Improved Lattice-based CCA2-Secure PKE
in the Standard Model

Jiang Zhang[1], Yu Yu[2], Shuqin Fan[1], and Zhenfeng Zhang[3]

[1] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[2] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[3] Trusted Computing and Information Assurance Laboratory,
Institute of Software, Chinese Academy of Sciences, China
`jiangzhang09@gmail.com,yuyu@yuyu.hk, shuqinfan78@163.com, zfzhang@tca.iscas.ac.cn`

Abstract. Based on the identity-based encryption (IBE) from lattices by Agrawal et al. (Eurocrypt'10), Micciancio and Peikert (Eurocrypt'12) presented a CCA1-secure public-key encryption (PKE), which has the best known efficiency in the standard model and can be used to obtain a CCA2-secure PKE from lattices by using the generic BCHK transform (SIAM J. Comput., 2006) with a cost of introducing extra overheads to both computation and storage for the use of other primitives such as signatures and commitments. In this paper, we propose a more efficient standard model CCA2-secure PKE from lattices by carefully combining a different message encoding (which encodes the message into the most significant bits of the LWE's "secret term") with several nice algebraic properties of the tag-based lattice trapdoor and the LWE problem (such as unique witness and additive homomorphism). Compared to the best known lattice-based CCA1-secure PKE in the standard model due to Micciancio and Peikert (Eurocrypt'12), we not only directly achieve the CCA2-security without using any generic transform (and thus do not use signatures or commitments), but also reduce the noise parameter roughly by a factor of 3. This improvement makes our CCA2-secure PKE more efficient in terms of both computation and storage. In particular, when encrypting a 256-bit (resp., 512-bit) message at 128-bit (resp., 256-bit) security, the ciphertext size of our CCA2-secure PKE is even 33-44% (resp., 36-46%) smaller than that of their CCA1-secure PKE.

1   Introduction

In the seminal work of [25], Diffie and Hellman introduced the concept of public-key cryptography. Soon afterwards, Rivest, Shamir and Adleman [56] proposed the first public-key encryption (PKE), which is known as RSA. Since then, PKE has aroused widespread public attention from the community, and has become one of the most fundamental and widely used cryptographic primitives. The basic security notion for PKE (i.e., CPA-security) which was formalized by Goldwasser and Micali [33] roughly requires that it should be computationally infeasible for a passive adversary to obtain any useful information from a honestly generated challenge ciphertext. Later, this notion was enhanced by Naor and Yung [46] to deal with the "lunchtime attack". Specifically, they [46] considered the security against non-adaptive chosen ciphertext attacks (i.e., CCA1-security) for PKE, where the adversary can access a decryption oracle to decrypt any ciphertext of his choice before seeing the challenge ciphertext. Now, the de facto standard security for PKE is CCA2-security [54], where the adversary can adaptively access the decryption oracle during the whole attack period (with a restriction that the decryption oracle cannot be directly used to decrypt the challenge ciphertext). For example, the National Institute of Standards and Technology (NIST) considered CCA2-security as a basic security requirement for the PKE submissions to the post-quantum cryptography (PQC) standardization [47].

By definition, CCA2-security is stronger than CCA1-security, which in turn is stronger than CPA-security. One of the main problems in this area is to construct CCA2-secure PKEs from primitives as weak as possible (e.g., a CPA-secure one). Using the Random Oracle (RO) heuristic, one can efficiently boost a CPA-secure PKE into a CCA2-secure one [30,53,65]. However, a scheme provably secure in the RO model may not be secure in the real world [19], and it is of great theoretical and practical interest to construct CCA2-secure

PKE in the standard model. But this task becomes very challenging and highly non-trivial. In fact, Gertner et al. [32] showed that it is hard, if not impossible, to even construct a CCA1-secure PKE solely from a CPA-secure one in the standard model.

By relying on primitives with "stronger" functionality or security, there are roughly four approaches to CCA2-secure PKEs in the standard model. The first one is due to Naor and Yung [46], who showed a paradigm for transforming CPA-secure PKEs into CCA1-secure ones by using the non-interactive zero-knowledge (NIZK) proofs, which was further extended to achieve CCA2-security [58,26]. The second one is a framework under the name of hash proof systems (HPS) or extractable HPS [24,67], which essentially stems from high-level abstraction of some existing schemes. The third one is the BCHK transform [12] from identity-based encryption (IBE), which was later extended to the more general tag-based encryption (TBE) by Kiltz [37]. The last one follows the generic constructions from special types of injective trapdoor functions [52,57,38], such as lossy trapdoor functions [52] and adaptive trapdoor functions [38].

The above approaches have been shown very useful in constructing CCA2-secure PKEs from various hardness assumptions, but most of the instantiations were based on traditional number theoretic problems such as discrete logarithm and integer factorization, which are not quantum resistant [60]. Compared to the big success in the traditional setting, the progress on designing lattice-based CCA2-secure PKE in the standard model was relatively slow. For example, many practical CCA2-secure PKEs in the traditional setting were obtained by using the generic framework from HPS (e.g., [23]), but it is still hard to construct an HPS from lattices [36,51,11]. Moreover, it is also unclear how to obtain NIZKs from lattices in the standard model [11,72,39]. This means that we currently cannot use the first two approaches to construct standard model CCA2-secure PKEs from lattices.

In fact, almost all existing standard model CCA2-secure PKEs from lattices are, to the best of our knowledge, obtained either by using the techniques from special types of injective trapdoor functions [52,48,68,64] which are typically very inefficient (e.g., having large public key and ciphertext sizes due to the use of Dolev-Dwork-Naor like technique [26,28]), or by applying the BCHK transform from IBEs/TBEs [1,2,21,69,71,70,27]. Based on the standard model IBE from lattices due to Agrawal et al. [1], Micciancio and Peikert [45] presented the best known standard model (tag-based) CCA1-secure PKE from lattices by using a more efficient trapdoor technique and a new message encoding. They [45] also mentioned that the CCA2-security can be achieved by using the generic BCHK transform [12], which has two modes: BCHK-SIG [20] and BCHK-MAC [14]. BCHK-SIG requires (one-time) signatures, and typically incurs noticeable overheads to both computation and storage [12]. This becomes even worse on lattices, since the resulting ciphertext should include a verification key of the (one-time) signature, which, to the best of our knowledge, has at least one matrix [43]. In contrast, BCHK-MAC [14] makes use of message authentication codes (MAC) and commitments, and thus can reduce the extra overheads, e.g., it only adds a MAC tag and a commitment to the resulting ciphertext.

Given the difficulties in adapting traditional techniques to the lattice setting, and the insufficiencies of existing CCA2-secure PKEs from lattices, it is natural to ask: *Can we directly construct a standard model CCA2-secure PKE from lattices (possibly by carefully exploiting the rich algebraic properties of lattices), such that it has better performance than those following the generic approaches?*

1.1 Our Results

In this paper, we construct a standard model CCA2-secure PKE from lattices, which does not follow the generic approaches mentioned above. Technically, our PKE is obtained by carefully combining a different message encoding with several nice algebraic properties of the tag-based lattice trapdoor and the learning with errors (LWE) problem (e.g., unique witness and additive homomorphism). Unlike previous LWE-based PKEs which typically encode the message into the LWE's "error term", we encode the message into the most significant bits of the LWE's "secret term", which not only provides a better way to control the error size in the decryption, but also allows us to directly achieve CCA2-security. Compared to the best known standard model CCA1-secure PKE [45], our CCA2-secure PKE reduces the noise parameter for encryption roughly by a factor of 3, and thus improves the efficiency in both computation and storage. In the supplemental

Table 1. A concrete comparison of the ciphertext size of our CCA2-secure PKE and the CCA1-secure one in [45] for encrypting a 256-bit (resp., 512-bit) message at 128-bit (resp., 256-bit) security.

Schemes	LWE Parameter $(n, m, q, \alpha q)$	Ciphertext Sizes	Decryption Error	Security Strength
MP12 [45]	$(450, 11905, 3^{10}, 2.45)$	24.74 KB (20.89 KB†)	2^{-88}	2^{128}
	$(660, 19041, 3^{11}, 5.2)$	44.19 KB (37.10 KB†)	2^{-115}	2^{257}
This paper	$(450, 10740, 3^9, 1.5)$	**13.80 KB** (33-44% ↓)	$\mathbf{2^{-100}}$	$\mathbf{2^{131}}$
	$(660, 17333, 3^{10}, 3.1)$	**23.47 KB** (36-46% ↓)	$\mathbf{2^{-138}}$	$\mathbf{2^{260}}$

material, we also extend our techniques to a special type of rings and obtain an efficient CCA2-secure PKE from ring-LWE in the standard model.

In Table 1, we give a concrete comparison of the ciphertext size of our CCA2-secure PKE and the best known standard model CCA1-secure PKE from lattices [45] (note that the authors of [45] only mentioned to achieve CCA2-security by using generic approaches such as the BCHK transform [12]). For better efficiency, we set q as a power of 3 for both schemes. The LWE parameter $(n, m, q, \alpha q)$ was chosen by taking account of the probability of decryption error and the security strength, where n is the LWE dimension, m is the number of LWE samples, q is the LWE modulus and αq is the LWE Gaussian parameter. Since the noise parameter of our encryption algorithm is equal to the underlying LWE Gaussian parameter αq and that of their encryption algorithm is about 3 times larger than αq (as we will discuss later), we have to choose different parameter sets to roughly achieve the same degree of correctness and security. The concrete decryption error and security strength for the parameter sets given in Table 1 is estimated by using a Python script and the online LWE estimator [4], respectively. The ciphertext sizes at rows 1 and 3 (resp., rows 2 and 4) correspond to encrypting a 256-bit message at 128-bit security (resp., a 512-bit message at 256-bit security). We also apply known ciphertext compressing techniques to the PKE in [45], and estimate the final ciphertext sizes (marked by '$\dagger$').[4] Since the key sizes and the computational costs of both schemes are essentially dominated by the LWE parameter $(n, m, q, \alpha q)$, our PKE also has advantages in these two aspects due to the smaller m and αq. For example, the public key size of the PKE in [45] for 128-bit security is about 3.86 MB, while ours for 131-bit security is about 3.26 MB (i.e., a 15% reduction). In all, our CCA2-secure PKE is more efficient than the CCA1-secure PKE in [45].

1.2 Overview of Techniques

As the lattice-based CCA2-secure PKEs in the standard model [48,1,45,68], our scheme also relies on the lattice trapdoor technique, which dates back to the seminal work of Ajtai [3]. In 2008, Gentry et al. [31] first showed how to combine the lattice trapdoor technique and the LWE problem in designing encryption schemes. Technically, they [31] introduced the dual variant of the first LWE-based encryption due to Regev [55], which is nicely compatible with the trapdoor technique. Briefly, the public key of the dual encryption consists of two matrices $\mathsf{pk} = (\mathbf{A}, \mathbf{U}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{n \times \ell}$. In order to encrypt a message $\mu \in \{0,1\}^\ell$, the encryption algorithm randomly chooses $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e}_1 \xleftarrow{\$} D_{\mathbb{Z}^m, \alpha q}, \mathbf{e}_2 \xleftarrow{\$} D_{\mathbb{Z}^\ell, \alpha q}$, and computes the ciphertext $C = (\mathbf{c}_1, \mathbf{c}_2)$, where $D_{\mathbb{Z}^m, \alpha q}$ is the Gaussian distribution with parameter αq (and we call αq the noise parameter for encryption), and

$$\mathbf{c}_1 = \mathbf{A}^T \mathbf{s} + \mathbf{e}_1 \in \mathbb{Z}_q^m, \qquad \mathbf{c}_2 = \mathbf{U}^T \mathbf{s} + \mathbf{e}_2 + \mu \cdot \frac{q}{2} \in \mathbb{Z}_q^\ell.$$

As Regev's LWE-based PKE [55], the message $\mu \in \{0,1\}^\ell$ is encoded into the most significant bits of the LWE's "error term". In this case, the secret key can either be a small norm matrix $\mathbf{E} \in \mathbb{Z}^{m \times \ell}$ satisfying $\mathbf{AE} = \mathbf{U}$, or a trapdoor of the matrix $\mathbf{A}$ which can be used to extract a required matrix $\mathbf{E}$, so that one can

---

[4] Note that [45] actually did not give a concrete choice of parameters, and did not consider any ciphertext compressing technique.

recover the message from the most significant bits of the noise vector $\mathbf{c}_2 - \mathbf{E}^T\mathbf{c}_1 = \mu \cdot \frac{q}{2} + \mathbf{e}_2 - \mathbf{E}^T\mathbf{e}_1$ as long as the l_∞ norm $\|\mathbf{e}_2 - \mathbf{E}^T\mathbf{e}_1\|_\infty < q/4$.

The dual encryption [31] was latter employed by Peikert [48] to construct a standard model CCA2-secure PKE from lattices with similar techniques in spirit to the ones in[52,57], and by Agrawal et al. [1] to construct an efficient IBE which can be transformed into a CCA2-secure by using the generic approaches such as the BCHK transform [12]. Specifically, Agrawal et al. [1] proved that the matrix $(\mathbf{A}\|\mathbf{AR}+\mathbf{CB}) \in \mathbb{Z}_q^{n \times (m_1+m_2)}$ is a trapdoor matrix if 1) $\mathbf{A} \in \mathbb{Z}_q^{n \times m_1}$ is a trapdoor matrix, or 2) $\mathbf{B} \in \mathbb{Z}_q^{n \times m_2}$ is a trapdoor matrix, $\mathbf{R} \in \mathbb{Z}^{m_1 \times m_2}$ is a small norm matrix and $\mathbf{C} \in \mathbb{Z}_q^{n \times n}$ is invertible. By combining this technique with a full-rank difference (FRD) encoding $\mathsf{FRD} : \{0,1\}^\kappa \to \mathbb{Z}_q^{n \times n}$, Agrawal et al. [1] obtained an efficient IBE secure against selective identity and chosen ciphertext attacks, where the public key has four matrices $\mathsf{pk} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{U})$, and the ciphertext $C = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ under a user identity $id \in \{0,1\}^\kappa$ has three vectors:

$$\mathbf{c}_1 = \mathbf{A}_1^T\mathbf{s} + \mathbf{e}_1 \in \mathbb{Z}_q^{m_1}, \qquad \mathbf{c}_2 = (\mathbf{A}_2 + \mathsf{FRD}(id)\mathbf{B})^T\mathbf{s} + \mathbf{R}^T\mathbf{e}_1 \in \mathbb{Z}_q^{m_2},$$
$$\mathbf{c}_3 = \mathbf{U}^T\mathbf{s} + \mathbf{e}_2 + \mu \cdot \frac{q}{2} \in \mathbb{Z}_q^\ell,$$

where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e}_1 \xleftarrow{\$} D_{\mathbb{Z}^{m_1}, \alpha q}, \mathbf{R} \xleftarrow{\$} D_{\mathbb{Z}^{m_1 \times m_2}, \omega(\sqrt{\log n})}$ and $\mathbf{e}_2 \xleftarrow{\$} D_{\mathbb{Z}^\ell, \alpha q}$. The secret key is a trapdoor of the matrix $\mathbf{A}_1$, which can used to extract a small norm matrix $\mathbf{E}$ satisfying $\mathbf{A}_{id}\mathbf{E} = (\mathbf{A}_1\|\mathbf{A}_2 + \mathsf{FRD}(id)\mathbf{B})\mathbf{E} = \mathbf{U}$. In the security proof, the matrix $\mathbf{A}_1$ is uniformly chosen at random, while the matrix $\mathbf{A}_2$ is set to be $\mathbf{A}_2 = \mathbf{A}_1\mathbf{R} - \mathsf{FRD}(id^*)\mathbf{B}$ for a small norm matrix $\mathbf{R} \xleftarrow{\$} D_{\mathbb{Z}^{m_1 \times m_2}, \omega(\sqrt{\log n})}$ and a challenge identity $id^* \xleftarrow{\$} \{0,1\}^\kappa$. By doing this, on the first hand, we have that $\mathbf{A}_{id} = (\mathbf{A}_1\|\mathbf{A}_2 + \mathsf{FRD}(id)\mathbf{B})$ for any $id \neq id^*$ is trapdoor matrix, and thus any ciphertext associated with identity $id \neq id^*$ can be decrypted by using $\mathbf{R}$ and the trapdoor of $\mathbf{B}$. On the other hand, we have that $\mathbf{A}_{id^*} = (\mathbf{A}_1\|\mathbf{A}_1\mathbf{R})$ and the distribution of $\mathbf{c}_2^*$ of the challenge ciphertext $C^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{c}_3^*)$ is essentially statistically close to the distribution of $\mathbf{R}^T\mathbf{c}_1^*$. Thus, in the security proof one can safely replace $\mathbf{c}_2^*$ with $\mathbf{R}^T\mathbf{c}_1^*$, and base the security of the IBE scheme on the hardness of the LWE instances $(\mathbf{A}_1, \mathbf{c}_1^* = \mathbf{A}_1^T\mathbf{s}^* + \mathbf{e}_1^*)$ and $(\mathbf{U}, \mathbf{U}^T\mathbf{s}^* + \mathbf{e}_2^*)$.

