th cyclotomic polynomial. Choose h_s, h_r less than or equal to n, a private key distribution $\mathcal{D}_s$ over R^n, an ephemeral secret distribution $\mathcal{D}_r$ over R^n, and a parameter σ for the discrete Gaussian distribution $\mathcal{DG}_\sigma$. Output $params \leftarrow (n, t, p, q, \mathcal{D}_s, \mathcal{D}_r, \sigma)$.
- $\mathsf{RLizard.KeyGen}(params)$: Generate a random polynomial $a \leftarrow R_q$. Sample a secret polynomial $s \leftarrow \mathcal{D}_s$, and an error polynomial $e \leftarrow \mathcal{DG}_\sigma^n$. Let $b = a \cdot s + e \in R_q$. Output the public key $\mathsf{pk} \leftarrow (a, b) \in R_q^2$ and the secret key $\mathsf{sk} \leftarrow s \in R$.

- $\mathsf{RLizard.Enc}_{\mathsf{pk}}(m)$: For a plaintext $m \in R_t = R/tR$, choose $r \leftarrow \mathcal{D}_r$ and compute $c_1' \leftarrow a \cdot r$ and $c_2' \leftarrow b \cdot r$. Output the vector
$$\mathbf{c} \leftarrow (c_1, c_2) \in R_p^2,$$
where $c_1 \leftarrow \lfloor (p/q) \cdot c_1' \rceil \in R_p$ and $c_2 \leftarrow \lfloor (p/t) \cdot m + (p/q) \cdot c_2' \rceil \in R_p$.
- $\mathsf{RLizard.Dec}_{\mathsf{sk}}(\mathbf{c})$: For a ciphertext $\mathbf{c} = (c_1, c_2)$, compute and output the polynomial
$$m' \leftarrow \left\lfloor \frac{t}{p}(c_2 - c_1 \cdot s) \right\rceil \in R_t.$$

Note that all the polynomial multiplications with s or r required in key generation, encryption, and decryption phases can be done very efficiently by shifting and adding vectors.

Parameter Consideration. Since the best known attacks do not utilize the ring structure so far, we analyze the hardness of Ring-LWE as the LWE problem without ring structure as in the previous section. Setting $\mathcal{D}_s = \mathcal{D}_r = \mathcal{HWT}_n(128)$, we can achieve our parameter set: we recommend to use the parameter
$$n = 1024, \quad \log q = 10, \quad p = 256, \quad \alpha^{-1} = 154 \tag{2}$$
to resist all known quantum attacks for the security parameter $\lambda = 128$. For the Challenge and Classical parameter sets, since n should be a power of two, just use the same set as in the condition (2).

Hardness of Ring-LWR. There have been a lot of progress in studying the hardness of the ring-LWR problem. Banerjee et al. [6] proved that the decision version of the ring-LWR problem is harder than that of the ring-LWE problem for large modulus. Bogdanov et al. [8] extended the scope of the modulus, but the extension holds only for the search version of the ring-LWR problem. They stated that the search version of the ring-LWR problem is not easier than that of the ring-LWE problem when the number of samples is bounded with a flexible upper bound [8].

5.2 Additive Homomorphic Encryption Scheme

Our IND-CPA scheme Lizard can be naturally seen as an additive homomorphic encryption supporting the bounded number of additions together with the following addition procedure:

- $\mathsf{Lizard.Add}(\mathbf{c}_1, \cdots, \mathbf{c}_k)$: Output $\sum_{i=1}^{k} \mathbf{c}_i$ through componentwise modular p addition.

