Lizard: Cut Off the Tail! A Practical Post-quantum Public-Key Encryption from LWE and LWR

Jung Hee Cheon[1], Duhyeong Kim[1], Joohee Lee[1(✉)], and Yongsoo Song[2]

[1] Seoul National University, Seoul, Republic of Korea
{jhcheon,doodoo1204,skfro6360}@snu.ac.kr
[2] University of California, San Diego, USA
yongsoosong@ucsd.edu

Abstract. The LWE problem has been widely used in many constructions for post-quantum cryptography due to its reduction from the worst-case of lattice hard problems and the lightweight operations for generating its instances. The PKE schemes based on the LWE problem have a simple and fast decryption, but the encryption phase requires large parameter size for the leftover hash lemma or 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. To the best of our knowledge, Lizard is the first IND-CPA secure PKE under the hardness assumptions of the LWE and LWR problems, and its variant, namely CCALizard, achieves IND-CCA security in the (quantum) random oracle model.

Our approach accelerates the encryption speed to a large extent and also reduces the size of ciphertexts. We present an optimized C implementation of our schemes, which shows outstanding performances with concrete security: On an Intel single core processor, an encryption and decryption for CCALizard with 256-bit plaintext space under 128-bit quantum security take only 32,272 and 47,125 cycles, respectively. To achieve these results, we further take some advantages of sparse small secrets. Lizard is submitted to NIST's post-quantum cryptography standardization process.

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00616, Development of lattice-based post-quantum public-key cryptographic schemes) and Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-TB1403-52, and Duhyeong Kim has been supported by NRF (National Research Foundation of Korea) Grant funded by Korean Government (NRF-2016H1A2A1906584-Global Ph.D. Fellowship Program).

D. Catalano and R. De Prisco (Eds.): SCN 2018, LNCS 11035, pp. 160–177, 2018.
https://doi.org/10.1007/978-3-319-98113-0_9

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 [26], 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 [31] 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 by Regev [31] to construct a Public-Key Encryption (PKE). Some well-known variants of Regev's scheme [21,29] had a drawback requiring too large parameters to be used in practice. It was improved by Lindner and Peikert [25] using a method to insert noises to a combination of LWE samples in the encryption stage. Recently, several post-quantum key exchanges [6,10–12,17,28], key encapsulation mechanism [11], and one more efficient PKE [15] with sparse small secrets have been proposed on the hardness assumptions of the LWE problem and its ring (or module) variant. They enjoy fast performances in practice as well as quantum-resistant security, but the noise sampling causes some overheads.

The *learning with rounding* (LWR) problem, introduced by Banerjee, Peikert and Rosen [8], 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 [7–9].

Our Contributions. We propose a PKE scheme based on LWE and LWR for the first time, called Lizard. Lizard has a conceptually simple encryption procedure consisting of subset sum and rounding operations without Gaussian samplings. We also apply cryptanalytic strategies for LWE to LWR and estimate the concrete hardness of LWR for the first time, which is expected to be useful in the future studies.

Through the 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 [25], so our scheme enjoys two advantages of smaller ciphertext and faster encryption speed compared to their scheme under the same setup of distributions, security level, and decryption failure probability.

Taking some advantages of sparse binary secrets as well, we further show that our PKE scheme Lizard is very practical. We implement CCA variants

of Lizard and achieve a comparable performance to NTRU [18,22,24] in spite of the better security grounds: Our scheme has a 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 $(A^T\mathbf{r},\ \langle \mathbf{b}, \mathbf{r} \rangle) \in \mathbb{Z}_q^{n+1}$. Then, we rescale the vector into a lower modulus $p < q$ using the rounding function defined by

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

where the function $\lfloor \cdot \rceil$ denotes the component-wise rounding of entries to the closest integers. After then, encoded plaintext $\tilde{M} \in \mathbb{Z}_p$ is added to the second component of the rescaled vector.

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 low Hamming weights. The Hamming weight of ephemeral secret vectors has an effect on the error sizes 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 not only in efficiency of our scheme including encryption and decryption speeds, but also in setting feasible parameter sets to achieve negligible decryption failure probabilities.

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 possible attacks on LWE and LWR in our setup, we concluded that the best attack on the LWR problem with sparse small secrets is a variant of dual attack combined with Albrecht's combinatorial attack for the sparse secrets [3].