Let $\mathbf{G} \in \mathbb{Z}_q^{n \times nk}$ be the public known primitive matrix [45], and $\mathbf{S} \in \mathbb{Z}_q^{nk \times nk}$ be any basis of the lattice $\mathbf{\Lambda}_q(\mathbf{G}) = \{\mathbf{y} \in \mathbb{Z}_q^{nk} : \mathbf{y} = \mathbf{G}^T\mathbf{x} \mod q, \mathbf{x} \in \mathbb{Z}_q^n\}$, where $k = \lceil \log_2 q \rceil$. By improving the tag-based trapdoor technique in [1] with the public known matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times nk}$ and using a new message encoding, Micciancio and Peikert [45] further improved the IBE scheme in [1] and obtained the best known CCA1-secure (tag-based) PKE which saves two matrices in the public key and a vector in the ciphertext (note that an IBE can be naturally treated as a tag-based PKE). Formally, the public key of the PKE in [45] only has two random matrices $\mathsf{pk} = (\mathbf{A}_1, \mathbf{A}_2) \in \mathbb{Z}_q^{n \times m_1} \times \mathbb{Z}_q^{n \times m_2}$, and for a message $\mu \in \{0,1\}^{nk}$, the ciphertext $C = (tag, \mathbf{c}_1, \mathbf{c}_2)$ consists of two vectors:

$$\mathbf{c}_1 = 2(\mathbf{A}_1^T\mathbf{s} \mod q) + \mathbf{e}_1 \mod 2q \in \mathbb{Z}_{2q}^{m_1},$$
$$\mathbf{c}_2 = 2((\mathbf{A}_2 + \mathsf{FRD}(tag)\mathbf{G})^T\mathbf{s} \mod q) + \mathbf{e}_2 + \mathbf{S}\mu \mod 2q \in \mathbb{Z}_{2q}^{m_2},$$

where $tag \xleftarrow{\$} \{0,1\}^\kappa, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e}_1 \xleftarrow{\$} D_{\mathbb{Z}^{m_1}, \alpha' q}$ and $\mathbf{e}_2 \xleftarrow{\$} D_{\mathbb{Z}^{nk}, \sqrt{\|\mathbf{e}_1\|^2 + m_1(\alpha' q)^2} \cdot \omega(\sqrt{\log n})}$. The secret key is a small norm matrix $\mathbf{R}$ satisfying $\mathbf{A}_2 = -\mathbf{A}_1\mathbf{R}$, which can be used to run a trapdoor inversion algorithm to recover the message $\mu \in \{0,1\}^{nk}$ as long as $\|\mathbf{e}_1\|$ and $\|\mathbf{e}_2\|$ are small. Although this encoding allows to save a matrix $\mathbf{U}$ in the public key and a vector in the ciphertext, it only works when the modulus in the encryption is changed from previous q to $2q$, which in turn requires a larger noise parameter $\alpha' q = 3\alpha q$ for lifting the LWE problem with Gaussian parameter αq and modulus q to modulus $2q$ in the security proof (see [45, Theorem 6.3] for details).

We first note that the trapdoor inversion algorithm [45] crucially depends on the size of the LWE's "error term" (i.e., it only works when the error size is small), but does not care about the size of the LWE "secret term" $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$. Furthermore, by the fact that the HNF variant [8] of the LWE problem where the secret $\mathbf{s} \xleftarrow{\$} D_{\mathbb{Z}^n, \alpha q}$ is as hard as the standard LWE problem where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, we can safely replace $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ with $\mathbf{s} \xleftarrow{\$} D_{\mathbb{Z}^n, \alpha q}$ in the encryption. Note that for fixed $\alpha q \in \mathbb{R}$ and randomly chosen $\mathbf{s} \xleftarrow{\$} D_{\mathbb{Z}^n, \alpha q}$, we have that $\|\mathbf{s}\|_\infty \leq B$ holds for some constant $B > 0$ with overwhelming probability by the Gaussian tail inequality.

Thus, for $B \ll q$ (as is usually the case), if we encode the secret $\mathbf{s} = (s_0, \ldots, s_{n-1})^T$ as an element of $\mathbb{Z}_q^n$, the most significant bits of each s_i in the binary representation are "not used" (i.e., those bits are always zeros with overwhelming probability). Our starting point is to encode the message into those "unused" bit-slots. Formally, we introduce a pair of message encode/decode algorithms ($\mathsf{encode}_d : \mathbb{Z}_d^n \to \mathbb{Z}_q^n, \mathsf{decode}_d : \mathbb{Z}_q^n \to \mathbb{Z}_d^n$) for some integer $d \geq 2$ (see Sec. 3.1 for more details) such that for any $\mathbf{v} \in \mathbb{Z}_d^n$ and $0 < B \ll q$, we have $\mathsf{decode}_d(\mathbf{s} + \mathsf{encode}_d(\mathbf{v})) = \mathbf{v}$ as long as $\|\mathbf{s}\|_\infty \leq B$. By doing this, we can keep the advantage of the PKE in [45] (i.e., only having two matrices in the public key $\mathsf{pk} = (\mathbf{A}_1, \mathbf{A}_2)$ and two vectors in the ciphertext $C = (tag, \mathbf{c}_1, \mathbf{c}_2)$) without lifting the modulus in the encryption from q to $2q$, and thus reduce the noise parameter for the encryption from $\alpha' = 3\alpha$ in [45] to the LWE Gaussian parameter α. Specifically, given a message $\mu \in \mathbb{Z}_d^n$, our new encryption algorithm first computes $\hat{\mathbf{s}} = \mathbf{s} + \mathsf{encode}_d(\mu)$, and then computes the ciphertext $C = (tag, \mathbf{c}_1, \mathbf{c}_2)$ as follows:

$$\mathbf{c}_1 = \mathbf{A}_1^T \hat{\mathbf{s}} + \mathbf{e}_1 \in \mathbb{Z}_q^{m_1}, \qquad \mathbf{c}_2 = (\mathbf{A}_2 + \mathsf{FRD}(tag)\mathbf{G})^T \hat{\mathbf{s}} + \mathbf{e}_2 \in \mathbb{Z}_q^{m_2},$$

where $tag \xleftarrow{\$} \{0,1\}^\kappa, \mathbf{s} \xleftarrow{\$} D_{\mathbb{Z}^n, \alpha q}, \mathbf{e}_1 \xleftarrow{\$} D_{\mathbb{Z}^{m_1}, \alpha q}$ and $\mathbf{e}_2 \xleftarrow{\$} D_{\mathbb{Z}^{nk}, \alpha q \sqrt{2m_1} \cdot \omega(\sqrt{\log n})}$. The secret key is still a small norm matrix $\mathbf{R}$ satisfying $\mathbf{A}_2 = -\mathbf{A}_1 \mathbf{R}$. In the security proof, given an LWE tuple $(\mathbf{A}_1, \mathbf{b}) \in \mathbb{Z}_q^{n \times m_1} \times \mathbb{Z}_q^{m_1}$, one can set the public key $\mathsf{pk} = (\mathbf{A}_1, \mathbf{A}_2)$ with $\mathbf{A}_2 = -\mathbf{A}_1\mathbf{R} - \mathsf{FRD}(tag^*)\mathbf{G} \in \mathbb{Z}_q^{n \times nk}$ for $\mathbf{R} \xleftarrow{\$} D_{\mathbb{Z}^{m_1 \times nk}, \omega(\sqrt{\log n})}$ and $tag^* \xleftarrow{\$} \{0,1\}^\kappa$, and simulate the challenge ciphertext $C^* = (tag^*, \mathbf{c}_1^* = \mathbf{b} + \mathbf{A}_1^T \mathsf{encode}_d(\mu_\delta), \mathbf{c}_2^* = -\mathbf{R}^T \mathbf{c}_1 + \mathbf{e}_2')$ by using the additive homomorphism of the LWE problem (i.e., $\mathbf{A}_1^T \hat{\mathbf{s}} + \mathbf{e}_1 = \mathbf{A}_1^T \mathbf{s} + \mathbf{e}_1 + \mathbf{A}_1^T \mathsf{encode}_d(\mu)$) and an independently chosen $\mathbf{e}_2' \xleftarrow{\$} D_{\mathbb{Z}^{nk}, \alpha q \sqrt{m_1} \cdot \omega(\sqrt{\log n})}$, where (μ_0, μ_1) is the challenge message pair and $\delta \xleftarrow{\$} \{0,1\}$. Note that $\mathbf{c}_2^*$ in the challenge ciphertext $C^* = (tag^*, \mathbf{c}_1^*, \mathbf{c}_2^*)$ is directly generated from $\mathbf{c}_1^*$, which relies on the fact that the distribution of $-\mathbf{R}^T \mathbf{e}_1^* + \mathbf{e}_2'$ for $\mathbf{e}_1^* \xleftarrow{\$} D_{\mathbb{Z}^{m_1}, \alpha q}, \mathbf{R} \xleftarrow{\$} D_{\mathbb{Z}^{m_1 \times nk}, \omega(\sqrt{\log n})}$ and $\mathbf{e}_2' \xleftarrow{\$} D_{\mathbb{Z}^{nk}, \alpha q \sqrt{m_1} \cdot \omega(\sqrt{\log n})}$ is statistically close to $\mathbf{e}_2^* \xleftarrow{\$} D_{\mathbb{Z}^{nk}, \alpha q \sqrt{2m_1} \cdot \omega(\sqrt{\log n})}$ (see Lemma 3). In other words, the simulated $C^* = (tag^*, \mathbf{c}_1^*, \mathbf{c}_2^*)$ is statistically close the distribution of the real challenge ciphertext if $(\mathbf{A}_1, \mathbf{b})$ is a real LWE instance, and perfectly hides the message μ_δ if $(\mathbf{A}_1, \mathbf{b})$ is uniformly random.

Our second observation is that under appropriate choice of parameters, the first ciphertext part $\mathbf{c}_1 = \mathbf{A}_1^T \hat{\mathbf{s}} + \mathbf{e}_1 \in \mathbb{Z}_q^{m_1}$ essentially uniquely fixes $\hat{\mathbf{s}}$ (see Lemma 6) and thus the message $\mu \in \mathbb{Z}_d^n$. By applying some necessary checks in the decryption algorithm, we can be assured that there is only a single valid message μ for all ciphertexts $C = (*, \mathbf{c}_1, *)$ which share the same first ciphertext part $\mathbf{c}_1$, i.e., conditioned on that the decryption algorithm does not return a failure symbol $\perp$, its output is essentially independent from the choice of the tag and the second ciphertext part $\mathbf{c}_2$. This feature basically says that the ciphertext is (partially) non-malleable, and our idea is to extend this non-malleability to the whole ciphertext (which is required for achieving CCA2-security). For this, we further modify the encryption algorithm. Formally, given a message $\mu \in \mathbb{F}_{2^\kappa}$, it first chooses $x, y, z \xleftarrow{\$} \mathbb{F}_{2^\kappa}$ from the finite field $\mathbb{F}_{2^\kappa}$, and interpret the bit-concatenation of $(x, y, z) \in (\mathbb{F}_{2^\kappa})^3$ as a vector $\mathbf{v} = x\|y\|z \in \mathbb{Z}_d^n$ (which can always be done if $n \log_2 d \geq 3\kappa$). Then, it computes $\hat{\mathbf{s}} = \mathbf{s} + \mathsf{encode}_d(\mathbf{v})$ and the ciphertext $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ as follows:

$$\begin{aligned} \mathbf{c}_1 &= \mathbf{A}_1^T \hat{\mathbf{s}} + \mathbf{e}_1 \in \mathbb{Z}_q^{m_1}, & tag &= \mathsf{H}(\mathbf{c}_1) \in \mathbb{F}_{2^\kappa} \\ \mathbf{c}_2 &= (\mathbf{A}_2 + \mathsf{FRD}(tag)\mathbf{G})^T \hat{\mathbf{s}} + \mathbf{e}_2 \in \mathbb{Z}_q^{m_2}, & c_3 &= x + \mu \in \mathbb{F}_{2^\kappa}, \\ \tau &= \mathsf{H}(\mathbf{c}_2, c_3) \in \mathbb{F}_{2^\kappa} & c_4 &= \tau y + z \in \mathbb{F}_{2^\kappa}, \end{aligned}$$

where $\mathbf{s} \xleftarrow{\$} D_{\mathbb{Z}^n, \alpha q}, \mathbf{e}_1 \xleftarrow{\$} D_{\mathbb{Z}^{m_1}, \alpha q}, \mathbf{e}_2 \xleftarrow{\$} D_{\mathbb{Z}^{nk}, \alpha q \sqrt{2m_1} \cdot \omega(\sqrt{\log n})}$ and the function $\mathsf{H} : \{0,1\}^* \to \mathbb{F}_{2^\kappa}$ is a collision resistant hash function. Technically, the third part c_3 is a one-time padding encryption which is used to ensure that we can generate $\mathbf{c}_1$ and thus the tag before seeing the challenge message pair in the security proof. The last part c_4 is a Carter-Wegman style one-time MAC, which is used to ensure the integrity of $(\mathbf{c}_2, c_3)$ when $\mathbf{c}_1$ (and thus y, z) is fixed.

At first glance, the above construction seems to raise a circularity issue: the MAC key (y, z) is used to authenticate $\mathbf{c}_2$ which depends on the MAC key. However, the key point is that we will only invoke the security of the MAC to reject the decryption query with $C = (\mathbf{c}_1 = \mathbf{c}_1^*, \mathbf{c}_2, c_3, c_4 \neq c_4^*)$, where $\mathbf{c}_1^*$ uniquely fixes and computationally hides the real MAC key (y^*, z^*) for the challenge ciphertext $C^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*, c_4^*)$. Namely,

we will only rely on the security of the MAC when the key (y^*, z^*) is not determined by the adversary's choice of $\mathbf{c}_2$. Thus, in order to ensure that the adversary cannot find a valid MAC tag $c_4 \neq c_4^*$ under the key (y^*, z^*) and thus cannot make a valid decryption query with $C = (\mathbf{c}_1^*, \mathbf{c}_2, c_3, c_4)$, we only have to show that $(\mathbf{c}_1^*, \mathbf{c}_2^*)$ leaks no information of (y^*, z^*), which in turn can be proven by using the pseudorandomness of the LWE problem and the fact that $\mathbf{c}_2^*$ can be generated by directly using $\mathbf{c}_1^*$.

We now give a sketch of the security proof. Formally, given an LWE tuple $(\mathbf{A}_1, \mathbf{b})$, the reduction first randomly chooses $x^*, y^*, z^* \xleftarrow{\$} \mathbb{F}_{2^\kappa}$, and interpret the bit-concatenation of $(x^*, y^*, z^*) \in (\mathbb{F}_{2^\kappa})^3$ as a vector $\mathbf{v}^* = x^* \| y^* \| z^* \in \mathbb{Z}_d^n$. Then, it computes $\mathbf{c}_1^* = \mathbf{b} + \mathbf{A}_1^T \mathsf{encode}_d(\mathbf{v})$, $\mathbf{A}_2 = -\mathbf{A}_1 \mathbf{R} - \mathsf{FRD}(tag^*)\mathbf{G}$, $\mathbf{c}_2^* = -\mathbf{R}^T \mathbf{c}_1^* + \mathbf{e}_2'$ and sets the public key $\mathsf{pk} = (\mathbf{A}_1, \mathbf{A}_2)$, where $tag^* = \mathsf{H}(\mathbf{c}_1^*)$, $\mathbf{R} \xleftarrow{\$} D_{\mathbb{Z}^{m_1 \times nk}, \omega(\sqrt{\log n})}$, and $\mathbf{e}_2' \xleftarrow{\$} D_{\mathbb{Z}^{nk}, \alpha q \sqrt{m_1} \cdot \omega(\sqrt{\log n})}$. Given a challenge message pair (μ_0, μ_1), the reduction randomly chooses $\delta \xleftarrow{\$} \{0, 1\}$, computes $c_3^* = x^* + \mu_\delta$, $c_4^* = \tau^* y^* + z^* \in \mathbb{F}_{2^\kappa}$, and returns the challenge ciphertext $C = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*, c_4^*)$. By the pseudorandomness of $(\mathbf{A}_1, \mathbf{b})$, we have that (x^*, y^*, z^*) is computationally hidden in $(\mathbf{c}_1^*, \mathbf{c}_2^*)$, which means that μ_δ is computationally hidden in the ciphertext. Thus, it suffices to show that the adversary cannot obtain non-negligible advantage from the decryption query. Note that for a decryption query $C = (\mathbf{c}_1 \neq \mathbf{c}_1^*, \mathbf{c}_2, c_3, c_4)$, we have that $tag = \mathsf{H}(\mathbf{c}_1) \neq \mathsf{H}(\mathbf{c}_1^*) = tag^*$ holds by the collision-resistant of H, which means that the ciphertext C can be correctly decrypted by using $\mathbf{R}$. As for a decryption query $C = (\mathbf{c}_1 = \mathbf{c}_1^*, \mathbf{c}_2, c_3, c_4) \neq C^*$, the reduction will directly return $\perp$, since by the unique witness of the LWE problem, such a ciphertext $C \neq C^*$ is valid if and only if $(\mathbf{c}_2, c_3) \neq (\mathbf{c}_2^*, c_3^*)$ and $c_4 = \tau y^* + z^*$, where $\tau = \mathsf{H}(\mathbf{c}_2, c_3)$. By the collision resistant of H, we have that $\tau \neq \tau^* = \mathsf{H}(\mathbf{c}_2^*, c_3)$ and $c_4 \neq c_4^*$ hold. In other words, if the adversary can output a valid decryption query $C = (\mathbf{c}_1 = \mathbf{c}_1^*, \mathbf{c}_2, c_3, c_4) \neq C^*$, it must be able to uniquely determine the one-time MAC key (y^*, z^*) (since given $c_4^* = \tau^* y^* + z^*$ and $c_4 = \tau y^* + z^* \neq c_4^*$, one can efficiently recover the pair (y^*, z^*)), which contradicts the facts that (y^*, z^*) are uniformly chosen at random and are computationally hidden in the challenge ciphertext C^*.

Finally, we emphasize that our message encoding is very crucial for our construction because: 1) the trapdoor inversion algorithm will not work if one encodes the message into the most significant bits of the "error term" of $\mathbf{c}_1$ or $\mathbf{c}_2$; 2) it would require a large noise parameter and thus increase the error size in decryption if one encodes the message into the "error term" of $\mathbf{c}_2$ as that in [45]; and 3) most importantly, the above proof of the CCA2-security will not work if one encodes the message into the "error term" of $\mathbf{c}_2$ as that in [45] since we cannot rely on the MAC security to reject a decryption query $C = (\mathbf{c}_1^*, \mathbf{c}_2, c_3, c_4) \neq C^*$ when the MAC key is determined by $\mathbf{c}_2$ which itself is chosen by the adversary (i.e., there is a circularity issue). Besides, as an independent of interest, this message encoding may also be very useful in other applications such as lattice-based IBEs and attribute-based encryptions (ABEs).