Corollary 1 (Correctness). *The Additive homomorphic encryption described above works correctly for k number of homomorphic additions as long as the following inequality holds for security parameter λ:*
$$\Pr\left[|\langle \mathbf{e}, \mathbf{r} \rangle + \langle \mathbf{s}, \mathbf{f} \rangle| \geq \frac{q}{2tk} - \frac{q}{2pk} : \mathbf{e} \leftarrow \mathcal{DG}_{\sigma}^{m}, \mathbf{r} \leftarrow \mathcal{D}_r, \mathbf{s} \leftarrow \mathcal{D}_s, \mathbf{f} \leftarrow \mathbb{Z}_{q/p}^{n} \right] < \mathsf{negl}(\lambda).$$

Proof. This is easily proved by Lemma 1 and the triangle inequality.

6 Implementation

In this section, we present the implementation result and compare it to other lattice-based schemes. We set our counterparts to be the most competitive schemes: NTRU [22, 23], Frodo [9], and one more efficient LWE-based PKE scheme [14], say CHK+. The recently proposed CHK+ scheme [14] was inspired from the Peikert's key encapsulation mechanism (KEM) [32], and they adapt sparse small secrets for LWE to achieve the better performance as in our case. We also measure and present the performance of Lizard for the small plaintext space, and that as an additive homomorphic encryption scheme which allows bounded number of additions.

All the implementations of our schemes, NTRU and the key exchange (KE) scheme Frodo here were written in C, and performed on Macbook Pro containing Intel(R) Core(TM) i5-5287U CPU running at 2.90GHz with Turbo Boost and Multithreading disabled. The version of gcc compiler is 7.1.0, and we compiled our C reference implementation with flags `-O3 -fomit-frame-pointer -mavx2 -march=native -std=c99`. Note that we used AVX2 vector instructions for optimizing the implementation results of our schemes as in [4]. The performances of our schemes in Table 4, 5, and 6 were reported as a mean value across 1000 measurements. We measured the performance of NTRU and Frodo with their open sources uploaded on the github [18] and [38], respectively. The implementation result of IND-CCA version of CHK+ is taken from [14].[2]

Optimization Techniques for Key Generation. In the key generation phase, we need to sample Gaussian errors to make an LWE instance. We follow the approach of Frodo to do it, that is, we sample the errors by reading a precomputed look-up table of its cumulative density function (CDF). This approach for the sampling procedure is faster than previous and gives us error distributions which are very close to discrete Gaussian distributions with respect to the Rényi divergence.

We also adapt the idea of [16]: we can compress our public key by substituting the random matrix A of public key with a random seed. Our scheme with compressed public key is still semantically secure in the random oracle model. This replacement of matrix A to a random seed reduces down the bitsize of public key from $m(n+\ell)\log q$ to $\lambda + m\ell \log q$ for the security parameter λ.

Hash functions in CCALizard. We use SHA-3 for hash functions used in the encryption/decryption phases of CCALizard and public key compression. Since the SHA-3 algorithm takes only about a microsecond, it hardly affect the performance of whole procedure.

6.1 Comparison to Other Lattice-based Schemes

In this subsection, we compare our implementation results to those of NTRU, CHK+, and Frodo for the 128-bit quantum security. To make a fair comparison, we present the implementation of the CCALizard, the IND-CCA version of our scheme in the quantum random oracle model, with the recommended parameters in Table 2. For completeness, we also present the precise description of CCALizard in Appendix A.

CCALizard vs PKEs. As suggested in Table 4, the encryption and decryption speeds of CCALizard are comparable to those of NTRU. Even the ciphertext size of Lizard is only about 1.3 times larger than that of NTRU, which is sufficiently small. Compared to CCA version of CHK+, the encryption and decryption of CCALizard are about 20 times and 8 times faster, respectively.

[2] The experiment of CHK+ was performed on Macbook Pro with an Intel core i5 running at 2.6 GHz processor without parallelization.