Through complete analyses on the correctness conditions, we also present our parameter sets for three different security levels based on the best attacks against LWE and LWR, following the methodology of [6,10]. In particular, we provide the *recommended* parameter set for the long-term security, which remains secure against all known quantum attacks. Due to the lack of space, we do not include the complete analyses in the conference version; for more details, see the full version of this paper [16].

[1] A provably secure variant of NTRU [32] is secure under 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.

IND-CCA Variant of Lizard. We present CCA-secure version of Lizard, namely CCALizard. We converted Lizard with negligible decryption failure probability into CCALizard using a variant of Fujisaki-Okamoto transformation [19,20,23,33] which make it IND-CCA PKE in the random oracle model (ROM) and quantum random oracle model (QROM), respectively. Note that CCALizard achieves IND-CCA security in standard ROM with tighter security reduction.

Implementation and Comparison. We provide our implementation results for Lizard and CCALizard. The proposed PKE schemes were implemented in C language and we measured the performances on Linux with an Intel Xeon E5-2620 CPU running at 2.10 GHz processor. With 128-bit quantum security, the encryption and decryption of CCALizard take about 32,272 and 47,125 cycles, respectively. We compare CCALizard with NTRU [22,24] and the recently proposed LWE-based PKE scheme [15], which shows comparable results to NTRU in terms of both enc/dec speed and ciphertext size. Our source code is publicly available at https://github.com/LizardOpenSource/Lizard_c.

Organization. The rest of the paper is organized as follows. In Sect. 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 Sect. 3, presenting its correctness condition, security proof and advantages. Finally, we provide implementation results of our schemes, and compare their performances with other lattice-based schemes in Sect. 4. We also describe an IND-CCA variant of Lizard in Appendix A.

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. For positive integers t, p, and q, $t|p|q$ denotes $t|p$ and $p|q$. 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 $\mathsf{negl} : \mathbb{N} \to \mathbb{R}^+$ 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 $C = (A\|B)$ is defined as $c_{ij} = \begin{cases} a_{i,j} & 1 \leq j \leq n_1 \\ b_{i,(j-n_1)} & n_1 < j \leq n_1 + n_2 \end{cases}$. Let $B_{m,h}$ be the subset of $\{-1, 0, 1\}^m$ of which elements have exactly h number of non-zero components.

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 $\mathcal{DG}_\sigma$, 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 $\mathcal{DG}_\sigma$ is very close to $\sigma^2/2\pi$ unless σ is very small. For an integer $0 \leq h \leq n$, the distribution $\mathcal{HWT}_n(h)$ 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 $\mathcal{ZO}_n(\rho)$ 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 [31] introduced the *learning with errors* (LWE), a number of LWE-based cryptosystems 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 for a distribution $\mathcal{D}$ over $\mathbb{Z}_q^n$ of a secret vector $\mathbf{s}$, 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 $\chi = \mathcal{DG}_{\alpha q}$ 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 $\mathsf{LWE}_{n,m,q,\alpha}(\mathcal{D})$. 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 [31] presented the quantum reduction from those lattice problems to the LWE problem, Peikert et al. [14,27] 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 [14], 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 [14], Cheon et al. [15] 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 [15], the hardness of the LWE problems with (sparse) small secret, $\mathsf{LWE}_{n,m,q,\beta}(\mathcal{HWT}_n(h))$ and $\mathsf{LWE}_{n,m,q,\beta}(\mathcal{ZO}_n(\rho))$, 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* $\left((1-\rho)\log\left(\frac{1}{1-\rho}\right) + \rho - \rho\log\rho\right) n > k\log q$ *and* $\beta > \alpha\sqrt{10n}$, *then 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 [13,29,30], 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 $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}_1 \rangle + \mathbf{e}_1, \ldots, \langle \mathbf{a}, \mathbf{s}_k \rangle + \mathbf{e}_k)$ where $\mathbf{s}_1, \ldots, \mathbf{s}_k$ are secret vectors and $\mathbf{e}_1, \ldots, \mathbf{e}_k$ are independently chosen error vectors. From a standard hybrid argument, multi-secret LWE is proved to be at least as hard as LWE with single secret [1].