1.3 Related Work, and Discussion

Along with the introduction of the LWE problem, Regev [55] proposed the first LWE-based PKE, which can only encrypt a 1-bit message. Later, several works were extended to support longer messages (e.g., [51]). At STOC 2008, Gentry et al. [31] gave a "dual" variant of Regev's scheme, which was used to construct the first identity-based encryption (IBE) from lattices. Lindner and Peikert [42] gave a more compact LWE-based PKE, which almost reduces all the parameters by a factor of $\log q$. At CCS 2016, Bos et al. [15] proposed a practical public key-encapsulation mechanism (KEM). The ring-LWE was considered in [63,44] to construct PKE with small key and ciphertext sizes. Stehlé and Steinfeld [62] gave a variant of the NTRU cryptosystem, which has a security proof based on the ring-LWE assumption. Recently, several practical PKEs/KEMs from ring-LWE were proposed, e.g., NewHope [6,5]. All the above PKEs only have CPA-security, but can be boosted into CCA2-secure ones by applying the Fujisaki-Okamoto transform in the random oracle (RO) model. However, a RO-based solution may not be always satisfiable in the real world [19]. In the post-quantum setting, this becomes more subtle since a classic RO-based scheme may even not be secure against adversaries who can query the RO with quantum state [13]. This is why some NIST PQC submissions also provide security arguments in the quantum RO model [13]. Unfortunately, the PKEs in the quantum RO model typically have large security reduction loss (e.g., at least a quadratic security loss [34]) w.r.t. the underlying hard assumptions [65,59,34], which has become one of the main concerns when estimating the

actual security of a scheme with given concrete parameters. We note that this security reduction loss is solely introduced by the security arguments in the quantum RO model, and is irrelevant to the possibly generic speedup of the quantum algorithm in solving hard problems (namely, if the underlying hard problem also suffers from a generic quadratic speedup, e.g., by applying the Grover algorithm, the resulting scheme will suffer from a quartic security loss in the quantum setting). In contrast, a standard model PKE scheme usually has much tighter security reduction, and is thus relatively more interesting in the post-quantum era.

As an instantiation of the generic framework from lossy trapdoor functions (LTDF), Peikert and Waters [52] gave the first standard-model CCA2-secure PKE from LWE. The LTDF techniques were later extended to construct several standard model CCA2-secure PKE from lattices [48,68,64,41]. However, they are relatively inefficient (e.g., having large public-key and ciphertext sizes) due to the use of Dolev-Dwork-Naor like technique [26,28] and signatures. By applying the BCHK transform [12], one can obtain CCA2-secure PKEs from IBEs or TBEs on lattices in the standard model [1,2,21,69,71,70,27]. In this setting, Micciancio and Peikert [45] improved the standard model IBE in [1], and presented the best known CCA1-secure PKE from lattices. Unlike many existing LWE-based PKEs which encode the messages into the "error term", we encode the messages into the most significant bits of the "secret term", which is very crucial to our CCA2-secure PKE construction, and might be of independent interest. We also note that the recent independent work [59] constructed a deterministic PKE by simply encoding the message as the "secret term" and the "error term" of the LWE problem, which is very different from ours from the perspective of both techniques and functionalities.

2 Preliminaries

2.1 Notation

Denote the natural logarithm (resp., the logarithm with base b) as $\log$ (resp., $\log_b$). The standard notations O, ω are used to classify the growth of functions. A function $f(n)$ is negligible in n if for every positive c, we have $f(n) < n^{-c}$ for sufficiently large n. By $\mathrm{negl}(n)$ we denote an arbitrary negligible function. A probability is said to be overwhelming if it is $1 - \mathrm{negl}(n)$. The notation $\xleftarrow{\$}$ denotes randomly choosing elements from a distribution (or the uniform distribution over a finite set). By $x \backsim D$ we mean the random variable x follows a distribution D.

Denote $\mathbb{R}$ (resp., $\mathbb{Z}$) as the set of real numbers (resp., integers). Vectors are column vectors and denoted by bold lower-case letters (e.g., $\mathbf{v}$), and $\mathbf{v}^T$ denote the transpose of $\mathbf{v}$. Matrices are the sets of column vectors and denoted by bold capital letters (e.g., $\mathbf{X}$). The concatenation of a matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ followed by another matrix $\mathbf{Y} \in \mathbb{R}^{n \times m'}$ is denoted as $(\mathbf{X} \| \mathbf{Y}) \in \mathbb{R}^{n \times (m+m')}$. By $\| \cdot \|$ and $\| \cdot \|_\infty$ we denote the ℓ_2 and ℓ_∞ norm, respectively. The largest singular value of $\mathbf{X}$ is $s_1(\mathbf{X}) = \max_{\mathbf{u}} \|\mathbf{X}\mathbf{u}\|$, where the maximum is taken over all unit vector $\mathbf{u}$.

2.2 Public-Key Encryption

A public-key encryption (PKE) Π with message space $\mathcal{P}$ consists of three probabilistic polynomial time (PPT) algorithms (KeyGen, Enc, Dec):

- KeyGen(1^κ) is a PPT algorithm that takes a security parameter κ as input, and outputs a pair of public and secret keys (pk,sk).
- Enc(pk, μ) is a PPT algorithm that encrypts a message $\mu \in \mathcal{P}$ under the public key pk and outputs the corresponding ciphertext C.
- Dec(sk, C) is an efficient deterministic algorithm that decrypts a ciphertext C using the secret key sk and outputs a message μ (or a symbol $\perp$).

We say that a PKE scheme $\Pi = $ (KeyGen,Enc,Dec) is correct, if for any $\mu \in \mathcal{P}$, $(pk, sk) \leftarrow$ KeyGen(1^κ) and $C \leftarrow$ Enc(pk, μ), the probability that Dec(sk, C) $\neq \mu$ is negligible over the random coins used in both KeyGen and Enc. The de facto standard security notion for PKE is (adaptively) chosen-ciphertext security, which is modeled by a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

KeyGen. The challenger $\mathcal{C}$ first computes $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\kappa)$. Then, it gives the public key pk to the adversary $\mathcal{A}$, and keeps sk secret.

Phase 1. The adversary $\mathcal{A}$ is allowed to make any polynomial number of decryption queries by using any (different) ciphertext C of his choice. The challenger $\mathcal{C}$ computes $\mu \leftarrow \mathsf{Dec}(sk, C)$, and returns μ to $\mathcal{A}$.

Challenge. The adversary $\mathcal{A}$ outputs two equal-length messages (μ_0, μ_1). The challenger $\mathcal{C}$ chooses a bit $\delta^* \xleftarrow{\$} \{0, 1\}$, and computes $C^* \leftarrow \mathsf{Enc}(pk, \mu_{\delta^*})$. Finally, it returns the challenge ciphertext C^* to $\mathcal{A}$.

Phase 2. The adversary is allowed to make more decryption queries with any ciphertext $C \neq C^*$. The challenger $\mathcal{C}$ responds as in Phase 1.

Guess. Finally, $\mathcal{A}$ outputs a guess $\delta \in \{0, 1\}$. If $\delta = \delta^*$, the challenger $\mathcal{C}$ outputs 1, else outputs 0.

Definition 1 (CCA2-Security). *We say that a PKE scheme Π is* CCA2-secure *if for any PPT adversary $\mathcal{A}$, its advantage*

$$\mathrm{Adv}_{\Pi, \mathcal{A}}^{\text{ind-cca2}}(\kappa) = |\Pr[\delta = \delta^*] - \frac{1}{2}|$$

in the above game is negligible in security parameter κ.

The CPA-security and CCA1-security can be defined via modified games. Concretely, the CPA-security game does not allow the adversary to make any decryption queries, while the CCA1-security game only allows the adversary to make decryption queries before the challenge phase.

2.3 Gaussian, Learning with Errors and Trapdoors

Gaussian. The Gaussian function $\rho_{s,\mathbf{c}}(\mathbf{x})$ over $\mathbb{R}^m$ centered at $\mathbf{c} \in \mathbb{R}^m$ with parameter $s > 0$ is defined as $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi \|\mathbf{x} - \mathbf{c}\|^2 / s^2)$. For lattice $\mathbf{\Lambda} \subseteq \mathbb{R}^m$, let $\rho_{s,\mathbf{c}}(\mathbf{\Lambda}) = \sum_{\mathbf{x} \in \mathbf{\Lambda}} \rho_{s,\mathbf{c}}(\mathbf{x})$, and define the discrete Gaussian distribution over $\mathbf{\Lambda}$ as $D_{\mathbf{\Lambda}, s, \mathbf{c}}(\mathbf{y}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{y})}{\rho_{s,\mathbf{c}}(\mathbf{\Lambda})}$, where $\mathbf{y} \in \mathbf{\Lambda}$. We omit the subscript $\mathbf{c}$ in the above notations if $\mathbf{c} = \mathbf{0}$.

Lemma 1 ([10,42]). *For any real $s, t > 0$, $c \geq 1$, $C = c \cdot \exp(\frac{1-c^2}{2}) < 1$, integer $m > 0$, and any $\mathbf{y} \in \mathbb{R}^m$ we have the followings hold:*

- $\Pr_{\mathbf{x} \xleftarrow{\$} D_{\mathbb{Z}^m, s}}[\|\mathbf{x}\|_\infty > t \cdot s] \leq 2e^{-\pi t^2}$
- $\Pr_{\mathbf{x} \xleftarrow{\$} D_{\mathbb{Z}^m, s}}[\|\mathbf{x}\| > c \cdot \frac{1}{\sqrt{2\pi}} \cdot s\sqrt{m}] \leq C^m$
- $\Pr_{\mathbf{x} \xleftarrow{\$} D_{\mathbb{Z}^m, s}}[|\langle \mathbf{x}, \mathbf{y} \rangle| > t \cdot s\|\mathbf{y}\|] \leq 2e^{-\pi t^2}$

Lemma 2 ([45]). *Let integer $n > 0$, and q a power of some prime $p \geq 2$. Let integer $m \geq n \log_2 q + \omega(\log n)$. Then, for any $\ell = \mathrm{poly}(n)$ and real $r \geq \omega(\sqrt{\log n})$, the distribution $(\mathbf{A}, \mathbf{AR})$ is statistically close to uniform over $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{n \times \ell}$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $\mathbf{R} \xleftarrow{\$} (D_{\mathbb{Z}^m, r})^\ell$.*

The following lemma is implicit in the proof of [45, Theorem 6.3], which can be proven by combining [49, Theorem 3.1] and [55, Corollary 3.10].

Lemma 3 ([45]). *Let $r \geq \omega(\sqrt{\log n})$. Then, for any vectors $\mathbf{v} \in \mathbb{Z}^m$ and $\mathbf{c} \in \mathbb{Z}^m$, any matrix $\mathbf{r} \xleftarrow{\$} D_{\mathbb{Z}^m, r, \mathbf{c}}$, and $e \xleftarrow{\$} D_{\mathbb{Z}, \alpha q r \cdot \sqrt{m}}$, the distribution $\mathbf{r}^T \mathbf{v} + e$ is statistically close to $D_{\mathbb{Z}, s}$, where $s = r \cdot \sqrt{\|\mathbf{v}\|^2 + m(\alpha q)^2}$.*

Following [45,29], we say that a random variable X over $\mathbb{R}$ is subgaussian with parameter s if for all $t \in \mathbb{R}$, the (scaled) moment-generating function satisfies $\mathbb{E}(\exp(2\pi t X)) \leq \exp(\pi s^2 t^2)$. For any lattice $\mathbf{\Lambda} \subset \mathbb{R}^m$ and $s > 0$, $D_{\mathbf{\Lambda}, s}$ is subgaussian with parameter s. Besides, any B-bounded symmetric random variable X (i.e., $|X| \leq B$) is subgaussian with parameter $B\sqrt{2\pi}$ [45]. For random subgaussian matrix, we have the following result from the non-asymptotic theory of random matrices [66].

Lemma 4. *Let $\mathbf{X} \in \mathbb{R}^{n \times m}$ be a random subgaussian matrix with parameter s. There exists a universal constant $C \approx 1/\sqrt{2\pi}$ such that for any $t \geq 0$, we have $s_1(\mathbf{X}) \leq C \cdot s \cdot (\sqrt{m} + \sqrt{n} + t)$ except with probability at most $2 \exp(-\pi t^2)$.*

Learning with Errors. For any positive integers $n, q \in \mathbb{Z}$, real $\alpha > 0$ and vector $\mathbf{s} \in \mathbb{Z}_q^n$, define $A_{\mathbf{s},\alpha} = \{(\mathbf{a}, \mathbf{a}^T\mathbf{s} + e \mod q) : \mathbf{a} \overset{\$}{\leftarrow} \mathbb{Z}_q^n, e \overset{\$}{\leftarrow} D_{\mathbb{Z},\alpha q}\}$. For m independent samples $(\mathbf{a}_1, y_1), \ldots, (\mathbf{a}_m, y_m)$ from $A_{\mathbf{s},\alpha}$, we denote it in a matrix form $(\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$, where $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_m)$ and $\mathbf{y} = (y_1, \ldots, y_m)^T$. We say that a PPT algorithm solves the $\text{LWE}_{n,m,q,\alpha}$ problem if, for uniformly random $\mathbf{s} \overset{\$}{\leftarrow} \mathbb{Z}_q^n$, given m samples from $A_{\mathbf{s},\alpha}$ it outputs $\mathbf{s}$ with non-negligible probability. The decisional LWE is asked to distinguish $A_{\mathbf{s},\alpha}$ from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ (with only polynomial samples). For certain parameters, the decisional LWE problem is polynomially equivalent to its search version, which in turn is provably at least as hard as quantumly approximating SIVP on n-dimensional lattices to within polynomial factors in the worst case [55,50]. A variant of the LWE problem (known as the *Hermite normal form*) where the secret $\mathbf{s} \in \mathbb{Z}_q^n$ is chosen from the error distribution (i.e., $\mathbf{s} \overset{\$}{\leftarrow} D_{\mathbb{Z}^n,\alpha q}$) is also polynomially equivalent to the standard LWE problem [8].

q-ary Lattices and Trapdoors. Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, define two q-ary lattices

$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m \ \ s.t. \ \mathbf{A}\mathbf{e} = \mathbf{0} \mod q\},$$
$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \ s.t. \ \exists \mathbf{s} \in \mathbb{Z}^n, \ \mathbf{A}^t\mathbf{s} = \mathbf{y} \mod q\}.$$

We have PPT algorithms [3,7,45] to generate an essentially uniform matrix $\mathbf{A}$ together with a trapdoor (or a short basis of $\Lambda_q^\perp(\mathbf{A})$). We will use the trapdoor notion in [45]. Formally, let $\mathbf{G}_b \in \mathbb{Z}_q^{n \times nk}$ be the public primitive matrix with base integer $b \geq 2$ in [45, Theorem 4.1], where $k = \lceil \log_b q \rceil$. We usually omit the subscript b if $b = 2$, and denote $\mathbf{G} = \mathbf{G}_2$ in brief. As shown in [45], there exists a PPT algorithm that inverts $\mathbf{y} = \mathbf{G}_b^T\mathbf{s} + \mathbf{e}$ as long as $\|\mathbf{e}\| < \frac{q}{2\sqrt{b^2+1}}$. Moreover, if $q = b^k$, the algorithm can invert $\mathbf{y} = \mathbf{G}_b^T\mathbf{s} + \mathbf{e}$ if $\|\mathbf{e}\|_\infty < \frac{q}{2b}$. The following lemma is implicit in [45, Theorem. 5.4].

Lemma 5 ([45]). *Let $\mathbf{I}_{nk}$ be the $nk \times nk$ identity matrix. For any matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{R} \in \mathbb{Z}_q^{(m-nk) \times nk}$ and invertible matrix $\mathbf{S} \in \mathbb{Z}_q^{n \times n}$ satisfying $\mathbf{A} \begin{pmatrix} \mathbf{R} \\ \mathbf{I}_{nk} \end{pmatrix} = \mathbf{S}\mathbf{G}_b$, there exists a PPT algorithm $\mathsf{Solve}(\mathbf{A}, \mathbf{R}, \mathbf{y})$ that given any $\mathbf{y} = \mathbf{A}^T\mathbf{s} + \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \in \mathbb{Z}_q^m$ satisfying $\|\mathbf{R}^T\mathbf{e}_1 + \mathbf{e}_2\| < \frac{q}{2\sqrt{b^2+1}}$, outputs $\mathbf{s} \in \mathbb{Z}_q^n$, where $\mathbf{e}_1 \in \mathbb{Z}^{m-nk}$ and $\mathbf{e}_2 \in \mathbb{Z}^{nk}$.*

Moreover, if $q = b^k$, the algorithm $\mathsf{Solve}(\mathbf{A}, \mathbf{R}, \mathbf{y})$ can invert any $\mathbf{y} = \mathbf{A}^T\mathbf{s} + \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \in \mathbb{Z}_q^m$ satisfying $\|\mathbf{R}^T\mathbf{e}_1 + \mathbf{e}_2\|_\infty < \frac{q}{2b}$.

We also need the following useful lemma, which is important for our CCA2-secure PKE construction, and may be of independent interest.