Table 4. Comparison of CCALizard, NTRU, and CCA version of CHK+

IND-CCA Encryption	Enc (ms)	Dec (ms)	Ciphertext (bytes)
NTRU	0.029	0.025	816
CCA-CHK+	0.313	0.302	804
CCALizard	**0.014**	**0.027**	**1072**

CCALizard vs Frodo. Frodo [9] is an LWE-based KE with fairly nice performance. This protocol is a Diffie-Hellman style KE on lattices, so one can think of PKE scheme from Frodo as an analogue of the ElGamal encryption. Roughly speaking, Alice's first phase, Bob's phase, and Alice's second phase after receiving the protocol message correspond to key generation, encryption, and decryption procedures, respectively. For example, Bob may mask a plaintext with the computed shared key by XORing and then send the protocol message with the masked message. In this sense, PKE variant of Frodo takes 0.726, 0.909, and 0.157 milliseconds for key generation, encryption, and decryption, respectively. Compared to this PKE, Lizard takes more time for key generation, but its encryption and decryption procedures are much faster.

6.2 Performances of Lizard for special applications

Lizard for small devices. We implement our IND-CPA scheme Lizard with 32-bit plaintext space under 128-bit classical security, which is expected to be utilized on small-device environments.

Table 5. The Performance of Lizard with a 32-bit plaintext space under 128-bit classical IND-CPA security

Ciphertext (bytes)	Public Key (bytes)	Private Key (bytes)	KeyGen (ms)	Enc (ms)	Dec (ms)
576	268,816	4,352	5.221	0.013	0.001

Additive Homomorphic Encryption. Our Lizard can be used as a post-quantum additive homomorphic encryption scheme which support the bounded number of additions. We present the sample result of this case for 256-bit plaintexts and 128-bit quantum security in Table 6.

Table 6. The Performance of the additive homomorphic encryption which supports 100 number of additions

Ciphertext (bytes)	Public Key (bytes)	Private Key (bytes)	KeyGen (ms)	Enc (ms)	Dec (ms)	HomAdd (ms)
2,123	745,296	56,064	20.26	0.013	0.009	0.0005

Acknowledgement. This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-TB1403-03. We would like to thank Leo Ducas for valuable discussions, Peter Schwabe, Andrey Kim, Kyoohyung Han, Sungmin Hong, and some members of Korea Internet & Security Agency (KISA) and Estsoft for optimizing our implementation, and Junbeom Shin for editorial comments.

References

1. Proposed submission requirements and evaluation criteria for the post-quantum cryptography standardization process. `http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-for-proposals-draft-aug-2016.pdf`.
2. Martin R Albrecht. On dual lattice attacks against small-secret lwe and parameter choices in helib and seal. *IACR Cryptology ePrint Archive*, 2017:047, 2017.
3. Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.
4. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum Key Exchange—A New Hope. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 327–343, Austin, TX, August 2016. USENIX Association.
5. Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited. In *Advances in Cryptology–CRYPTO 2013*, pages 57–74. Springer, 2013.
6. Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 719–737. Springer, 2012.
7. Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 10–24. SIAM, 2016.
8. Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen. On the hardness of learning with rounding over small modulus. In *Theory of Cryptography Conference*, pages 209–224. Springer, 2016.
9. Joppe Bos, Craig Costello, Leo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 1006–1018, New York, NY, USA, 2016. ACM.
10. Joppe W Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *2015 IEEE Symposium on Security and Privacy*, pages 553–570. IEEE, 2015.
11. Zvika Brakerski, Craig Gentry, and Shai Halevi. Packed ciphertexts in LWE-based homomorphic encryption. In *Public-Key Cryptography–PKC 2013*, pages 1–13. Springer, 2013.
12. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 575–584. ACM, 2013.
13. Yuanmi Chen and Phong Q Nguyen. BKZ 2.0: Better lattice security estimates. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 1–20. Springer, 2011.
14. Jung Hee Cheon, Kyoohyung Han, Jinsu Kim, Changmin Lee, and Yongha Son. Practical post-quantum public key cryptosystem based on LWE. In *the 19th Annual international Conference on Information Security and Cryptology*, 2016. Available at `https://eprint.iacr.org`.
15. Jung Hee Cheon, Hyung Tae Lee, and Jae Hong Seo. A new additive homomorphic encryption based on the Co-ACD problem. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 287–298. ACM, 2014.
16. Jean-Sébastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In *Advances in Cryptology–EUROCRYPT 2012*, pages 446–464. Springer, 2012.