2.4 Learning with Rounding

The LWR problem was firstly introduced by Banerjee et al. [8] 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 [8], 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. [7] and Bogdanov et al. [9] 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 bounded, 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 preserves 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.

3 (LWE+LWR)-Based Public-Key Encryption

In this section, we present a (probabilistic) public-key encryption 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 Construction

We now describe our public-key encryption Lizard based on both the LWE and LWR problems. The public key consists of m number of n-dimensional LWE samples with ℓ multiple secrets. A plaintext is an ℓ-dimensional vector of which each component is contained in $\mathbb{Z}_t$, and a ciphertext is $(n+\ell)$-dimensional vector in $\mathbb{Z}_p^{n+\ell}$. The PKE scheme Lizard 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 $\mathcal{DG}_\sigma$. 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 $\mathcal{DG}_\sigma^{m\times\ell}$ and let $B \leftarrow AS + E \in \mathbb{Z}_q^{m\times\ell}$ where the operations are held modulo 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 $\mathbf{c}_2 \leftarrow \lfloor (p/t)\cdot \mathbf{m} \rceil + \lfloor (p/q)\cdot \mathbf{c}_2' \rceil \in \mathbb{Z}_p^\ell$.
- $\mathsf{Lizard.Dec}_{\mathsf{sk}}(\mathbf{c})$: 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 allows us to compute $\mathbf{c}_2$ by a single rounding process, *i.e.,* $\mathbf{c}_2 = \lfloor (p/t)\cdot\mathbf{m} + (p/q)\cdot\mathbf{c}_2' \rceil$, and makes the implementation of rounding procedures faster. However, our scheme still works correctly for parameters not satisfying this condition when $t < p < q$.

3.2 Correctness and Security

The following lemma shows a required condition of parameter setup to ensure the correctness of our PKE scheme. Note that the assumption $t \mid p \mid q$ in Lemma 1 is not necessary for the correctness of our scheme, but it makes the correctness condition more tight.

Lemma 1 (Correctness). *Assuming that $t \mid p \mid q$, the public key encryption Lizard works correctly as long as the following inequality holds for the security parameter λ:*

$$\Pr\left[|\langle \mathbf{e}, \mathbf{r}\rangle + \langle \mathbf{s}, \mathbf{f}\rangle| \geq \frac{q}{2t} - \frac{q}{2p}\right] < \mathsf{negl}(\lambda)$$

where $\mathbf{e} \leftarrow \mathcal{DG}_\sigma^m$, $\mathbf{r} \leftarrow \mathcal{D}_r, \mathbf{s} \leftarrow \mathcal{D}_s$, and $\mathbf{f} \leftarrow \mathbb{Z}_{q/p}^n$.

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 $\mathbf{c} \leftarrow (\mathbf{c}_1 = \lfloor (p/q)\cdot \mathbf{c}_1' \rceil, \mathbf{c}_2 = \lfloor (p/t)\cdot \mathbf{m}\rceil + \lfloor (p/q)\cdot \mathbf{c}_2' \rceil)$.

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 $(q/p)\cdot(\mathbf{c}_2 - \lfloor (p/t)\cdot\mathbf{m}\rceil) = \mathbf{c}_2' - \mathbf{f}_2$. 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

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

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 $\mathsf{LWE}_{n,m,q,\mathcal{DG}_\sigma}(\mathcal{D}_s)$ and $\mathsf{LWR}_{m,n+\ell,q,p}(\mathcal{D}_r)$.*

Proof. An encryption of $\mathbf{m}$ can be generated by adding $\lfloor (p/t)\cdot\mathbf{m}\rceil$ 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 Sect. 2.3. Hence, distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ are computationally indistinguishable under the $\mathsf{LWE}_{n,m,q,\mathcal{DG}_\sigma}(\mathcal{D}_s)$ 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 $\mathsf{LWE}_{n,m,q,\mathcal{DG}_\sigma}(\mathcal{D}_s)$ and $\mathsf{LWR}_{m,n+\ell,q,p}(\mathcal{D}_r)$, which denotes the IND-CPA security of the PKE scheme. □

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) [31] and Lindner-Peikert's scheme (LP) [25], 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 generalized versions 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[2].