Lemma 6 (Unique Witness). *Let $n, k > 0$ be integers. Let $q = p^k$ for some prime $p \geq 2$, and let $m \geq n\log_2 q + \omega(\log n)$. Then, for all but a negligible fraction of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and for any $\mathbf{u} \in \mathbb{Z}_q^m$, there exists at most one pair $(\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^n \times \mathbb{Z}^m$ such that $\|\mathbf{e}\|_\infty < q/8$ and $\mathbf{u} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$.*

Proof. The proof is adapted from [31, Lemma 5.3]. For any $\mathbf{u} \in \mathbb{Z}_q^m$, we assume that there exist two tuples $(\mathbf{s}, \mathbf{e}) \neq (\mathbf{s}', \mathbf{e}') \in \mathbb{Z}_q^n \times \mathbb{Z}^m$, such that $\|\mathbf{e}\|_\infty, \|\mathbf{e}'\|_\infty < q/8$ and $\mathbf{u} = \mathbf{A}^T\mathbf{s} + \mathbf{e} = \mathbf{A}^T\mathbf{s}' + \mathbf{e}'$. Letting $\tilde{\mathbf{s}} = \mathbf{s} - \mathbf{s}'$ and $\tilde{\mathbf{e}} = \mathbf{e}' - \mathbf{e}$, we have that $\mathbf{A}^T\tilde{\mathbf{s}} = \tilde{\mathbf{e}}$ for some $\tilde{\mathbf{s}} \neq \mathbf{0}$ and $\|\tilde{\mathbf{e}}\|_\infty < q/4$. Now, it suffices to show that for all but an at most $2^{-\omega(\log n)} = \text{negl}(n)$ fraction of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the vector $\mathbf{A}^T\tilde{\mathbf{s}}$ has norm $\|\mathbf{A}^T\tilde{\mathbf{s}}\|_\infty \geq q/4$ for any $\tilde{\mathbf{s}} \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\}$.

Formally, consider the open ℓ_∞ "cube" $\mathcal{V}$ of radius $q/4$ (i.e., each edge has length $q/2$). Denote $(\mathbb{Z}_q^n)^* \subseteq \mathbb{Z}_q^n$ as the set of vectors such that each vector has at least one coordinate which is invertible in $\mathbb{Z}_q$. For any fixed nonzero $\tilde{\mathbf{s}} \in \mathbb{Z}_q^n$, we can write $\tilde{\mathbf{s}} = p^{k'}\tilde{\mathbf{s}}'$ for some integer $k' \in \{0, \ldots, k-1\}$ and $\tilde{\mathbf{s}}' \in (\mathbb{Z}_q^n)^*$. Then, for a uniformly random choice of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we have that $\mathbf{A}^T\tilde{\mathbf{s}}'$ is uniformly over $\mathbb{Z}_q^m$, and that $\mathbf{A}^T\tilde{\mathbf{s}} = p^{k'}\mathbf{A}^T\tilde{\mathbf{s}}'$ is uniformly over $\mathbb{Z}_q^m \cap p^{k'}\mathbb{Z}^m$. Denote $S_{k'} = \mathcal{V} \cap p^{k'}\mathbb{Z}^m$, which contains at most $(p^{k-k'}/2)^m$

points. Thus, over the uniformly random choice of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the probability that $\mathbf{A}^T \tilde{\mathbf{s}} \in S_{k'}$ is at most $(p^{k-k'}/2)^m / p^{(k-k')m} \leq 2^{-m}$. Taking a union bound over all nonzero $\tilde{\mathbf{s}} \in \mathbb{Z}_q^n$, the probability that $\mathbf{A}^T \tilde{\mathbf{s}} \in S_0$ is at most $2^{-\omega(\log n)}$ (note that $S_{k-1} \subset \cdots \subset S_0 = \mathcal{V} \cap \mathbb{Z}^m$ by definition). Since S_0 contains all integer vectors with ℓ_∞ norm $< q/4$, we have that for all but an at most $2^{-\omega(\log n)}$ fraction of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and for any non-zero $\tilde{\mathbf{s}} \in \mathbb{Z}_q^n$, the vector $\mathbf{A}^T \tilde{\mathbf{s}}$ has norm $\|\mathbf{A}^T \tilde{\mathbf{s}}\|_\infty \geq q/4$. $\qquad \square$

3 CCA2-Secure PKE from Lattices

In this section, we first introduce some ingredients for our construction.

3.1 Some Ingredients

Collision-resistant hash function. We say that $\mathsf{H} : X \to Y$ is a collision resistant hash (CRH) if given a security parameter κ and a description of H as inputs, no PPT algorithm $\mathcal{F}$ can find two elements $x_1 \neq x_2 \in X$ such that $\mathsf{H}(x_1) = \mathsf{H}(x_2)$ holds except with negligible probability, where the probability is over the random coins used by $\mathcal{F}$. Namely, if $\mathsf{H} : X \to Y$ is a CRH, we have that

$$\Pr[(x_1, x_2) \leftarrow \mathcal{F}(1^\kappa, \mathsf{H}) : x_1 \neq x_2 \wedge \mathsf{H}(x_1) = \mathsf{H}(x_2)] \leq \mathrm{negl}(\kappa)$$

holds. Note that CRH exists under the LWE assumption [52], and it suffices to use the standard SHA3 in practice.

Full-rank difference encoding. Let κ be the security parameter. We say that $\mathsf{FRD} : \{0,1\}^\kappa \to \mathbb{Z}_q^{n \times n}$ is an encoding with full-rank differences (FRD) if the following two conditions hold: 1) for any $\mathbf{u} \neq \mathbf{v}$, the matrix $\mathsf{FRD}(\mathbf{u} - \mathbf{v}) = \mathsf{FRD}(\mathbf{u}) - \mathsf{FRD}(\mathbf{v}) \in \mathbb{Z}_q^{n \times n}$ is invertible over $\mathbb{Z}_q^{n \times n}$; and 2) $\mathsf{FRD}(\cdot)$ is computable in polynomial time in $n \log q$. As shown in [1,22,45], FRD encodings over $\{0,1\}^\kappa$ can be efficiently constructed for any $\kappa \leq n$ and q a power of some prime $p \geq 2$.

Message Encoding. We define a pair of algorithms $(\mathsf{encode}_d, \mathsf{decode}_d)$, which are parameterized by positive integers (n, q, d). Formally, given any $\mathbf{v} \in \mathbb{Z}_d^n$, the algorithm $\mathsf{encode}_d : \mathbb{Z}_d^n \to \mathbb{Z}_q^n$ is defined as $\mathsf{encode}_d(\mathbf{v}) = (v_1 \cdot \lfloor \frac{q}{d} \rfloor, \ldots, v_n \cdot \lfloor \frac{q}{d} \rfloor)$, where $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{Z}_d^n$. For any $\mathbf{u} \in \mathbb{Z}_q^n$, the algorithm $\mathsf{decode}_d : \mathbb{Z}_q^n \to \mathbb{Z}_d^n$ is defined as $\mathsf{decode}_d(\mathbf{u}) = (\lfloor u_1 \cdot \frac{d}{q} \rceil, \ldots, \lfloor u_n \cdot \frac{d}{q} \rceil)$, where $\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{Z}_q^n$.

Lemma 7. *Let n, q be positive integers, and integer $2 \leq d < \sqrt{q}$. Then, for any $\mathbf{v} \in \mathbb{Z}_d^n$, any $\mathbf{e} \in \mathbb{Z}^n$ satisfying $\|\mathbf{e}\|_\infty < \frac{q - (d-1)d}{2d}$, and $\mathbf{w} = \mathsf{encode}_d(\mathbf{v}) + \mathbf{e}$, we have that $\mathbf{v} = \mathsf{decode}_d(\mathbf{w})$ always holds.*

Proof. Since both algorithms simply apply the same operations on their inputs in a coordinate-wise way, it suffices to show that for any $v \in \mathbb{Z}_d$, any $e \in \mathbb{Z}$ satisfying $|e| < \frac{q-(d-1)d}{2d}$ and $w = v \cdot \lfloor \frac{q}{d} \rfloor + e$, we alway have $v = \lfloor w \cdot \frac{d}{q} \rceil$. By definition, we have $w = v \cdot (\frac{q}{d} + x) + e$ holds for some x satisfying $|x| \leq 1/2$. Thus, $w \cdot \frac{d}{q} = v + (vx + e) \cdot \frac{d}{q}$. Since $|(vx + e) \cdot \frac{d}{q}| \leq (|vx| + |e|) \cdot \frac{d}{q} < \left(\frac{d-1}{2} + \frac{q-(d-1)d}{2d} \right) \cdot \frac{d}{q} = 1/2$ holds by assumption, we have $v = \lfloor w \cdot \frac{d}{q} \rceil$. This completes the proof. $\qquad \square$

3.2 The Construction

Let κ be the security parameter. Let $n, \bar{m} > 0$ be integers, and let q be a prime or a power of prime $b \geq 2$. Let $k = \lceil \log_b q \rceil$ and $m = \bar{m} + nk$. Let $(\mathsf{encode}_d, \mathsf{decode}_d)$ be the pair of encode/decode algorithms parameterized by (n, q, d) satisfying that $n \log_2 d \geq 3\kappa$. Let $\mathbb{F}_{2^\kappa}$ be a finite field of order 2^κ. Let $\mathsf{H} : \{0,1\}^* \to \mathbb{F}_{2^\kappa} \backslash \{0\}$ be a collision-resistant hash function (Namely, we assume that the output of H does not contain the zero element in $\mathbb{F}_{2^\kappa}$ for simplicity). Let $\mathsf{FRD} : \mathbb{F}_{2^\kappa} \to \mathbb{Z}_q^{n \times n}$ be an FRD encoding. Our CCA2-secure PKE with parameters $(n, \bar{m}, q, b, d, \alpha)$ is given as follows.

- KeyGen(1^κ): randomly choose $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times \bar{m}}, \mathbf{R} \overset{\$}{\leftarrow} (D_{\mathbb{Z}^{\bar{m}}, \omega(\sqrt{\log n})})^{nk}$, and compute $\mathbf{B} = -\mathbf{AR}$. Return the pair of public and secret keys $(pk, sk) = ((\mathbf{A}, \mathbf{B}), \mathbf{R})$.
- Enc($pk, \mu \in \mathbb{F}_{2^\kappa}$): first randomly choose $\mathbf{s} \overset{\$}{\leftarrow} D_{\mathbb{Z}^n, \alpha q}, \mathbf{e}_1 \overset{\$}{\leftarrow} D_{\mathbb{Z}^{\bar{m}}, \alpha q}, \mathbf{e}_2 \overset{\$}{\leftarrow} D_{\mathbb{Z}^{nk}, \gamma}$, and $x, y, z \overset{\$}{\leftarrow} \mathbb{F}_{2^\kappa}$, where $\gamma = \sqrt{\|\mathbf{e}_1\|^2 + \bar{m}(\alpha q)^2} \cdot \omega(\sqrt{\log n})$. Then, interpret the bit-concatenation of $(x, y, z) \in (\mathbb{F}_{2^\kappa})^3$ as a vector $\mathbf{v} = x\|y\|z \in \mathbb{Z}_d^n$ (which can always be done since $n \log_2 d \geq 3\kappa$), and compute

$$\tilde{\mathbf{s}} = \mathbf{s} + \mathsf{encode}_d(\mathbf{v}), \qquad\qquad \mathbf{c}_1 = \mathbf{A}^T \tilde{\mathbf{s}} + \mathbf{e}_1,$$
$$\mathbf{c}_2 = (\mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b)^T \tilde{\mathbf{s}} + \mathbf{e}_2, \qquad c_3 = x + \mu \in \mathbb{F}_{2^\kappa},$$
$$c_4 = \tau y + z \in \mathbb{F}_{2^\kappa},$$

where $tag = \mathsf{H}(\mathbf{c}_1) \in \mathbb{F}_{2^\kappa}$ and $\tau = \mathsf{H}(\mathbf{c}_2, c_3) \in \mathbb{F}_{2^\kappa}$. Finally, return the ciphertext $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4) \in \mathbb{Z}_q^{m_1} \times \mathbb{Z}_q^{nk} \times \mathbb{F}_{2^\kappa} \times \mathbb{F}_{2^\kappa}$.
- Dec($sk, C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$): first compute $tag = \mathsf{H}(\mathbf{c}_1)$,

$$\mathbf{A}_{tag} = (\mathbf{A}\|\mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b), \text{ and } \mathbf{u} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix}.$$

Then, compute $\tilde{\mathbf{s}} \leftarrow \mathsf{Solve}(\mathbf{A}_{tag}, \mathbf{R}, \mathbf{u})$, $\mathbf{v} = \mathsf{decode}_d(\tilde{\mathbf{s}}) \in \mathbb{Z}_d^n$, and parse $\mathbf{v} = \|x\|y\|z$, where $(x, y, z) \in (\mathbb{F}_{2^\kappa})^3$. Let $\mathbf{e}_1 = \mathbf{c}_1 - \mathbf{A}^T \tilde{\mathbf{s}}$ and $\mathbf{e}_2 = \mathbf{c}_2 - (\mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b)^T \tilde{\mathbf{s}}$. Return $\perp$ if one of the following conditions holds:
  - $\|\mathbf{e}_1\| > \alpha q \sqrt{\bar{m}}$, or
  - $\|\mathbf{e}_2\| > \gamma\sqrt{nk}$ for prime q (or $\|\mathbf{e}_2\|_\infty > \gamma \cdot \omega(\sqrt{\log n})$ for $q = b^k$ a power of prime b), or
  - $c_4 \neq \mathsf{H}(\mathbf{c}_2, c_3)y + z \in \mathbb{F}_{2^\kappa}$.

  Otherwise, return $\mu = c_3 - x \in \mathbb{F}_{2^\kappa}$.

Correctness. Note that $\mathbf{c}_1 = \mathbf{A}^T \tilde{\mathbf{s}} + \mathbf{e}_1$ and $\mathbf{c}_2 = (\mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b)^T \tilde{\mathbf{s}} + \mathbf{e}_2$, the algorithm $\mathsf{Solve}(\mathbf{A}_{tag}, \mathbf{R}, \mathbf{u})$ can invert $\tilde{\mathbf{s}} \in \mathbb{Z}_q^n$ if $\|\mathbf{R}^T\mathbf{e}_1 + \mathbf{e}_2\| < \frac{q}{2\sqrt{b^2+1}}$ by Lemma 5. Because $\tilde{\mathbf{s}} = \mathbf{s} + \mathsf{encode}_d(\mathbf{v})$, one can correctly recover $\mathbf{v} \in \mathbb{Z}_d^n$ if $\|\mathbf{s}\|_\infty < \frac{q-(d-1)d}{2d}$ by Lemma 7. Note that $\mathbf{s} \overset{\$}{\leftarrow} D_{\mathbb{Z}^n, \alpha q}, \mathbf{e}_1 \overset{\$}{\leftarrow} D_{\mathbb{Z}^{\bar{m}}, \alpha q}$, and $\mathbf{e}_2 \overset{\$}{\leftarrow} D_{\mathbb{Z}^{nk}, \gamma}$, we have that $\|\mathbf{s}\|_\infty \leq \alpha q \cdot \omega(\sqrt{\log n}), \|\mathbf{e}_1\| \leq \alpha q \sqrt{\bar{m}}, \|\mathbf{e}_2\|_\infty \leq \gamma \cdot \omega(\sqrt{\log n})$ and $\|\mathbf{e}_2\| \leq \gamma\sqrt{nk}$ hold with overwhelming probability by Lemma 1. Since $\mathbf{R} \overset{\$}{\leftarrow} (D_{\mathbb{Z}^{\bar{m}}, \omega(\sqrt{\log n})})^{nk}$, the inequality $s_1(\mathbf{R}) \leq \sqrt{\bar{m}} \cdot \omega(\sqrt{\log n})$ holds with overwhelming probability by Lemma 4. Since $\gamma = \sqrt{\|\mathbf{e}_1\|^2 + \bar{m}(\alpha q)^2} \cdot \omega(\sqrt{\log n})$, we have $\|\mathbf{R}^T\mathbf{e}_1 + \mathbf{e}_2\| \leq \alpha q \bar{m} \cdot \omega(\sqrt{\log n})$. Besides, we need $\alpha q \geq 2\sqrt{n}$ for the hardness of the LWE problem [55]. We also need Lemma 2 and Lemma 6 in the security proof, which require $\bar{m} \geq (n + 1)\log_2 q + \omega(\log n)$ and $\|\mathbf{e}_1\|_\infty < q/8$.

In all, for the case where $b = 2$ and q is a prime, the decryption algorithm is correct if we set the parameters $\bar{m}, \alpha, q$ such that

$$\bar{m} = (n + 1)\log_2 q + \omega(\log n), \quad 1/\alpha = \bar{m} \cdot \omega(\sqrt{\log n}), \quad \alpha q = 2\sqrt{n}, \tag{1}$$

which means that $m = \bar{m} + nk = \tilde{O}(n), 1/\alpha = \tilde{O}(n)$ and $q = \tilde{O}(n^{1.5})$.

To obtain better efficiency, one can set q as a power of a small prime b (e.g., $b = 3$), which allows us to use the inequality $\|\mathbf{R}^T\mathbf{e}_1 + \mathbf{e}_2\|_\infty \leq \alpha q\sqrt{\bar{m}} \cdot \omega(\sqrt{\log n})^2 < \frac{q}{2b}$ in the correctness analysis. In this case, it suffices to set the parameters $\bar{m}, \alpha, q$ such that

$$\bar{m} = (n + 1)\log_2 q + \omega(\log n), \quad 1/\alpha = \sqrt{\bar{m}} \cdot \omega(\sqrt{\log n})^2, \quad \alpha q = 2\sqrt{n}, \tag{2}$$

which means that $m = \bar{m} + nk = \tilde{O}(n), 1/\alpha = \tilde{O}(\sqrt{n})$ and $q = \tilde{O}(n)$. In both cases, we can set $2 \leq d \leq \tilde{O}(\sqrt{n})$.

As commented in [42,4], the requirement $\alpha q \geq 2\sqrt{n}$ used for the theoretical worst-case reduction [55] is not tight, and it is better to mainly consider concrete hardness against known attacks when choosing actual parameters. For example, one can set $n = 450, \bar{m} = 6690, m = 10740, q = 3^9 \approx 2^{14.27}, \alpha q = 1.5$ to achieve a decryption error rate less than 2^{-100}, and a security level about 131-bit by the online LWE estimator [4]. In this case, the sizes of the public key and the secret key are about $nm\lceil \log_2 q \rceil \approx 8.64$ MB, and $\bar{m}nk(\log_2(\alpha q \cdot \omega(\sqrt{\log n})) + 1) \approx 16.15$ MB, respectively. For 128-bit security, we set $\kappa = 256$, the ciphertext size for encrypting a 256-bit message is $m\lceil \log_2 q \rceil + 512$ bits ≈ 19.73 KB. By using the compressing technique in Section 4, we can reduce the size of the public key, the secret key and the ciphertext to 3.26 MB, 32 bytes and 13.80 KB, respectively.