17. Jintai Ding, Xiang Xie, and Xiaodong Lin. A simple provably secure key exchange scheme based on the learning with errors problem. *IACR Cryptology ePrint Archive*, 2012:688, 2012.
18. Mark Etzel, William Whyte, and Zhenfei Zhang. An open source of NTRU, 2016. `https://github.com/NTRUOpenSourceProject/ntru-crypto`.
19. Pierre-Alain Fouque, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Cryptanalysis of the Co-ACD assumption. In *Annual Cryptology Conference*, pages 561–580. Springer, 2015.
20. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of cryptology*, 26(1):80–101, 2013.
21. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206. ACM, 2008.
22. Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. NTRU: A ring-based public key cryptosystem. In *International Algorithmic Number Theory Symposium*, pages 267–288. Springer, 1998.
23. Nick Howgrave-Graham, Joseph H. Silverman, Ari Singer, and William Whyte. Naep: Provable security in the presence of decryption failures. Cryptology ePrint Archive, Report 2003/172, 2003. `http://eprint.iacr.org/2003/172`.
24. Thijs Laarhoven. *Search problems in cryptography.* PhD thesis, PhD thesis, Eindhoven University of Technology, 2015. `http://www.thijs.com/docs/phd-final.pdf`, 2015.
25. Thijs Laarhoven. Sieving for shortest vectors in lattices using angular locality-sensitive hashing. In *Annual Cryptology Conference*, pages 3–22. Springer, 2015.
26. Thijs Laarhoven, Michele Mosca, and Joop Van De Pol. Finding shortest lattice vectors faster using quantum search. *Designs, Codes and Cryptography*, 77(2-3):375–400, 2015.
27. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In *CryptographersâĂŹ Track at the RSA Conference*, pages 319–339. Springer, 2011.
28. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–23. Springer, 2010.
29. Tatsuaki Okamoto and Shigenori Uchiyama. A new public-key cryptosystem as secure as factoring. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 308–318. Springer, 1998.
30. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 1999.
31. Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 333–342. ACM, 2009.
32. Chris Peikert. Lattice cryptography for the internet. In *International Workshop on Post-Quantum Cryptography*, pages 197–219. Springer, 2014.
33. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *Annual International Cryptology Conference*, pages 554–571. Springer, 2008.
34. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 187–196. ACM, 2008.
35. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.
36. Claus-Peter Schnorr and Martin Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematical programming*, 66(1-3):181–199, 1994.
37. Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134. IEEE, 1994.
38. Douglas Stebila. An opensource of the LWE-based key exchange protocol Frodo, 2016. `https://github.com/lwe-frodo/lwe-frodo`.
39. Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 27–47. Springer, 2011.
40. Ehsan Ebrahimi Targhi and Dominique Unruh. Quantum Security of the Fujisaki-Okamoto and OAEP Transforms. Cryptology ePrint Archive, Report 2015/1210, 2015. `http://eprint.iacr.org/2015/1210`.

A IND-CCA Secure Version of Lizard

In this section, following the hybrid conversion technique Fujisaki-Okamoto conversion [20] (*resp.* Targhi-Unruh conversion [40]), we convert our IND-CPA PKE Lizard into an encryption scheme that is IND-CCA secure in the Random Oracle Model (ROM) (*resp.* Quantum Random oracle Model (QROM)), so-called CCALizard. Since Targhi-Unruh (TU) conversion includes Fujisaki-Okamoto (FO) conversion as a subroutine, we describe our IND-CCA encryption scheme applying TU conversion. We define three hash functions $G : \mathbb{Z}_t^\ell \to \{0,1\}^d$, $H : \{0,1\}^* \to \mathbb{Z}_q^m$ and $H' : \mathbb{Z}_t^\ell \to \mathbb{Z}_t^\ell$, where $\{0,1\}^d$ is a plaintext space of the IND-CCA secure encryption scheme. CCALizard is a hybrid encryption scheme of Lizard in Section 3.1 and the One-Time pad as a symmetric encryption scheme required for the conversion.