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 $\mathsf{LWE}_{n,m,q,\sigma}(\mathcal{D}_s)$ 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 $\mathbf{c}_2 \leftarrow \lfloor(p/q)\cdot\mathbf{c}_2'\rceil + \lfloor(p/t)\cdot\mathbf{m}\rceil \in \mathbb{Z}_p^\ell$.

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,

[2] Hence, the parameter choices of [25] are irrelevant of this comparison. Note that the chosen parameter sets in [25] do not achieve the claimed security anymore, due to many recent attacks in the literatures [3–5].

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 $\mathsf{LWE}_{m,n+\ell,q,\sigma'}(\mathcal{D}_r)$ 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 $\mathsf{LWE}_{m,n+\ell,q,\sigma'}(\mathcal{D}_r)$. 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.

Table 1. Comparison of Lizard, Regev, and LP

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

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 of each scheme are summarized in Table 1.

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. 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 $\mathbb{Z}^n_{q/p}$ 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^2_{enc}. Therefore, the failure probabilities of Lizard and LP are approximately measured by the complementary error function:

$$\Pr[|\langle \mathbf{e}_i, \mathbf{r} \rangle + \langle \mathbf{s}_i, \mathbf{f}_1 \rangle| < \frac{q}{2t} - \frac{q}{2p}] \approx \mathrm{erfc}\left(\frac{q/2t - q/2p}{\sqrt{2}\sigma_{enc}}\right), \text{ and}$$

$$\Pr[|\langle \mathbf{e}_i, \mathbf{r} \rangle + \langle \mathbf{s}_i, \mathbf{f}_1 \rangle + \mathbf{f}_2[i]| < \frac{q}{2t}] \approx \mathrm{erfc}\left(\frac{q/2t}{\sqrt{2(\sigma^2_{enc} + \sigma'^2)}}\right),$$

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 decryption failure probability. For instance, in the case of our recommended parameter set ($t = 2$, $q = 2048$, $p = 512$, $m = 1024$, $n = 536$, $\mathcal{D}_s = \mathcal{ZO}_n(1/2)$, $\mathcal{D}_r = \mathcal{HWT}_m(134)$), the decryption failure probability of Lizard and LP is approximately measured by $\mathrm{erfc}((q/2t - q/2p)/\sqrt{2}\sigma_{enc}) \approx 2^{-154}$ and $\mathrm{erfc}((q/2t)/\sqrt{2(\sigma^2_{enc} + \sigma'^2)}) \approx 2^{-155}$, respectively.

Moreover, in an attacker's point of view, the hardness of LWR is somewhat equivalent to that of LWE: So far, there is no known specialized attack strategy for the deterministic rounding errors so that we applied LWE attacks for LWR to estimate its hardness. It resulted as the following lemma which implies the attack complexity against the LWR problem of the modulus q and rounding modulus 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 $\sigma'^2/2\pi = q^2/12p^2$, in case of applying the dual attack strategies in [5,6,15][3].

Lemma 2. *Let m, k, q and p be positive integers. A lattice reduction algorithm which achieves $\delta > 0$ such that*

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

for $\hat{p} = \sqrt{6/\pi} \cdot p$ and $\hat{q} = \sqrt{12}\sigma_r \cdot p$ where σ_r^2 is the variance of component of secret vector $\mathbf{r}$ leads an algorithm to solve the $\mathsf{LWR}_{m,k,q,p}(\mathcal{D}_r)$ problem with advantage 1/23.

[3] After approving it, Albrecht's combinatorial strategy for sparse secrets in [3] can be exploited naturally: As far as we know, the adjusted dual attack in [3] is the best attack for LWR using sparse signed binary secrets.

Proof. See the full version [16] of our paper.

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 $f = -\langle \mathbf{a}, \mathbf{r} \rangle \pmod{q/p}$.

Combining these two 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 Implementation

In this section, we present our implementation result for Lizard and its CCA version called CCALizard. CCALizard is obtained by applying a variant of Fujisaki-Okamoto (FO) transformation [19,20,23,33] to our Lizard encryption scheme. Full description of CCALizard is presented in Appendix A.

In Sect. 4.1, we propose parameter sets for Lizard (and CCALizard) in three perspectives, respectively. In Sect. 4.2, we present implementation results of Lizard and its CCA version with referred parameters achieving 128-bit quantum security.