3.3 The Security

In this section, we show that the above PKE is CCA2-secure. Formally, we have the following theorem.

Theorem 1. *Let positive integers $n, \bar{m}, b, d, q \in \mathbb{Z}$ and real $\alpha \in (0,1)$ satisfy Equation (1) or (2). If $\mathrm{LWE}_{n,\bar{m},q,\alpha}$ is hard and H is a collision-resistant hash function, then the above PKE scheme is CCA2-secure in the standard model.*

Our proof uses a sequence of games $G_1, \ldots, G_{11}$, with G_1 being the real CCA2-security game (where the challenge ciphertext $C^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*, c_4^*)$ is honestly generated by first randomly choosing $\delta^* \xleftarrow{\$} \{0,1\}$ and then encrypting μ_{δ^*}) and G_{11} a random game (where the challenge ciphertext C^* is essentially uniformly random, and thus the adversary's advantage in game G_{11} is negligible). The security is established by showing that G_1 and G_{11} are computationally indistinguishable in the adversary's view. We outline the changes of game G_i with respect to its previous game G_{i-1} in Table 2.

Proof. We now give the formal proof of Theorem 1. Let $\mathcal{A}$ be an adversary which can break the CCA2-security of our PKE with advantage ϵ. Let F_i be the event that $\mathcal{A}$ correctly guesses $\delta = \delta^*$ in game $i \in \{1, \ldots, 11\}$. By definition, the adversary's advantage $\mathrm{Adv}_{\mathcal{PKE},\mathcal{A}}^{\text{ind-cca2}}(\kappa)$ in game i is exactly $|\Pr[F_i] - 1/2|$.

Game G_1 This game is the real security game as defined in Section 2.2. Formally, the challenger $\mathcal{C}$ works as follows:

KeyGen. first randomly choose $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{R} \xleftarrow{\$} (D_{\mathbb{Z}^{\bar{m}},\omega(\sqrt{\log n})})^{nk}$, and compute $\mathbf{B} = -\mathbf{AR}$. Then, return the pair of public key $\mathsf{pk} = (\mathbf{A}, \mathbf{B})$ to the adversary $\mathcal{A}$, and keeps the secret key $\mathbf{R}$ private.

Phase I. Upon receiving a decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$, first compute

$$\mathbf{A}_{tag} = (\mathbf{A} \| \mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b), \text{ and } \mathbf{u} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix},$$

where $tag = \mathsf{H}(\mathbf{c}_1)$. Then, compute $\tilde{\mathbf{s}} \leftarrow \mathsf{Solve}(\mathbf{A}_{tag}, \mathbf{R}, \mathbf{u})$, $\mathbf{v} = \mathsf{decode}_d(\tilde{\mathbf{s}})$, and parse $\mathbf{v} = x \| y \| z$, where $(x, y, z) \in (\mathbb{F}_{2^\kappa})^3$. Let $\mathbf{e}_1 = \mathbf{c}_1 - \mathbf{A}^T \tilde{\mathbf{s}}$ and $\mathbf{e}_2 = \mathbf{c}_2 - (\mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b)^T \tilde{\mathbf{s}}$. Return $\bot$ to the adversary $\mathcal{A}$ if one of the following conditions holds:
  - $\|\mathbf{e}_1\| > \alpha q \sqrt{\bar{m}}$, or
  - $\|\mathbf{e}_2\| > \gamma \sqrt{nk}$ for prime q (or $\|\mathbf{e}_2\|_\infty > \gamma \cdot \omega(\sqrt{\log n})$ for $q = b^k$ a power of prime b), or
  - $c_4 \neq \mathsf{H}(\mathbf{c}_2, c_3) y + z \in \mathbb{F}_{2^\kappa}$.

Otherwise, return $\mu = c_3 - x \in \mathbb{F}_{2^\kappa}$ to the adversary $\mathcal{A}$.

Challenge. Upon receiving two challenge messages $(\mu_0, \mu_1) \in \mathbb{F}_{2^\kappa} \times \mathbb{F}_{2^\kappa}$ from the adversary $\mathcal{A}$, first randomly choose $\delta^* \xleftarrow{\$} \{0,1\}$, $\mathbf{s}^* \xleftarrow{\$} D_{\mathbb{Z}^n,\alpha q}$, $\mathbf{e}_1^* \xleftarrow{\$} D_{\mathbb{Z}^{\bar{m}},\alpha q}$, $\mathbf{e}_2^* \xleftarrow{\$} D_{\mathbb{Z}^{nk},\gamma}$ and $x^*, y^*, z^* \xleftarrow{\$} \mathbb{F}_{2^\kappa}$, where $\gamma = \sqrt{\|\mathbf{e}_1^*\|^2 + \bar{m}(\alpha q)^2} \cdot \omega(\sqrt{\log n})$. Then, interpret the bit-concatenation of $(x^*, y^*, z^*) \in (\mathbb{F}_{2^\kappa})^3$ as a vector $\mathbf{v}^* = x^* \| y^* \| z^* \in \mathbb{Z}_d^n$, and compute

$$\begin{aligned}
\tilde{\mathbf{s}}^* &= \mathbf{s}^* + \mathsf{encode}_d(\mathbf{v}^*), & \mathbf{c}_1^* &= \mathbf{A}^T \tilde{\mathbf{s}}^* + \mathbf{e}_1^*, \\
\mathbf{c}_2^* &= (\mathbf{B} + \mathsf{FRD}(tag^*)\mathbf{G}_b)^T \tilde{\mathbf{s}}^* + \mathbf{e}_2^*, & c_3^* &= x^* + \mu_{\delta^*} \in \mathbb{F}_{2^\kappa}, \\
c_4^* &= \tau^* y^* + z^* \in \mathbb{F}_{2^\kappa}
\end{aligned}$$

where $tag^* = \mathsf{H}(\mathbf{c}_1^*) \in \mathbb{F}_{2^\kappa}$ and $\tau^* = \mathsf{H}(\mathbf{c}_2^*, c_3^*) \in \mathbb{F}_{2^\kappa}$. Finally, return the challenge ciphertext $C^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*, c_4^*)$ to $\mathcal{A}$.

Phase II. Upon receiving a decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$, directly return $\bot$ to the adversary $\mathcal{A}$ if $C = C^*$, otherwise answer this query as in Phase I.

By definition, we have the following lemma.

Lemma 8. $|\Pr[F_1] - 1/2| = \epsilon$.

Table 2. Outline of the game sequences for proving Theorem 1 (where μ_0 and μ_1 are the challenge messages. We use $\overset{c}{\approx}$ and $\overset{s}{\approx}$ to represent the computational indistinguishability and statistical indistinguishability between two games, respectively.)

Games	Changes w.r.t. Previous Game	Note
G_1	Public key: $pk = (\mathbf{A}, \mathbf{B} = -\mathbf{AR})$, Secret key: $sk = \mathbf{R}$, Challenge C^*: $C^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*, c_4^*)$, where $\mathbf{c}_1^* = \mathbf{A}^T \tilde{\mathbf{s}}^* + \mathbf{e}_1^*$, $\mathbf{c}_2^* = (\mathbf{B} + \mathsf{FRD}(tag^*)\mathbf{G}_b)^T \tilde{\mathbf{s}}^* + \mathbf{e}_2^*$, $c_3^* = x^* + \mu_{\delta^*} \in \mathbb{F}_{2^\kappa}$ $c_4^* = \tau^* y^* + z^* \in \mathbb{F}_{2^\kappa}$, for some $\mathbf{s}^* \overset{\$}{\leftarrow} D_{\mathbb{Z}^n, \alpha q}$ $\mathbf{e}_1^* \overset{\$}{\leftarrow} D_{\mathbb{Z}^{\bar{m}}, \alpha q}$ $\mathbf{e}_2^* \overset{\$}{\leftarrow} D_{\mathbb{Z}^{nk}, \sqrt{\|\mathbf{e}_1^*\|^2 + \bar{m}(\alpha q)^2} \cdot \omega(\sqrt{\log n})}$ $x^*, y^*, z^* \overset{\$}{\leftarrow} \mathbb{F}_{2^\kappa}$ $\mathbf{v}^* = x^* \| y^* \| z^* \in \mathbb{Z}_d^n$ $\tilde{\mathbf{s}}^* = \mathbf{s}^* + \mathsf{encode}_d(\mathbf{v}^*)$, $tag^* = \mathsf{H}(\mathbf{c}_1^*)$, $\tau^* = \mathsf{H}(\mathbf{c}_2^*, c_3)$. $\delta^* \overset{\$}{\leftarrow} \{0,1\}$, Dec. query C: run $\mathsf{Dec}(sk, C)$ for any $C \neq C^*$	real game
G_2	generate $(\mathbf{c}_1^*, \mathbf{c}_2^*, x^*, y^*, z^*)$ before giving pk to the adversary (i.e., in the KeyGen phase)	The change is conceptual: $G_2 = G_1$
G_3	immediately return $\bot$ to the decryption query with $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ if $\mathbf{c}_1 \neq \mathbf{c}_1^* \wedge \mathsf{H}(\mathbf{c}_1) = \mathsf{H}(\mathbf{c}_1^*)$	By the collision resistance of H: $G_3 \overset{c}{\approx} G_2$
G_4	immediately return $\bot$ to the decryption query with $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ if $(\mathbf{c}_2, c_3) \neq (\mathbf{c}_2^*, c_3^*) \wedge \mathsf{H}(\mathbf{c}_2, c_3) = \mathsf{H}(\mathbf{c}_2^*, c_3^*)$	By the collision resistance of H: $G_4 \overset{c}{\approx} G_3$
G_5	immediately return $\bot$ to the decryption query with $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ in Phase I if $\mathbf{c}_1 = \mathbf{c}_1^*$	By the high min-entropy of $\mathbf{c}_1^*$: $G_5 \overset{s}{\approx} G_4$
G_6	immediately return $\bot$ to the decryption query with $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ in Phase II if $(\mathbf{c}_1, \mathbf{c}_2, c_3) = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*)$ or $\mathbf{c}_1 = \mathbf{c}_1^* \wedge (\mathbf{c}_2, c_3) \neq (\mathbf{c}_2^*, c_3^*) \wedge c_4 \neq \mathsf{H}(\mathbf{c}_2, c_3)y^* + z^*$	By the unique witness of LWE (i.e., Lemma 6) and the definition of Dec: $G_6 \overset{s}{\approx} G_5$
G_7	immediately return $\bot$ to the decryption query with $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ in Phase II if $\mathbf{c}_1 = \mathbf{c}_1^*$	By the pseudorandomness of LWE and the definition of Dec:[*] $G_7 \overset{c}{\approx} G_6$
G_8	set $pk = (\mathbf{A}, \mathbf{B} = -\mathbf{AR}' - \mathsf{FRD}(tag^*)\mathbf{G}_b)$ and use $\mathbf{R}'$ to answer the decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ if C does not satisfy the "immediate rejection" rules in game G_7 (which means that $tag = \mathsf{H}(\mathbf{c}_1) \neq \mathsf{H}(\mathbf{c}_1^*) = tag^*$)	By the properties of trapdoor generation and inversion algorithms, and the definition of Dec: $G_8 \overset{s}{\approx} G_7$
G_9	use $\mathbf{R}'$ and $\mathbf{c}_1^*$ to generate $\mathbf{c}_2^* = (-\mathbf{R}')^T \mathbf{c}_1^* + \mathbf{e}_2' = (\mathbf{B} + \mathsf{FRD}(tag^*)\mathbf{G}_b)^T \tilde{\mathbf{s}}^* + (-\mathbf{R}')^T \mathbf{e}_1^* + \mathbf{e}_2'$, where $\mathbf{e}_2' \overset{\$}{\leftarrow} D_{\mathbb{Z}^{\bar{n}k}, \alpha q \sqrt{\bar{m}} \cdot \omega(\sqrt{\log n})}$	By Lemma 3 and the definition of Enc: $G_9 \overset{s}{\approx} G_8$
G_{10}	choose $\mathbf{c}_1^* \overset{\$}{\leftarrow} \mathbb{Z}_q^{\bar{m}}$ at random	By the pseudorandomness of LWE: $G_{10} \overset{c}{\approx} G_9$
G_{11}	choose $\mathbf{c}_2^* \overset{\$}{\leftarrow} \mathbb{Z}_q^{nk}$ at random	By Lemma 2: $G_{11} \overset{s}{\approx} G_{10}$

[*] The proof of this claim is relatively involved, and we will use the proof technique of game transitions based on failure events in [61].

Game G_2 This game is similar to game G_1 except that the challenger $\mathcal{C}$ changes the KeyGen and Challenge phases as follows:

KeyGen. first randomly choose $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{R} \xleftarrow{\$} (D_{\mathbb{Z}^{\bar{m}}, \omega(\sqrt{\log n})})^{nk}$, $\mathbf{s}^* \xleftarrow{\$} D_{\mathbb{Z}^n, \alpha q}$, $\mathbf{e}_1^* \xleftarrow{\$} D_{\mathbb{Z}^{\bar{m}}, \alpha q}$, $\mathbf{e}_2^* \xleftarrow{\$} D_{\mathbb{Z}^{nk}, \gamma}$ and $x^*, y^*, z^* \xleftarrow{\$} \mathbb{F}_{2^\kappa}$, where $\gamma = \sqrt{\|\mathbf{e}_1^*\|^2 + \bar{m}(\alpha q)^2} \cdot \omega(\sqrt{\log n})$. Then, interpret the bit-concatenation of $(x^*, y^*, z^*) \in (\mathbb{F}_{2^\kappa})^3$ as a vector $\mathbf{v}^* = x^* \| y^* \| z^* \in \mathbb{Z}_d^n$, and compute

$$\begin{aligned} \tilde{\mathbf{s}}^* &= \mathbf{s}^* + \mathsf{encode}_d(\mathbf{v}^*), & \mathbf{c}_1^* &= \mathbf{A}^T \tilde{\mathbf{s}}^* + \mathbf{e}_1^*, \\ \mathbf{B} &= -\mathbf{A}\mathbf{R} & \mathbf{c}_2^* &= (\mathbf{B} + \mathsf{FRD}(tag^*)\mathbf{G}_b)^T \tilde{\mathbf{s}}^* + \mathbf{e}_2^*, \end{aligned}$$

where $tag^* = \mathsf{H}(\mathbf{c}_1^*)$. Finally, give the public key $pk = (\mathbf{A}, \mathbf{B})$ to the adversary $\mathcal{A}$, keep the secret $sk = \mathbf{R}$ and $(\mathbf{c}_1^*, \mathbf{c}_2^*, x^*, y^*, z^*)$ secret.

Challenge. Upon receiving two challenge messages $(\mu_0, \mu_1) \in \mathbb{F}_{2^\kappa} \times \mathbb{F}_{2^\kappa}$ from the adversary $\mathcal{A}$, first choose a bit $\delta^* \xleftarrow{\$} \{0, 1\}$ and retrieve $(\mathbf{c}_1^*, \mathbf{c}_2^*, x^*, y^*, z^*)$. Then, compute $c_3^* = x^* + \mu_{\delta^*} \in \mathbb{F}_{2^\kappa}$, $c_4^* = \tau^* y^* + z^* \in \mathbb{F}_{2^\kappa}$, where $\tau^* = \mathsf{H}(\mathbf{c}_2^*, c_3^*)$. Finally, return the challenge ciphertext $C^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*, c_4^*)$ to $\mathcal{A}$.

Lemma 9. *Games G_2 and G_1 are identical in the adversary's view. Moreover, $\Pr[F_2] = \Pr[F_1]$.*

Proof. This lemma follows from the fact that $(\mathbf{c}_1^*, \mathbf{c}_2^*, x^*, y^*, z^*)$ is independent from the adversary's choices of the challenge messages, and game G_2 is essentially a conceptual change of game G_1 in the adversary's view. $\qquad \square$

Game G_3 This game is similar to game G_2 except that the challenger $\mathcal{C}$ immediately returns $\bot$ to the decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ from the adversary $\mathcal{A}$ if $\mathbf{c}_1 \neq \mathbf{c}_1^* \wedge \mathsf{H}(\mathbf{c}_1) = \mathsf{H}(\mathbf{c}_1^*)$.

Lemma 10. *If H is a collision-resistant hash function, then games G_3 and G_2 are computationally indistinguishable. Moreover, $|\Pr[F_3] - \Pr[F_2]| \leq \mathrm{negl}(\kappa)$.*

Proof. Let $\mathcal{E}$ be the event that the adversary makes a decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ in Phase I such that $\mathbf{c}_1 \neq \mathbf{c}_1^* \wedge \mathsf{H}(\mathbf{c}_1) = \mathsf{H}(\mathbf{c}_1^*)$. Note that if $\mathcal{E}$ can only happen with negligible probability, then games G_3 and G_2 are computationally indistinguishable in the adversary's view. Now, we show that if there is a PPT adversary $\mathcal{A}$ that makes $\mathcal{E}$ happen with non-negligible probability, there is a PPT adversary $\mathcal{F}$ that finds a collision of H with the same probability by honestly simulating the attack environment for $\mathcal{A}$ as in game G_3. Whenever $\mathcal{A}$ outputs a ciphertext $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ such that $\mathbf{c}_1 \neq \mathbf{c}_1^* \wedge \mathsf{H}(\mathbf{c}_1) = \mathsf{H}(\mathbf{c}_1^*)$ at some time in Phase I, $\mathcal{F}$ returns the pairs $(\mathbf{c}_1, \mathbf{c}_1^*)$ as its own output and aborts. Obviously, the probability that $\mathcal{F}$ succeeds is equal to the probability that $\mathcal{A}$ makes $\mathcal{E}$ happen. Thus, under the assumption that H is collision-resistant, the probability that $\mathcal{E}$ happens is negligible, which completes the proof. $\qquad \square$

Game G_4 This game is similar to game G_3 except that the challenger $\mathcal{C}$ immediately returns $\bot$ to the decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ from the adversary $\mathcal{A}$ if $(\mathbf{c}_2, c_3) \neq (\mathbf{c}_2^*, c_3^*) \wedge \mathsf{H}(\mathbf{c}_2, c_3) = \mathsf{H}(\mathbf{c}_2^*, c_3^*)$.