- $\mathsf{CCALizard.Setup}(1^\lambda)$: Take $params = (m, n, q, p, t, \ell, \mathcal{D}_s, \mathcal{D}_r, \sigma)$ as same as $\mathsf{Lizard.Setup}(1^\lambda)$. Choose hash functions $G : \mathbb{Z}_t^\ell \to \{0,1\}^d$, $H : \{0,1\}^* \to \mathbb{Z}_q^m$ and $H' : \mathbb{Z}_t^\ell \to \mathbb{Z}_t^\ell$.
- $\mathsf{CCALizard.KeyGen}(params)$: Run and output the secret and public keys $\mathsf{sk} = S$, $\mathsf{pk} = (B \| A) \leftarrow \mathsf{Lizard.KeyGen}(params)$.
- $\mathsf{CCALizard.Enc}_{\mathsf{pk}}(\mathbf{m})$: For a plaintext $\mathbf{m} \in \{0,1\}^d$, choose $\delta \leftarrow \mathbb{Z}_t^\ell$ and compute

$$\begin{aligned} \mathbf{c}_1 &\leftarrow G(\delta) \oplus \mathbf{m}, \\ \mathbf{v} &\leftarrow H(\delta \| \mathbf{c}_1), \\ \mathbf{c}_2 &\leftarrow (\lfloor (p/q) \cdot A^T \mathbf{v} \rceil, \lfloor (p/t) \cdot \delta + (p/q) \cdot B^T \mathbf{v} \rceil), \\ \mathbf{c}_3 &\leftarrow H'(\delta). \end{aligned}$$

  Then, output the ciphertext $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \in \{0,1\}^d \times \mathbb{Z}_p^{n+\ell} \times \mathbb{Z}_t^\ell$.
- $\mathsf{CCALizard.Dec}_{\mathsf{sk}}(\mathbf{c})$: Compute $\delta' \leftarrow \mathsf{Dec}_{\mathsf{sk}}(\mathbf{c}_2)$ and $\mathbf{v}' \leftarrow H(\delta' \| \mathbf{c}_1)$. If $\mathbf{c}_2 = \mathsf{Enc}_{\mathsf{pk}}(\delta'; \mathbf{v}')$ and $\mathbf{c}_3 = H'(\delta')$, compute and output $\mathbf{m}' \leftarrow G(\delta') \oplus \mathbf{c}_1$. Else, abort and output $\perp$.

Here, $\mathsf{Lizard.Enc}_{\mathsf{pk}}(\delta'; \mathbf{v}')$ denotes the encryption of δ' with random vector $\mathbf{v}'$, *i.e.*, $\mathsf{Lizard.Enc}_{\mathsf{pk}}(\delta'; \mathbf{v}') = (\lfloor (p/q) \cdot A^T \mathbf{v}' \rceil, \lfloor (p/t) \cdot \delta' + (p/q) \cdot B^T \mathbf{v}' \rceil)$.

Remark 3. When deleting the hash function H' and all procedures in Enc and Dec related to H' anc $\mathbf{c}_3$, then the scheme is exactly the FO conversion of our IND-CPA encryption scheme.

Remark 4. Assuming the hardness of LWE and LWR, CCALizard is IND-CCA secure in QROM by the Theorem 4 in [40] since Lizard satisfies that the min-entropy of $\mathsf{Lizard.Enc}_{\mathsf{pk}}(0)$ is bounded by $\omega(\log \lambda)$ for every pk with overwhelming probability.