4.1 Proposed Parameters

In this section, we propose parameter sets secure against the best attacks on LWE and LWR using lattice basis reduction algorithm. Targeting 128-bit security, we suggest three parameter options following the criteria in [6,10] 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. Note that Recommended parameter set aims to achieve 128-bit quantum security.

Secret Distributions. We instantiate our scheme for the case that $\mathcal{D}_s = \mathcal{ZO}_n(\rho_s)$ and $\mathcal{D}_r = \mathcal{HWT}_m(h_r)$, proposing concrete parameter sets in Table 2. We have some evidence in mind (Theorem 1) that LWE and LWR of sufficiently large dimensions are secure even with the sparse secrets, and the sparse secret in the LWR instance accelerates our encryption phase.

Security Analysis. The security of our instantiation of Lizard relies on both of the LWE and LWR assumptions with signed binary and sparse signed binary secrets, respectively. We considered all known attacks for LWE including those in [5], the recent dual attack [3] for sparse secrets and primal attack revisited in [4], and also applied them to LWR with some helps from the lwe-estimator [2][4]. At the end, we came to the conclusion that the dual attack combined with BKW-style combinatorial attack [3] is the best attack for our LWE and LWR instances. To estimate the attack complexities, we adopted the methodology in [6,10] to calculate the core SVP hardness in BKZ lattice reduction algorithm, setting the time complexity of solving SVP as $T = 2^{0.292b}, 2^{0.265b}$, and $2^{0.2075b}$ for Classical, Recommended, and Paranoid parameter sets, respectively, where b is the BKZ block size. For lack of space, we present a detailed analysis on the dual attack applied for LWR and the attack complexities for parameter sets in the full version of our paper.

Note on Power-of-Twos. We set $t = 2$ to achieve cryptographically negligible decryption failure probability more easily, and set p and q to be power-of-twos for the following reasons: In the LWE and LWR attacks, one can reduce the modulus q to $q' < q$ via modulus switching first and then apply arbitrary attack scenarios. Especially since we use the binary (and even sparse) secrets, the benefits in the considered attacks obtained by the modulus switching overwhelms others with strategies for specific q's as far as we know. Hence, any particular choice for modulus q does not harm the security. Therefore, we set q and p as power-of-twos to make the rounding procedures efficiently done through the bitwise shift process.

Table 2. Suggested parameter sets for 128-bit security; n and m are dimensions of LWE and LWR, respectively. q is a large modulus shared in LWE and LWR, and p is a rounding modulus in LWR. α is an error rate in LWE, and ρ_s and h_r are parameters for secret distributions in LWE and LWR, respectively. ϵ denotes the estimated decryption failure probability.

Parameter	m	n	$\log q$	$\log p$	α^{-1}	ρ_s	h_r	ϵ
Classical	724	480	11	9	303	1/2	128	2^{-154}
Recommended	1024	536	11	9	316	1/2	134	2^{-154}
Paranoid	1024	704	13	9	404	1/2	200	2^{-150}

4.2 Performance and Comparison

We present the implementation results for Lizard and CCALizard in Table 3. Due to the lack of space, we defer a detailed sketch of our implementation which presents symmetric cryptographic primitives involved and techniques to boost up the speed of our algorithms to the full version of this paper.

[4] We used the lwe-estimator [2] reported on July 6th, 2017. We remark that one can find a guideline for attacking the LWE problem in [5].

All the implementations of our schemes were written in C, and performed on an Linux environment containing an Intel Xeon E5-2620 CPU running at 2.10 GHz with Turbo Boost and Multithreading disabled. The gcc compiler version is 5.4.0, and we compiled our C implementation with flags `-O3 -fomit-frame -pointer -march=native -std=c99` for the `x86_64` architecture. Throughout this subsection, the performances of key generations (*resp.* encryptions and decryptions) of our schemes were reported as a mean value across 100 (*resp.* 100000) measurements. We recorded public key sizes of our schemes used in our software.[5]

Table 3. Performances of Lizard and CCALizard with 256-bit plaintexts in milliseconds with recommended parameters in Table 2

Our schemes	KeyGen (ms)	Enc (ms)	Dec (ms)
Lizard	18.185	0.014	0.007
CCALizard	18.131	0.015	0.022

CCALizard *vs.* Lattice-based PKEs. We compare the performance of our CCALizard to those of NTRU [22,24] and an LWE-based PKE in [15], say CCA-CHK+, for the 128-bit quantum security. To make a fair comparison, we present an implementation of CCALizard with the recommended parameters in Table 2, the CCA-secure PKE scheme CCA-CHK+ with 128-bit post-quantum parameters in Table 2 of [15], and NTRU with the parameter set EES743EP1. For NTRU, we get its performance on Intel Core i5-6600 from eBACS (https://bench.cr.yp.to/results-encrypt.html). For CCA-CHK+, we refer the performances from their paper.