Lemma 11. *If H is a collision-resistant hash function, then games G_4 and G_3 are computationally indistinguishable. Moreover, $|\Pr[F_4] - \Pr[F_3]| \leq \mathrm{negl}(\kappa)$.*

Proof. The proof is the same to that of Lemma 10, we omit the details. $\qquad \square$

Game G_5 This game is similar to game G_4 except that the challenger $\mathcal{C}$ immediately returns $\bot$ to the decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ from the adversary $\mathcal{A}$ in Phase I if $\mathbf{c}_1 = \mathbf{c}_1^*$.

Lemma 12. *Let positive integers $n, \bar{m}, b, d, q \in \mathbb{Z}$ and real $\alpha \in (0, 1)$ satisfy Equation (1) or (2). Then, games G_5 and G_4 are statistically indistinguishable. Moreover, $|\Pr[F_5] - \Pr[F_4]| \leq \mathrm{negl}(\kappa)$.*

Proof. Let $\mathcal{E}$ be the event that the adversary makes a decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ in Phase I such that $\mathbf{c}_1 = \mathbf{c}_1^*$. Note that if $\mathcal{E}$ does not happen, then games G_5 and G_4 are identical in the adversary's view. Thus, it is enough to show that $\Pr[\mathcal{E}]$ is negligible for any (unbounded) adversary $\mathcal{A}$ making at most a polynomial number of decryption queries in Phase I. Note that in both games G_4 and G_5, the ciphertext part $\mathbf{c}_1^* = \mathbf{A}^T \tilde{\mathbf{s}}^* + \mathbf{e}_1^*$ is always generated by using $\tilde{\mathbf{s}}^* = \mathbf{s}^* + \mathsf{encode}_d(\mathbf{v}^*)$ and $\mathbf{e}_1^* \stackrel{\$}{\leftarrow} D_{\mathbb{Z}^{\bar{m}}, \alpha q}$, where $\mathbf{s}^* \stackrel{\$}{\leftarrow} D_{\mathbb{Z}^n, \alpha q}, x^*, y^*, z^* \stackrel{\$}{\leftarrow} \mathbb{F}_{2^\kappa}$ and $\mathbf{v}^* = x^* \| y^* \| z^* \in \mathbb{Z}_d^n$. By the high min-entropy of the Gaussian distribution, we have that $\mathbf{c}_1^*$ has min-entropy at least κ, where κ is the security parameter. In other words, the probability that for any (unbounded) adversary to output $\mathbf{c}_1 = \mathbf{c}_1^*$ in Phase I (i.e., before seeing $\mathbf{c}_1^*$) is negligible. This means that if $\mathcal{A}$ can make $\mathcal{E}$ happen with non-negligible probability, which completes the proof. $\qquad\square$

Game G_6 This game is similar to game G_5 except that the challenger $\mathcal{C}$ immediately returns $\bot$ to the decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ from the adversary $\mathcal{A}$ in Phase II if $(\mathbf{c}_1, \mathbf{c}_2, c_3) = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*)$ or $\mathbf{c}_1 = \mathbf{c}_1^* \wedge (\mathbf{c}_2, c_3) \neq (\mathbf{c}_2^*, c_3^*) \wedge c_4 \neq \mathsf{H}(\mathbf{c}_2, c_3) y^* + z^*$, where $C^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*, \mathbf{c}_4^*)$ is the challenge ciphertext.

Lemma 13. *Let positive integers $n, \bar{m}, b, d, q \in \mathbb{Z}$ and real $\alpha \in (0, 1)$ satisfy Equation (1) or (2), then games G_6 and G_5 are statistically indistinguishable. Moreover, $|\Pr[F_6] - \Pr[F_5]| \leq \mathsf{negl}(\kappa)$.*

Proof. It suffices to show that the challenger $\mathcal{C}$ in game G_5 will always return $\bot$ to a decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4) \neq C^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*, \mathbf{c}_4^*)$ from the adversary $\mathcal{A}$ in Phase II if $(\mathbf{c}_1, \mathbf{c}_2, c_3) = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*)$ or $\mathbf{c}_1 = \mathbf{c}_1^* \wedge (\mathbf{c}_2, c_3) \neq (\mathbf{c}_2^*, c_3^*) \wedge c_4 \neq \mathsf{H}(\mathbf{c}_2, c_3) y^* + z^*$ in Phase II except with negligible probability. Note that given a decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4) \neq C^*$, the challenger $\mathcal{C}$ in game G_5 will first compute $tag = \mathsf{H}(\mathbf{c}_1)$, and

$$\mathbf{A}_{tag} = (\mathbf{A} \| \mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b), \text{ and } \mathbf{u} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix}.$$

Then, compute $\tilde{\mathbf{s}} \leftarrow \mathsf{Solve}(\mathbf{A}_{tag}, \mathbf{R}, \mathbf{u})$, $\mathbf{v} = \mathsf{decode}_d(\tilde{\mathbf{s}})$, and parse $\mathbf{v} = x \| y \| z$, where $(x, y, z) \in (\mathbb{F}_{2^\kappa})^3$. Let $\mathbf{e}_1 = \mathbf{c}_1 - \mathbf{A}^T \tilde{\mathbf{s}}$ and $\mathbf{e}_2 = \mathbf{c}_2 - (\mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b)^T \tilde{\mathbf{s}}$. Finally, return $\bot$ to the adversary if one of the following conditions holds:

- $\|\mathbf{e}_1\| > \alpha q \sqrt{\bar{m}}$, or
- $\|\mathbf{e}_2\| > \gamma \sqrt{nk}$ for prime q (or $\|\mathbf{e}_2\|_\infty > \gamma \cdot \omega(\sqrt{\log n})$ for $q = b^k$ a power of prime b), or
- $c_4 \neq \mathsf{H}(\mathbf{c}_2, c_3) y + z \in \mathbb{F}_{2^\kappa}$.

Otherwise, return $\mu = c_3 - x \in \mathbb{F}_{2^\kappa}$.

Clearly, the challenger $\mathcal{C}$ in Game G_5 will not return $\bot$ to the decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4) \neq C^*$ only when $\|\mathbf{e}_1\|_\infty \leq \|\mathbf{e}_1\| \leq \alpha q \sqrt{\bar{m}}$ and $c_4 = \mathsf{H}(\mathbf{c}_2, c_3) y + z$. In addition, given $\mathbf{c}_1^* = \mathbf{A}^T \tilde{\mathbf{s}}^* + \mathbf{e}_1^*$ for $\mathbf{e}_1^* \stackrel{\$}{\leftarrow} D_{\mathbb{Z}^{\bar{m}}, \alpha q}$, the challenger $\mathcal{C}$ in Game G_5 will not return $\bot$ to a decryption query $C = (\mathbf{c}_1 = \mathbf{c}_1^*, \mathbf{c}_2, c_3, c_4)$ only if $c_4 = \mathsf{H}(\mathbf{c}_2, c_3) y^* + z^*$ except with negligible probability, since in this case we always have $(\tilde{\mathbf{s}}, \mathbf{e}_1) = (\tilde{\mathbf{s}}^*, \mathbf{e}_1^*)$ with overwhelming probability by the unique witness property in Lemma 6, which in turn implies that $\mathbf{v} = \mathbf{v}^*$ and $(x, y, z) = (x^*, y^*, z^*)$ by the correctness of decode_d. In other words, the challenger $\mathcal{C}$ in Game G_5 will always return $\bot$ to a decryption query $C = (\mathbf{c}_1 = \mathbf{c}_1^*, \mathbf{c}_2, c_3, c_4) \neq C^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*, \mathbf{c}_4^*)$ from the adversary $\mathcal{A}$ in Phase II if $(\mathbf{c}_1, \mathbf{c}_2, c_3) = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*)$ or $\mathbf{c}_1 = \mathbf{c}_1^* \wedge (\mathbf{c}_2, c_3) \neq (\mathbf{c}_2^*, c_3^*) \wedge c_4 \neq \mathsf{H}(\mathbf{c}_2, c_3) y^* + z^*$ holds, except with negligible probability. This completes the proof. $\qquad\square$

Game G_7 This game is similar to game G_6 except that the challenger $\mathcal{C}$ immediately returns $\bot$ to the decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ from the adversary $\mathcal{A}$ in Phase II if $\mathbf{c}_1 = \mathbf{c}_1^*$.

Note that our goal is to show games G_7 and G_6 are computationally indistinguishable under the LWE assumption, but for technical reason it is difficult to do this in game G_7. Fortunately, we can still continue the game sequences by using the proof strategy (i.e., game transitions based on failure events) in [61]. Formally, for $i \in \{6, 7, 8, \ldots, 11\}$, let E_i be the failure event in game G_i that the adversary makes a decryption query with $C = (\mathbf{c}_1^*, \mathbf{c}_2, c_3, c_4)$ such that $\tau = \mathsf{H}(\mathbf{c}_2, c_3) \neq \tau^* \wedge c_4 = \tau y^* + z^*$.

Lemma 14. *If E_7 and E_6 do not happen, then games G_7 and G_6 are identical in the adversary's view. Moreover, $\Pr[F_7|\neg E_7] = \Pr[F_6|\neg E_6]$ and $\Pr[E_7] = \Pr[E_6]$.*

Proof. Let $C^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*, c_4^*)$ be the corresponding challenge ciphertext, where $\tau^* = \mathsf{H}(\mathbf{c}_2^*, c_3^*)$, and $c_4^* = \tau^* y^* + z^*$ for some $y^*, z^* \in \{0,1\}^\kappa$. Note that upon receiving a decryption query with $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ in Phase II, the challenger $\mathcal{C}$ in both games will always return $\perp$ if $(\mathbf{c}_2, c_3) \neq (\mathbf{c}_2^*, c_3^*) \wedge \tau = \mathsf{H}(\mathbf{c}_2, c_3) = \tau^*$. Moreover, the challenger $\mathcal{C}$ in game G_6 will return $\perp$ if $(\mathbf{c}_1, \mathbf{c}_2, c_3) = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*)$ or $\mathbf{c}_1 = \mathbf{c}_1^* \wedge (\mathbf{c}_2, c_3) \neq (\mathbf{c}_2^*, c_3^*) \wedge c_4 \neq \mathsf{H}(\mathbf{c}_2, c_3) y^* + z^*$ holds. In other words, the only difference between games G_7 and G_6 is that the challenger $\mathcal{C}$ in game G_7 also returns $\perp$ to the decryption query $C = (\mathbf{c}_1^*, \mathbf{c}_2, c_3, c_4) \neq C^*$ even if $\tau = \mathsf{H}(\mathbf{c}_2, c_3) \neq \mathsf{H}(\mathbf{c}_2^*, c_3^*) = \tau^* \wedge c_4 = \tau y^* + z^*$. Clearly, if E_7 and E_6 do not happen, then both games are identical in the adversary's view. In particular, the adversary's view in game G_7 before E_7 happens is essentially identical to that in game G_6. Thus, we have $\Pr[F_7|\neg E_7] = \Pr[F_6|\neg E_6]$ and $\Pr[E_7] = \Pr[E_6]$. □

Game G_8 This game is similar to game G_7 except that the challenger $\mathcal{C}$ changes the KeyGen phase and handles the decryption queries as follows:

KeyGen. first randomly choose $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{R}' \overset{\$}{\leftarrow} (D_{\mathbb{Z}^{\bar{m}}, \omega(\sqrt{\log n})})^{nk}$, $\mathbf{s}^* \overset{\$}{\leftarrow} D_{\mathbb{Z}^n, \alpha q}$, $\mathbf{e}_1^* \overset{\$}{\leftarrow} D_{\mathbb{Z}^{\bar{m}}, \alpha q}$, $\mathbf{e}_2^* \overset{\$}{\leftarrow} D_{\mathbb{Z}^{nk}, \gamma}$ and $x^*, y^*, z^* \overset{\$}{\leftarrow} \mathbb{F}_{2^\kappa}$, where $\gamma = \sqrt{\|\mathbf{e}_1^*\|^2 + \bar{m}(\alpha q)^2} \cdot \omega(\sqrt{\log n})$. Then, interpret the bit-concatenation of $(x^*, y^*, z^*) \in (\mathbb{F}_{2^\kappa})^3$ as a vector $\mathbf{v}^* = x^* \| y^* \| z^* \in \mathbb{Z}_d^n$, and compute

$$\tilde{\mathbf{s}}^* = \mathbf{s}^* + \mathsf{encode}_d(\mathbf{v}^*), \qquad \mathbf{c}_1^* = \mathbf{A}^T \tilde{\mathbf{s}}^* + \mathbf{e}_1^*,$$
$$\mathbf{B} = -\mathbf{A}\mathbf{R}' - \mathsf{FRD}(tag^*)\mathbf{G}_b,$$
$$\mathbf{c}_2^* = (\mathbf{B} + \mathsf{FRD}(tag^*)\mathbf{G}_b)^T \tilde{\mathbf{s}}^* + \mathbf{e}_2^* = -(\mathbf{R}')^T \mathbf{A}^T \tilde{\mathbf{s}}^* + \mathbf{e}_2^*,$$

where $tag^* = \mathsf{H}(\mathbf{c}_1^*)$. Finally, give the public key $pk = (\mathbf{A}, \mathbf{B})$ to the adversary $\mathcal{A}$, and keep $(\mathbf{R}', \mathbf{c}_1^*, \mathbf{c}_2^*, x^*, y^*, z^*)$ secret.

Decryption Query. Upon receiving a decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ from the adversary $\mathcal{A}$, return $\perp$ to $\mathcal{A}$ if this query can be immediately responded with $\perp$ using the rules in previous games. Otherwise, first set

$$\mathbf{A}_{tag} = (\mathbf{A} \| \mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b), \text{ and } \mathbf{u} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix},$$

where $tag = \mathsf{H}(\mathbf{c}_1)$. Then, compute $\tilde{\mathbf{s}} \leftarrow \mathsf{Solve}(\mathbf{A}_{tag}, \mathbf{R}', \mathbf{u})$, $\mathbf{v} = \mathsf{decode}_d(\tilde{\mathbf{s}})$, and parse $\mathbf{v} = x \| y \| z$, where $(x, y, z) \in (\mathbb{F}_{2^\kappa})^3$. Let $\mathbf{e}_1 = \mathbf{c}_1 - \mathbf{A}^T \tilde{\mathbf{s}}$ and $\mathbf{e}_2 = \mathbf{c}_2 - (\mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b)^T \tilde{\mathbf{s}}$. Return $\perp$ to the adversary $\mathcal{A}$ if one of the following conditions holds:
- $\|\mathbf{e}_1\| > \alpha q \sqrt{\bar{m}}$, or
- $\|\mathbf{e}_2\| > \gamma \sqrt{nk}$ for prime q (or $\|\mathbf{e}_2\|_\infty > \gamma \cdot \omega(\sqrt{\log n})$ for $q = b^k$ a power of prime b), or
- $c_4 \neq \mathsf{H}(\mathbf{c}_2, c_3) y + z$.

Otherwise, return $\mu = c_3 - x$ to the adversary $\mathcal{A}$.

Lemma 15. *Let positive integers $n, \bar{m}, b, d, q \in \mathbb{Z}$ and real $\alpha \in (0,1)$ satisfy Equation (1) or (2). Then, games G_8 and G_7 are statistically indistinguishable. Moreover, $|\Pr[F_8|\neg E_8] - \Pr[F_7|\neg E_7]| \leq \mathsf{negl}(\kappa)$ and $|\Pr[E_8] - \Pr[E_7]| \leq \mathsf{negl}(\kappa)$.*

Proof. Note that the only differences between games G_8 and G_7 are the generation of the public key $pk = (\mathbf{A}, \mathbf{B})$ and the responses to the decryption queries. Concretely, in game G_7 the matrix $\mathbf{B} = -\mathbf{A}\mathbf{R}$ is generated by using $\mathbf{R} \overset{\$}{\leftarrow} (D_{\mathbb{Z}^{\bar{m}}, \omega(\sqrt{\log n})})^{nk}$, while in game G_7 the matrix $\mathbf{B} = -\mathbf{A}\mathbf{R}' - \mathsf{FRD}(tag^*)\mathbf{G}_b$, where $\mathbf{R}' \overset{\$}{\leftarrow} (D_{\mathbb{Z}^{\bar{m}}, \omega(\sqrt{\log n})})^{nk}$. Since $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$ is always uniformly chosen at random in both games, we have that $-\mathbf{A}\mathbf{R}$ and $-\mathbf{A}\mathbf{R}'$ are statistically close to uniform distribution over $\mathbb{Z}_q^{n \times nk}$ by Lemma 2. Namely, the public keys in games G_8 and G_7 are statistically close (and tag^* is statistically hidden in game G_8).