We present two implementation results of ours: one for generating the public matrix A with a random function, and the other for replacing A by a 256-bit seed which generates A. The later result is recorded in brackets in Table 4. The CCA-CHK+ scheme is obtained by adapting sparse small secrets for LWE and applying the FO variant conversion [33] to achieve IND-CCA security, as in our cases. It should be noticed that their parameter set is insecure now, and it only achieves *58-bit quantum security* in our perspective with the estimate of the LWE security estimator of Albrecht [2]. NTRU with the parameter set EES743EP1 achieves 159-*bit quantum security* according to the estimates from [6]. As suggested in Table 4, the encryption and decryption speeds, and the ciphertext size of CCALizard are comparable to those of NTRU. Compared to CCA-CHK+, the encryption and decryption of CCALizard are about 25 times and 17 times faster, respectively.

Lizard can be compared to other lattice-based Key Encapsulation Mechanisms (KEM) such as [6,10,11] as well. However, since we focused on improving performances of encryption and decryption rather than key generation, and KEM

[5] Since the data type of each component of public key is `uint16_t` and the modulus q is 2^{11}, our public key can be compressed by a factor 16/11.

Table 4. Comparison of CCALizard, NTRU, and the CCA version of CHK+; Records in brackets are results when generating the public matrix A with a 256-bit seed; "kcycles" denotes kilocycles

CCA-PKE scheme	KeyGen (kcycles)	Enc (kcycles)	Dec (kcycles)	ptxt (bytes)	ctxt (bytes)	pk (KB)	sk (KB)
NTRU	1,136	102	110	59	980	1	1
CCA-CHK+	≈76,700	≈814	≈785	32	804	-	-
CCALizard	**38,074 (34,615)**	**32**	**47**	**32**	**955**	**1,622 (524)**	**34**

usually requires somewhat balanced computational costs for Alice and Bob who want to establish a shared key using the KEM, it is hard to compare Lizard to KEMs in parallel. We note that a ring version of our scheme which can be naturally considered has more balanced features and it is highly competitive as a KEM.

Acknowledgments. We would like to thank Martin Albrecht and Fernando Virdia for valuable discussions on parameter selection. We would also like to thank Leo Ducas, Peter Schwabe, Tsuyoshi Takagi, Yuntao Wang and anonymous SCN 2018 reviewers for their useful comments.

A IND-CCA Variant of Lizard

In this section, we present CCA-secure encryption scheme, say CCALizard, achieved by applying a variant of Fujisaki-Okamoto (FO) transformation [19,20,23,33] to our Lizard encryption scheme. More precisely, we first convert Lizard into IND-CCA Key Encapsulation Mechanism (KEM) applying the transformation in [23], and then combine it with a (one-time) CCA-secure symmetric encryption scheme.

$\mathsf{G} : \mathbb{Z}_t^\ell \to B_{m,h_r}$, $\mathsf{H} : \mathbb{Z}_t^\ell \to \{0,1\}^d$, $\mathsf{H}' : \mathbb{Z}_t^\ell \to \mathbb{Z}_t^\ell$ are the hash functions, where $\{0,1\}^d$ is the plaintext space for CCALizard. Here, $\mathsf{Lizard.Enc}_{\mathsf{pk}}(\delta; \mathbf{v})$ denotes the encryption of δ with the random vector $\mathbf{v}$, *i.e.,* the output of $\mathsf{Lizard.Enc}_{\mathsf{pk}}(\delta; \mathbf{v})$ is $(\lfloor (p/q) \cdot A^T \mathbf{v} \rceil, \lfloor (p/t) \cdot \delta + (p/q) \cdot B^T \mathbf{v} \rceil)$.

CCALizard consists of three algorithms ($\mathsf{CCALizard.KeyGen}$, $\mathsf{CCALizard.Enc}$, $\mathsf{CCALizard.Dec}$). $\mathsf{CCALizard.KeyGen}$ is the same as $\mathsf{Lizard.KeyGen}$, and $\mathsf{CCALizard.Enc}$ and $\mathsf{CCALizard.Dec}$ are as follows:

- $\mathsf{CCALizard.Enc}_{\mathsf{pk}}(\mathbf{m} \in \{0,1\}^d)$:
  - Choose $\delta \leftarrow \mathbb{Z}_t^\ell$.
  - Compute a tuple of vectors $\mathbf{c}_1 := \mathsf{H}(\delta) \oplus \mathbf{m}$, $\mathbf{c}_2 := \mathsf{Lizard.Enc}_{\mathsf{pk}}(\delta; \mathsf{G}(\delta))$, $\mathbf{c}_3 := \mathsf{H}'(\delta)$.
  - Output the ciphertext $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \in \{0,1\}^d \times \mathbb{Z}_p^{n+\ell} \times \mathbb{Z}_t^\ell$.
- $\mathsf{CCALizard.Dec}_{\mathsf{sk}}(\mathbf{c})$:
  - Parse $\mathbf{c}$ into $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \in \{0,1\}^d \times \mathbb{Z}_p^{n+\ell} \times \mathbb{Z}_t^\ell$.

- Compute $\delta' \leftarrow \mathsf{Lizard.Dec}_{\mathsf{sk}}(\mathbf{c}_2)$ and $\mathbf{v}' \leftarrow \mathsf{G}(\delta')$.
- If $(\mathbf{c}_2, \mathbf{c}_3) = (\mathsf{Lizard.Enc}_{\mathsf{pk}}(\delta'; \mathbf{v}'), \mathsf{H}'(\delta'))$, then compute and output $\mathbf{m}' \leftarrow \mathsf{H}(\delta') \oplus \mathbf{c}_1$.
- Otherwise, output $\perp$.

Correctness. If Lizard is correct with the probability $1 - \epsilon$, then CCALizard is correct except with the probability $1 - \epsilon$ in the (quantum) random oracle model [23].

Security. CCALizard achieves tight IND-CCA security in the random oracle model, and non-tight IND-CCA security in the quantum random oracle model. For IND-CCA security in ROM, the hash function H' and the hash value $\mathbf{d}$ is not necessary.

Theorem 3. ([23], *Theorems 3.2 and 3.3*)**.** *For any IND-CCA adversary $\mathcal{B}$ on CCALizard issuing at most q_D queries to the decryption oracle, q_G queries to the random oracle G, and q_H queries to the random oracle H, there exists an IND-CPA adversary $\mathcal{A}$ on Lizard such that*

$$Adv^{CCA}_{\mathsf{CCALizard}}(\mathcal{B}) \leq q_G \cdot \epsilon + \frac{q_H}{2^{\omega(\log \lambda)}} + \frac{2q_G + 1}{t^{\ell}} + 3 \cdot Adv^{CPA}_{Lizard}(\mathcal{A})$$

where λ is a security parameter and ϵ is a decryption failure probability of Lizard and CCALizard.

Theorem 4. ([23], *Theorems 4.4 and 4.5*)**.** *For any IND-CCA quantum adversary $\mathcal{B}$ on CCALizard issuing at most q_D (classical) queries to the decryption oracle, q_G queries to the quantum random oracle G, q_H queries to the quantum random oracle H, and $q_{H'}$ queries to the quantum random oracle H', there exists an IND-CPA quantum adversary $\mathcal{A}$ on Lizard such that*

$$Adv^{CCA}_{\mathsf{CCALizard}}(\mathcal{B}) \leq (q_H + 2q_{H'})\sqrt{8\epsilon(q_G + 1)^2 + (1 + 2q_G)\sqrt{Adv^{CPA}_{Lizard}(\mathcal{A})}}$$

where ϵ is a decryption failure probability of Lizard and CCALizard.

Parameters for CCALizard. We use the recommended parameters in Table 2 for CCALizard and set $t = 2$, $\ell = d = 256$.