It suffices to show that in the adversary's view, the responses to the decryption queries are indistinguishable in games G_8 and G_7. Since for a decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$, the challenger will use

the same rules to check if the query can be immediately responded with $\perp$ in both games, we only have to consider the decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$ that needs the challenger $\mathcal{C}$ to perform the decryption operation. By the definition of game G_7, we must have that $tag = \mathsf{H}(\mathbf{c}_1) \neq tag^*$ holds for such decryption query $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4)$. Note that in game G_7, the challenger has the real secret key $sk = \mathbf{R}$, and can run the decryption algorithm to handle this query. We now show that the challenger $\mathcal{C}$ in game G_8 can almost perfectly simulate the decryption operation. Recall that $pk = (\mathbf{A}, \mathbf{B} = -\mathbf{A}\mathbf{R}' - \mathsf{FRD}(tag^*)\mathbf{G}_b)$, conditioned on $tag = \mathsf{H}(\mathbf{c}_1) \neq tag^*$ we have that $\mathbf{R}'$ is a valid trapdoor for $\mathbf{A}_{tag} = (\mathbf{A}\|\mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b)$, and thus can be used to compute $\tilde{\mathbf{s}} \leftarrow \mathsf{Solve}(\mathbf{A}_{tag}, \mathbf{R}', \mathbf{u})$. Now, either there exists a tuple $(\tilde{\mathbf{s}}, \mathbf{e}_1, \mathbf{e}_2)$ such that $\|\mathbf{e}_1\| \leq \alpha q \sqrt{\bar{m}}$, $\|\mathbf{e}_2\| \leq \gamma \sqrt{nk}$ for prime q (or $\|\mathbf{e}_2\|_\infty \leq \gamma \cdot \omega(\sqrt{\log n})$ for $q = b^k$), $\mathbf{c}_1 = \mathbf{A}^T \tilde{\mathbf{s}} + \mathbf{e}_1$ and $\mathbf{c}_2 = (\mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b)^T \tilde{\mathbf{s}} + \mathbf{e}_2$, or there does not. For the latter case, the challenger will always return $\perp$ in both games. While for the former case, the challenger $\mathcal{C}$ in game G_8 can recover $\tilde{\mathbf{s}}$ as long as $\|(\mathbf{R}')^T \mathbf{e}_1 + \mathbf{e}_2\| < \frac{q}{2\sqrt{b^2+1}}$ for prime q (or $\|(\mathbf{R}')^T \mathbf{e}_1 + \mathbf{e}_2\|_\infty < \frac{q}{2b}$ for $q = b^k$), which is essentially the same constraint for a correct decryption using $sk = \mathbf{R}$ in game G_7. Since both $\mathbf{R}$ and $\mathbf{R}'$ are chosen from the same Gaussian distribution, by Lemma 1 we have that the inequality will holds with the same overwhelming probability conditioned on $\|\mathbf{e}_1\| \leq \alpha q \sqrt{\bar{m}}$ and $\|\mathbf{e}_2\| \leq \gamma \sqrt{nk}$ for prime q (or $\|\mathbf{e}_2\|_\infty \leq \gamma \cdot \omega(\sqrt{\log n})$ for $q = b^k$). By the fact that the challengers in both games will perform the same operations after obtaining $\tilde{\mathbf{s}}$, we have that the responses to such kind of decryption queries are identical in both games except with negligible probability. This finishes the proof. $\square$

Remark 1. Note that the challenger in game G_8 actually does not have the "real" secret key, which implies that the adversary cannot obtain extra information about the secret key from the decryption queries (except what is obtained from the public key $pk = (\mathbf{A}, \mathbf{B})$). This fact will be used in our later proofs.

Game G_9 This game is similar to game G_8 except that the challenger $\mathcal{C}$ changes the KeyGen phase as follows:

KeyGen. first randomly choose $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{R}' \xleftarrow{\$} (D_{\mathbb{Z}^{\bar{m}}, \omega(\sqrt{\log n})})^{nk}$, $\mathbf{s}^* \xleftarrow{\$} D_{\mathbb{Z}^n, \alpha q}$ and $\mathbf{e}_1^* \xleftarrow{\$} D_{\mathbb{Z}^{\bar{m}}, \alpha q}, \mathbf{e}_2' \xleftarrow{\$} D_{\mathbb{Z}^{nk}, r}$ and $x^*, y^*, z^* \xleftarrow{\$} \mathbb{F}_{2^\kappa}$, where $r = \alpha q \sqrt{\bar{m}} \cdot \omega(\sqrt{\log n})$. Then, interpret the bit-concatenation of $(x^*, y^*, z^*) \in (\mathbb{F}_{2^\kappa})^3$ as a vector $\mathbf{v}^* = x^* \| y^* \| z^* \in \mathbb{Z}_d^n$, and compute

$$\tilde{\mathbf{s}}^* = \mathbf{s}^* + \mathsf{encode}_d(\mathbf{v}^*), \qquad \mathbf{c}_1^* = \mathbf{A}^T \tilde{\mathbf{s}}^* + \mathbf{e}_1^*,$$
$$\mathbf{B} = -\mathbf{A}\mathbf{R}' - \mathsf{FRD}(tag^*)\mathbf{G}_b, \qquad \mathbf{c}_2^* = (-\mathbf{R}')^T \mathbf{c}_1^* + \mathbf{e}_2'$$

where $tag^* = \mathsf{H}(\mathbf{c}_1^*)$. Finally, give the public key $pk = (\mathbf{A}, \mathbf{B})$ to $\mathcal{A}$, and keep $(\mathbf{R}', \mathbf{c}_1^*, \mathbf{c}_2^*, x^*, y^*, z^*)$ secret.

Lemma 16. *Let positive integers $n, \bar{m}, b, d, q \in \mathbb{Z}$ and real $\alpha \in (0, 1)$ satisfy Equation (1) or (2). Then, games G_9 and G_8 are statistically indistinguishable. Moreover, $|\Pr[F_9|\neg E_9] - \Pr[F_8|\neg E_8]| \leq \mathsf{negl}(\kappa)$ and $|\Pr[E_9] - \Pr[E_8]| \leq \mathsf{negl}(\kappa)$.*

Proof. Note that the only difference between games G_9 and G_8 is the generation of $\mathbf{c}_2^*$. In game G_8, $\mathbf{c}_2^* = (\mathbf{B} + \mathsf{FRD}(tag^*)\mathbf{G}_b)^T \tilde{\mathbf{s}}^* + \mathbf{e}_2^* = -(\mathbf{R}')^T \mathbf{A}^T \tilde{\mathbf{s}}^* + \mathbf{e}_2^*$ is generated by using $\mathbf{e}_2^* \xleftarrow{\$} D_{\mathbb{Z}^{nk}, \gamma}$ where $\gamma = \sqrt{\|\mathbf{e}_1^*\|^2 + \bar{m}(\alpha q)^2} \cdot \omega(\sqrt{\log n})$, while in game G_9, $\mathbf{c}_2^* = (-\mathbf{R}')^T \mathbf{c}_1^* + \mathbf{e}_2'$ is generated by using $\mathbf{e}_2' \xleftarrow{\$} D_{\mathbb{Z}^{nk}, r}$. Since $\mathbf{c}_1^* = \mathbf{A}^T \tilde{\mathbf{s}}^* + \mathbf{e}_1^*$ for some $\mathbf{e}_1^* \xleftarrow{\$} D_{\mathbb{Z}^{\bar{m}}, \alpha q}$, we have that $\mathbf{c}_2^* = (-\mathbf{R}')^T \mathbf{c}_1^* + \mathbf{e}_2' = (-\mathbf{R}')^T \mathbf{A}^T \tilde{\mathbf{s}}^* + \tilde{\mathbf{e}}_2$ for some $\tilde{\mathbf{e}}_2 = (-\mathbf{R}')^T \mathbf{e}_1^* + \mathbf{e}_2'$ which is distributed statistically close to $D_{\mathbb{Z}^{nk}, \gamma}$ by applying Lemma 3 nk times using a standard hybrid argument. Thus, the distributions of $\mathbf{c}_2^*$ in games G_9 and G_8 are actually statistically close, which in turn shows that both games are statistically indistinguishable in the adversary's view. $\square$

Game G_{10} This game is similar to game G_9 except that the challenger $\mathcal{C}$ changes the KeyGen phase as follows:

KeyGen. first randomly choose $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times \bar{m}}, \mathbf{b} \xleftarrow{\$} \mathbb{Z}_q^{m_1}, \mathbf{R}' \xleftarrow{\$} (D_{\mathbb{Z}^{\bar{m}}, \omega(\sqrt{\log n})})^{nk}, \mathbf{e}_2' \xleftarrow{\$} D_{\mathbb{Z}^{nk}, r}$ and $x^*, y^*, z^* \xleftarrow{\$} \mathbb{F}_{2^\kappa}$, where $r = \alpha q \sqrt{\bar{m}} \cdot \omega(\sqrt{\log n})$. Then, compute

$$\mathbf{c}_1^* = \mathbf{b}, \qquad \mathbf{B} = -\mathbf{A}\mathbf{R}' - \mathsf{FRD}(tag^*)\mathbf{G}_b,$$
$$\mathbf{c}_2^* = (-\mathbf{R}')^T \mathbf{c}_1^* + \mathbf{e}_2'$$

where $tag^* = \mathsf{H}(\mathbf{c}_1^*)$. Finally, give the public key $pk = (\mathbf{A}, \mathbf{B})$ to $\mathcal{A}$, and keep $(\mathbf{R}', \mathbf{c}_1^*, \mathbf{c}_2^*, x^*, y^*, z^*)$ secret.

Lemma 17. *If $\mathrm{LWE}_{n,\bar{m},q,\alpha}$ is hard, then games G_{10} and G_9 are computationally indistinguishable. Moreover, $|\Pr[F_{10}|\neg E_{10}] - \Pr[F_9|\neg E_9]| \le \mathrm{negl}(\kappa)$ and $|\Pr[E_{10}] - \Pr[E_9]| \le \mathrm{negl}(\kappa)$.*

Proof. We prove this lemma by showing that if there is a PPT adversary $\mathcal{A}$ that distinguishes game G_{10} from G_9 with non-negligible advantage, then there is an efficient algorithm $\mathcal{B}$ that solves the $\mathrm{LWE}_{n,\bar{m},q,\alpha}$ problem with the same advantage by interacting with $\mathcal{A}$.

Formally, given an LWE challenge tuple $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{n \times \bar{m}} \times \mathbb{Z}_q^{\bar{m}}$, $\mathcal{B}$ randomly chooses $\mathbf{R}' \xleftarrow{\$} (D_{\mathbb{Z}^{\bar{m}}, \omega(\sqrt{\log n})})^{nk}$, $\mathbf{e}_2' \xleftarrow{\$} D_{\mathbb{Z}^{nk}, r}$ and $x^*, y^*, z^* \xleftarrow{\$} \mathbb{F}_{2^\kappa}$, where $r = \alpha q \sqrt{\bar{m}} \cdot \omega(\sqrt{\log n})$. Then, it interprets the bit-concatenation of $(x^*, y^*, z^*) \in (\mathbb{F}_{2^\kappa})^3$ as a vector $\mathbf{v}^* = x^* \| y^* \| z^* \in \mathbb{Z}_d^n$, and computes

$$\mathbf{c}_1^* = \mathbf{b} + \mathbf{A}^T \mathsf{encode}_d(\mathbf{v}^*), \qquad \mathbf{B} = -\mathbf{A}\mathbf{R}' - \mathsf{FRD}(tag^*)\mathbf{G}_b,$$
$$\mathbf{c}_2^* = (-\mathbf{R}')^T \mathbf{c}_1^* + \mathbf{e}_2'$$

where $tag^* = \mathsf{H}(\mathbf{c}_1^*)$. Then, $\mathcal{B}$ sets the public key $pk = (\mathbf{A}, \mathbf{B})$, and keeps $(\mathbf{R}', \mathbf{c}_1^*, \mathbf{c}_2^*, x^*, y^*, z^*)$ private. Finally, $\mathcal{B}$ gives pk to the adversary $\mathcal{A}$, simulates the attack environment the same as in game G_9, and returns whatever $\mathcal{A}$ outputs as its own output.

Now, if $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{n \times \bar{m}} \times \mathbb{Z}_q^{\bar{m}}$ is a valid LWE tuple, i.e., $\mathbf{b} = \mathbf{A}^T \mathbf{s}^* + \mathbf{e}^*$ for some $\mathbf{s}^* \xleftarrow{\$} D_{\mathbb{Z}^n, \alpha q}$ and $\mathbf{e}^* \xleftarrow{\$} D_{\mathbb{Z}^{\bar{m}}, \alpha q}$, then we have that $\mathbf{c}_1^* = \mathbf{b} + \mathbf{A}^T \mathsf{encode}_d(\mathbf{v}) = \mathbf{A}^T \tilde{\mathbf{s}}^* + \mathbf{e}^*$, where $\tilde{\mathbf{s}}^* = \mathbf{s}^* + \mathsf{encode}_d(\mathbf{v}^*)$. In this case, $\mathcal{B}$ perfectly simulates the attack environment in game G_9 for $\mathcal{A}$. Else if $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{n \times \bar{m}} \times \mathbb{Z}_q^{\bar{m}}$ is uniformly random, then $\mathbf{c}_1^* = \mathbf{b} + \mathbf{A}^T \mathsf{encode}_d(\mathbf{v}^*)$ is also uniformly random over $\mathbb{Z}_q^{\bar{m}}$. This means that $\mathcal{B}$ perfectly simulates the attack environment in game G_{10} for $\mathcal{A}$. Thus, if $\mathcal{A}$ can distinguish game G_{10} from G_9 with non-negligible advantage, then $\mathcal{B}$ can solve the $\mathrm{LWE}_{n,\bar{m},q,\alpha}$ problem with the same advantage. $\square$

Game G_{11} This game is similar to game G_{10} except that the challenger $\mathcal{C}$ changes the KeyGen phase as follows:

KeyGen. first randomly choose $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{R}' \xleftarrow{\$} (D_{\mathbb{Z}^{\bar{m}}, \omega(\sqrt{\log n})})^{nk}$, $\mathbf{b} \xleftarrow{\$} \mathbb{Z}_q^{\bar{m}}, \mathbf{d} \xleftarrow{\$} \mathbb{Z}_q^{nk}$ and $x^*, y^*, z^* \xleftarrow{\$} \mathbb{F}_{2^\kappa}$. Then, compute
$$\mathbf{c}_1^* = \mathbf{b}, \qquad \mathbf{B} = -\mathbf{A}\mathbf{R}' - \mathsf{FRD}(tag^*)\mathbf{G}_b,$$
$$\mathbf{c}_2^* = \mathbf{d},$$

where $tag^* = \mathsf{H}(\mathbf{c}_1^*)$. Finally, give the public key $pk = (\mathbf{A}, \mathbf{B})$ to $\mathcal{A}$, and keep $(\mathbf{R}', \mathbf{c}_1^*, \mathbf{c}_2^*, x^*, y^*, z^*)$ secret.

Lemma 18. *Let positive $n, \bar{m}, b, d, q \in \mathbb{Z}$ and real $\alpha \in (0,1)$ satisfy Equation (1) or (2), then games G_{11} and G_{10} are statistically indistinguishable. Moreover, $|\Pr[F_{11}|\neg E_{11}] - \Pr[F_{10}|\neg E_{10}]| \le \mathrm{negl}(\kappa)$ and $|\Pr[E_{11}] - \Pr[E_{10}]| \le \mathrm{negl}(\kappa)$.*

Proof. Note that the only difference between games G_{11} and G_{10} is the generation of $\mathbf{c}_2^*$ in the challenge ciphertext. Thus, it is enough to show that $\mathbf{c}_2^*$ in game G_{11} is actually statistically close to that in game G_{10}. Note that $\mathbf{c}_1^* = \mathbf{b}$ is uniformly chosen from $\mathbb{Z}_q^{\bar{m}}$ at random in both games, and $\mathbf{c}_2^* = (-\mathbf{R}')^T \mathbf{c}_1^* + \mathbf{e}_2'$ in game G_{10}. Using the facts that $\bar{m} \ge (n+1) \log_2 q + \omega(\log n)$ and $\mathbf{R}' \xleftarrow{\$} (D_{\mathbb{Z}^{\bar{m}}, \omega(\sqrt{\log n})})^{nk}$, we have that $(\mathbf{A}, \mathbf{A}\mathbf{R}', \mathbf{b}, (\mathbf{R}')^T \mathbf{b})$ is statistically close to uniform by Lemma 2. In other words, $\mathbf{c}_2^*$ in game G_{10} is essentially statistically close to uniform over $\mathbb{Z}_q^{nk}$, which completes the proof. $\square$

Lemma 19. $\Pr[F_{11}] = 1/2$ *and* $\Pr[E_{11}] = \mathrm{negl}(\kappa)$.

Proof. Since $x^* \xleftarrow{\$} \{0,1\}^\ell$ is uniformly chosen at random, $\mu_{\delta^*} \in \{0,1\}^\ell$ is perfectly hidden in the challenge ciphertext $C^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, c_3^*, c_4^*)$, where $c_3^* = x^* + \mu_{\delta^*} \in \mathbb{F}_{2^\kappa}$. Thus, $\Pr[F_{11}] = \Pr[\delta = \delta^*] = 1/2$, where $\delta \in \{0,1\}$ is output by the adversary $\mathcal{A}$ for the guess of δ^* in game G_{11}.