References

1. Alamati, N., Peikert, C.: Three's compromised too: circular insecurity for any cycle length from (Ring-)LWE. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 659–680. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_23
2. Albrecht, M.R.: A Sage Module for estimating the concrete security of learning with errors instances (2017). https://bitbucket.org/malb/lwe-estimator
3. Albrecht, M.R.: On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10211, pp. 103–129. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56614-6_4

4. Albrecht, M.R., Göpfert, F., Virdia, F., Wunderer, T.: Revisiting the expected cost of solving uSVP and applications to LWE. Cryptology ePrint Archive, Report 2017/815 (2017, accepted). http://eprint.iacr.org/2017/815. ASIACRYPT 2017
5. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. J. Math. Cryptol. **9**(3), 169–203 (2015)
6. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange—A new hope. In: 25th USENIX Security Symposium, USENIX Security 2016, Austin, TX, pp. 327–343. USENIX Association, August 2016
7. Alwen, J., Krenn, S., Pietrzak, K., Wichs, D.: Learning with rounding, revisited. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 57–74. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_4
8. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 719–737. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_42
9. Bogdanov, A., Guo, S., Masny, D., Richelson, S., Rosen, A.: On the hardness of learning with rounding over small modulus. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9562, pp. 209–224. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49096-9_9
10. Bos, J., et al.: 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 2016, pp. 1006–1018. ACM, New York (2016)
11. Bos, J., et al.: CRYSTALS - Kyber: a CCA-secure module-lattice-based KEM. Cryptology ePrint Archive, Report 2017/634 (2017). http://eprint.iacr.org/2017/634
12. Bos, J.W., Costello, C., Naehrig, M., Stebila, D.: Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In: 2015 IEEE Symposium on Security and Privacy, pp. 553–570. IEEE (2015)
13. Brakerski, Z., Gentry, C., Halevi, S.: Packed ciphertexts in LWE-based homomorphic encryption. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 1–13. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_1
14. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, pp. 575–584. ACM (2013)
15. Cheon, J.H., Han, K., Kim, J., Lee, C., Son, Y.: A practical post-quantum public-key cryptosystem based on LWE. In: Hong, S., Park, J.H. (eds.) ICISC 2016. LNCS, vol. 10157, pp. 51–74. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53177-9_3. https://eprint.iacr.org
16. Cheon, J.H., Kim, D., Lee, J., Song, Y.: Lizard: cut off the tail! Practical post-quantum public-key encryption from LWE and LWR. Cryptology ePrint Archive, Report 2016/1126 (2016). https://eprint.iacr.org/2016/1126
17. Ding, J., Xie, X., Lin, X.: A simple provably secure key exchange scheme based on the learning with errors problem. IACR Cryptology ePrint Archive, 2012:688 (2012)
18. Etzel, M., Whyte, W., Zhang, Z.: An open source of NTRU (2016). https://github.com/NTRUOpenSourceProject/ntru-crypto
19. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_34

20. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. J. Cryptol. **26**, 1–22 (2013)
21. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, pp. 197–206. ACM (2008)
22. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: a ring-based public key cryptosystem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054868
23. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017. LNCS, vol. 10677, pp. 341–371. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_12
24. Howgrave-Graham, N., Silverman, J.H., Singer, A., Whyte, W.: NAEP: provable security in the presence of decryption failures. Cryptology ePrint Archive, Report 2003/172 (2003). http://eprint.iacr.org/2003/172
25. Lindner, R., Peikert, C.: Better key sizes (and attacks) for LWE-based encryption. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 319–339. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19074-2_21
26. National Institute of Standards and Technology: Proposed submission requirements and evaluation criteria for the post-quantum cryptography standardization process (2016). http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-for-proposals-draft-aug-2016.pdf
27. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, pp. 333–342. ACM (2009)
28. Peikert, C.: Lattice cryptography for the internet. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 197–219. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11659-4_12
29. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_31
30. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, pp. 187–196. ACM (2008)
31. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, STOC 2005, pp. 84–93. ACM, New York (2005)
32. Stehlé, D., Steinfeld, R.: Making NTRU as secure as worst-case problems over ideal lattices. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 27–47. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_4
33. Targhi, E.E., Unruh, D.: Quantum security of the Fujisaki-Okamoto and OAEP transforms. Cryptology ePrint Archive, Report 2015/1210 (2015). http://eprint.iacr.org/2015/1210