As for the second claim, since $y^*, z^* \in \{0,1\}^\kappa$ are uniformly chosen at random in game G_{11}, given $c_4^* = \tau^* y^* + z^* \in \mathbb{F}_{2^\kappa}$ there are still 2^κ possible choices of (y^*, z^*). Thus, for any adversary $\mathcal{A}$ with the knowledge of $c_4^* = \tau^* y^* + z^*$, the probability that it outputs $c_4 = \tau y^* + z^*$ for any $\tau \neq \tau^*$ is at most $1/2^\kappa$ (because the adversary can uniquely determine (y^*, z^*) if he can output a valid $c_4 = \tau y^* + z^*$), which means that $\Pr[E_{11}] \leq Q_{dec}/2^\kappa$, where Q_{dec} is the maximum number of decryption queries made by $\mathcal{A}$. $\square$

In all, we have that $\Pr[F_1] \leq 1/2 + \mathsf{negl}(\kappa)$ by Lemma 9~19. This completes the proof of Theorem 1. $\square$

4 Optimizations

4.1 Encrypting long message

In the description of our PKE, we only consider to encrypt a κ-bit message for simplicity. Although it suffices for many applications where a PKE is typically used to encrypt a session key for some symmetric encryption such as AES, our PKE can essentially encrypt messages of bit length up to $O(n \log n) - 2\kappa$. Note that in order to recover $\mathbf{v} \in \mathbb{Z}_d^n$ from $\tilde{\mathbf{s}} = \mathbf{s} + \mathsf{encode}_d(\mathbf{v}) \in \mathbb{Z}_q^n$, it is enough to set the parameters such that $\|\mathbf{s}\|_\infty < \frac{q-(d-1)d}{2d}$ by Lemma 7. For any $n, \bar{m}, q \in \mathbb{Z}$ and real $\alpha \in \mathbb{R}$ satisfying Equation (1) or (2), we always have $\|\mathbf{s}\|_\infty \leq \alpha q \cdot \omega(\sqrt{\log n}) < q/\tilde{O}(\sqrt{n})$. This makes it possible to set $d = \tilde{O}(\sqrt{n})$ and encrypt messages of bit length up to $O(n \log n) - 2\kappa$ (since one can encode $n \log_2 d = O(n \log n)$-bit information into the vector $\mathbf{v} \in \mathbb{Z}_d^n$). Concretely, for the choice of $(n, m, q, \alpha q) = (450, 10740, 3^9, 1.5)$, we have that $\|\mathbf{s}\|_\infty \leq 9$ holds except with probability less than 2^{-162}, which allows us to set $d = 128$ and encrypt messages of size up to 2638-bit. We also note that one can use a pseudorandom generator PRG to encrypt any polynomial long messages by replacing $c_3 = x + \mu$ with $c_3 = \mathsf{PRG}(x) \oplus \mu$.

4.2 Compressing the Ciphertext

As many lattice-based PKEs in the literature (e.g., [48,17,5]), it is possible to discard some lower bits of the ciphertext (and thus reduce the ciphertext size) without affecting the correctness of the PKEs (because those lower bits mainly carry noise). This can be seen as a modulus switch technique. Concretely, let $\mathsf{Switch}_{q,p}(\cdot) : \mathbb{Z}_q \to \mathbb{Z}_p$ be a function defined as $\mathsf{Switch}_{q,p}(x) = \lceil p/q \cdot x \rfloor \mod p$. It is easy to check that for any $x \in \mathbb{Z}_q$, $x - \mathsf{Switch}_{p,q}(\mathsf{Switch}_{q,p}(x)) \leq \lceil \frac{q}{2p} \rceil$. Thus, for $p < q$, one can use $\mathsf{Switch}_{q,p}(\cdot)$ to compress the ciphertext in the encryption algorithm (i.e., applying to vectors in a coordinate-wise way), and use $\mathsf{Switch}_{p,q}(\cdot)$ to approximately recover the original ciphertext in the decryption algorithm. This can be simply seen as adding a noise of size at most $\lceil \frac{q}{2p} \rceil$ to each coordinate of the lattice vectors in the ciphertext.

In our case, we cannot simply apply $\mathsf{Switch}_{q,p}(\cdot)$ to the ciphertext in a black-box way, since this will affect both the correctness and the security of our PKE. Instead, we have to plug it into the encryption algorithm to generate the ciphertext $C = (\mathbf{c}_1, \mathbf{c}_2', c_3, c_4')$ as follows (where p is an integer, and other notations are the same as before):

$$
\begin{aligned}
\mathbf{c}_1 &= \mathbf{A}^T \tilde{\mathbf{s}} + \mathbf{e}_1, & tag &= \mathsf{H}(\mathbf{c}_1) \\
\mathbf{c}_2' &= \mathsf{Switch}_{q,p}((\mathbf{B} + \mathsf{FRD}(tag)\mathbf{G}_b)^T \tilde{\mathbf{s}} + \mathbf{e}_2), & c_3 &= x + \mu \\
\tau' &= \mathsf{H}(\mathbf{c}_2', c_3), & c_4' &= \tau' y + z.
\end{aligned}
$$

For the choice of $(n, m, q, \alpha q) = (450, 10740, 3^9, 1.5)$, we can set $p = 8$ to compress the ciphertext from previous 19.73 KB to 13.80 KB, while still keep the decryption error rate less than 2^{-100}. Note that we do not use this technique to compress $\mathbf{c}_1$, because unlike the error in $\mathbf{c}_2$, any error in $\mathbf{c}_1$ will be sharply amplified by a factor of $s_1(\mathbf{R})$ in decryption.

4.3 Compressing the Keys

The key sizes of LWE-based PKEs (e.g., [52,48,1,45,68]) are usually very large due to consist of big matrices in both the public keys and the secret keys. For example, under the choice of $(n, m, q, \alpha q) = (450, 10740, 3^9, 1.5)$, the public key and secret key sizes of our PKE are about 8.64 MB and 16.15 MB, respectively. However, the first element in the public key is essentially a uniformly random matrix which can be treated as a system parameter and shared among all users. By doing this, one can reduce the sizes of the public key from previous 8.64 MB to 3.26 MB. Besides, one can also use a PRG with a 256-bit random seed to deterministically generate the secret key matrix $\mathbf{R}$, and reduce the secret key size from 16.15 MB to 32 Bytes. As we will show in the supplemental material, one can also reduce the key sizes by adapting our construction to the ring setting.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EURO-CRYPT 2010, LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
2. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010, LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010)
3. Ajtai, M.: Generating hard instances of the short basis problem. In: Wiedermann, J., van Emde Boas, P., Nielsen, M. (eds.) Automata, Languages and Programming, LNCS, vol. 1644, pp. 706–706. Springer, Heidelberg (1999)
4. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. In Journal of Mathematical Cryptology 9, 169–203 (oct 2015)
5. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Newhope without reconciliation. Cryptology ePrint Archive, Report 2016/1157 (2016)
6. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange-a new hope. In: USENIX Security Symposium. vol. 2016 (2016)
7. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In STACS pp. 75–86 (2009)
8. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) CRYPTO 2009, LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)
9. Babai, L.: On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica 6(1), 1–13 (Mar 1986)
10. Banaszczyk, W.: New bounds in some transference theorems in the geometry of numbers. Mathematische Annalen 296, 625–635 (1993)
11. Benhamouda, F., Blazy, O., Ducas, L., Quach, W.: Hash proof systems over lattices revisited. Cryptology ePrint Archive, Report 2017/997 (2017)
12. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Comput. 36(5), 1301–1328 (Dec 2006)
13. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D., Wang, X. (eds.) ASIACRYPT 2011, LNCS, vol. 7073, pp. 41–69. Springer, Heidelberg (2011)
14. Boneh, D., Katz, J.: Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In: Menezes, A. (ed.) CT-RSA 2005, LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
15. Bos, J., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. In: ACM CCS 2016. pp. 1006–1018. ACM (2016)

16. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: Fully homomorphic encryption without bootstrapping. Innovations in Theoretical Computer Science, ITCS pp. 309–325 (2012)
17. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: FOCS 2011. pp. 97 –106 (oct 2011)
18. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) CRYPTO 2011, LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011)
19. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM 51(4), 557–594 (Jul 2004)
20. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004, LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
21. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010, LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
22. Cramer, R., Damgård, I.: On the amortized complexity of zero-knowledge protocols. In: Halevi, S. (ed.) CRYPTO 2009, LNCS, vol. 5677, pp. 177–191. Springer, Heidelberg (2009)
23. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO ’98, LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
24. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Comput. 33, 167–226 (2001)
25. Diffie, W., Hellman, M.: New directions in cryptography. Information Theory, IEEE Transactions on 22(6), 644 – 654 (nov 1976)
26. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. SIAM J. Comput. 30(2), 391–437 (2000)
27. Döttling, N., Garg, S., Hajiabadi, M., Masny, D.: New constructions of identity-based and key-dependent message secure encryption schemes. Cryptology ePrint Archive, Report 2017/978 (2017)
28. Dowsley, R., Hanaoka, G., Imai, H., Nascimento, A.C.: Reducing the ciphertext size of Dolev-Dwork-Naor like public key cryptosystems. Cryptology ePrint Archive 2009, 271 (2009)
29. Ducas, L., Micciancio, D.: Improved short lattice signatures in the standard model. In: Garay, J., Gennaro, R. (eds.) CRYPTO 2014, LNCS, vol. 8616, pp. 335–352. Springer, Heidelberg (2014)
30. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology 26(1), 80–101 (Jan 2013)
31. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008. pp. 197–206. ACM (2008)
32. Gertner, Y., Malkin, T., Myers, S.: Towards a separation of semantic and CCA security for public key encryption. In: Vadhan, S.P. (ed.) Theory of Cryptography, LNCS, vol. 4392, pp. 434–455. Springer, Heidelberg (2007)
33. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270 – 299 (1984)
34. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisted. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018. pp. 96–125. Springer,Cham (2018)
35. Katsumata, S., Yamada, S.: Partitioning via non-linear polynomial functions: More compact IBEs from ideal lattices and bilinear maps. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, vol. 10032, pp. 682–712. Springer, Heidelberg (2016)
36. Katz, J., Vaikuntanathan, V.: Smooth projective hashing and password-based authenticated key exchange from lattices. In: Matsui, M. (ed.) ASIACRYPT 2009, LNCS, vol. 5912, pp. 636–652. Springer, Heidelberg (2009)
37. Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) Theory of Cryptography 2006, LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
38. Kiltz, E., Mohassel, P., O’Neill, A.: Adaptive trapdoor functions and chosen-ciphertext security. In: Gilbert, H. (ed.) EUROCRYPT 2010, LNCS, vol. 6110, pp. 673–692. Springer, Heidelberg (2010)
39. Kim, S., Wu, D.J.: Multi-theorem preprocessing NIZKs from lattices. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018. pp. 733–765. Springer,Cham (2018)
40. Klein, P.: Finding the closest lattice vector when it’s unusually close. In: Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 937–941. SODA ’00, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2000)
41. Libert, B., Sakzad, A., Stehlé, D., Steinfeld, R.: All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology – CRYPTO 2017. pp. 332–364. Springer, Cham (2017)
42. 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)

43. Lyubashevsky, V., Micciancio, D.: Asymptotically efficient lattice-based digital signatures. Journal of Cryptology 31(3), 774–797 (July 2018)
44. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010, LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010)
45. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012, LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)
46. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC 1990. pp. 427–437. ACM (1990)
47. NIST: Post-quantum cryptography standardization (2016), `http://csrc.nist.gov/groups/ST/post-quantum-crypto/submission-requirements/index.html`
48. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC 2009. pp. 333–342. ACM (2009)
49. Peikert, C.: An efficient and parallel Gaussian sampler for lattices. In: Rabin, T. (ed.) CRYPTO 2010, LNCS, vol. 6223, pp. 80–97. Springer, Heidelberg (2010)
50. Peikert, C., Regev, O., Stephens-Davidowitz, N.: Pseudorandomness of ring-LWE for any ring and modulus. In: STOC 2017. pp. 461–473. ACM (2017)
51. 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)
52. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC 2008. pp. 187–196. ACM (2008)
53. Pointcheval, D.: Chosen-ciphertext security for any one-way cryptosystem. In: Imai, H., Zheng, Y. (eds.) PKC '00, LNCS, vol. 1751, pp. 129–146. Springer, Heidelberg (2000)
54. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO '91. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
55. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005. pp. 84–93. ACM (2005)
56. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21(2), 120–126 (Feb 1978)
57. Rosen, A., Segev, G.: Chosen-ciphertext security via correlated products. In: Reingold, O. (ed.) Theory of Cryptography, LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)
58. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: FOCS '99. pp. 543–553. IEEE Computer Society (1999)
59. Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018. pp. 520–551. Springer, Cham (2018)
60. Shor, P.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26(5), 1484–1509 (1997)
61. Shoup, V.: Sequences of games: a taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332 (2004)
62. Stehlé, D., Steinfeld, R.: Making NTRU as secure as worst-case problems over ideal lattices. In: Paterson, K. (ed.) EUROCRYPT 2011, LNCS, vol. 6632, pp. 27–47. Springer, Heidelberg (2011)
63. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: Advances in Cryptology – ASIACRYPT 2009, pp. 617–635. Springer, Heidelberg (2009)
64. Steinfeld, R., Ling, S., Pieprzyk, J., Tartary, C., Wang, H.: NTRUCCA: How to strengthen ntruencrypt to chosen-ciphertext security in the standard model. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012, LNCS, vol. 7293, pp. 353–371. Springer, Heidelberg (2012)
65. Targhi, E.E., Unruh, D.: Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In: Hirt, M., Smith, A. (eds.) Theory of Cryptography, pp. 192–216. Springer, Heidelberg (2016)
66. Vershynin, R.: Introduction to the non-asymptotic analysis of random matrices. arXiv preprint arXiv:1011.3027 (2010)
67. Wee, H.: Efficient chosen-ciphertext security via extractable hash proofs. In: Rabin, T. (ed.) CRYPTO 2010, LNCS, vol. 6223, pp. 314–332. Springer, Heidelberg (2010)
68. Wee, H.: Public key encryption against related key attacks. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012, vol. 7293, pp. 262–279. Springer, Heidelberg (2012)
69. Yamada, S.: Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, LNCS, vol. 9666, pp. 32–62. Springer, Heidelberg (2016)

70. Yamada, S.: Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, vol. 10403, pp. 161–193. Springer, Cham (2017)
71. Zhang, J., Chen, Y., Zhang, Z.: Programmable hash functions from lattices: Short signatures and IBEs with small key sizes. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, LNCS, vol. 9816, pp. 303–332. Springer, Heidelberg (2016)
72. Zhang, J., Yu, Y.: Two-round pake from approximate SPH andinstantiations from lattices. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017, pp. 37–67. Springer, Cham (2017)

A   CCA2-secure PKE from Ring-LWE

In this section, we adapt our construction in Section 3 to the ring setting. We begin with the introduction of some necessary backgrounds.

A.1   Rings, Ring-LWE and Trapdoors

Rings. Let n be a power of 2, and consider the cyclotomic ring $R = \mathbb{Z}[X]/(X^n + 1)$. For any integer $q > 0$, define $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ analogously. As in [62,29], we require that $f(X) = X^n + 1$ does not split into low degree polynomials modulo the prime factors of q. For simplicity, we let q be a power of 3. In this case, $f(X) = X^n + 1 \mod 3$ will split into two irreducible polynomials with degree $n/2$ in $\mathbb{Z}_3[X]$.

Lemma 20 ([29]). *Let $n \geq 4$ be a power of 2, $q \geq 3$ a power of 3, and $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. Then, any nonzero polynomial $t \in R_q$ of degree $d < n/2$ and coefficients in $\{-1,0,1\}$ is invertible in R_q.*

For any element $a \in R$ (or R_q), we identify a with its coefficient vector in $\mathbb{Z}^n$ (or $\mathbb{Z}_q^n$). Formally, we let $\phi : R \to \mathbb{Z}^n$ be the coefficient embedding that maps a polynomial into its coefficient vector (in the column form). Then, we define the norm of a polynomial as the (Euclidean) norm of its coefficient vector, i.e., $\|a\| = \|\phi(a)\|$. By identifying R with $\mathbb{Z}^n$ under the map ϕ, the discrete Gaussian distribution over the ring R can be defined as $D_{R,s} = D_{\mathbb{Z}^n,s}$. We also identity R with the sub-ring of anti-circulant matrices in $\mathbb{Z}^{n \times n}$ by viewing each ring element $a \in R$ as a linear transformation $r \to a \cdot r$ over the coefficient embedding of R, i.e., by treating a as a matrix in $\mathbb{Z}^{n \times n}$ such that the i-th column is the coefficient of $a \cdot X^i \mod f(X)$ where $i \in \{0, \ldots, n-1\}$. Formally, we let $\mathsf{Rot} : \mathbf{R} \to \mathbb{Z}^{n \times n}$ be the ring homomorphism that maps a polynomial into its anti-circulant matrix, i.e., $\mathsf{Rot}(a) = (\phi(a), \phi(aX), \ldots, \phi(aX^{n-1})) \in \mathbb{Z}^{n \times n}$. The definition of ϕ and Rot can be naturally extended to vectors and matrices over R (over R_q) in a coordinate-wise way.

Lemma 21 ([29]). *Let $n \geq 4$ be a power of 2, $q \geq 3$ a power of 3, $R_q = \mathbb{Z}_q[X]/(X^n + 1)$, and constant $c > 0$. Let integers $\bar{m} \geq 2\log_2 q + c, k = \mathrm{poly}(n)$ and real $s \geq \omega(\sqrt{\log n\bar{m}})$. With overwhelming probability over the choice of $\mathbf{a} \xleftarrow{\$} (R_q)^{\bar{m}}$, if $\mathbf{R} \xleftarrow{\$} (D_{R,s})^{k \times \bar{m}}$, then $\mathbf{Ra}$ is within negligible statistical distance $2^{-cn/2}$ from uniform distribution over R_q^k.*

Lemma 22 ([29]). *Let $n \geq 4$ be a power of 2, $q \geq 3$ a power of 3, and $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. Let $m, k > 0$ be integers, and let $s > 0$ be a real. Then, if $\mathbf{R} \xleftarrow{\$} (D_{R,s})^{k \times m}$, with overwhelming probability we have $s_1(\mathsf{Rot}(\mathbf{R})) \leq s\sqrt{n} \cdot O(\sqrt{k} + \sqrt{m} + \omega(\sqrt{\log n}))$.*

We also need the following lemma from [35].

Lemma 23 ([35]). *Let $n, m, q > 0$ be integers. Let $\mathbf{b} \in \mathbb{Z}_q^m$ be arbitrary and $\mathbf{x} \xleftarrow{\$} D_{\mathbb{Z}^m,r}$ with $r \geq \max\{\omega(\sqrt{\log n}), \omega(\sqrt{\log m})\}$. Then, for any matrix $\mathbf{R} \in \mathbb{Z}^{m \times n}$ and real $\sigma > s_1(\mathbf{R})$, there exists a PPT algorithm $\mathbf{ReRand}(\mathbf{R}, \mathbf{b} + \mathbf{x}, r, \sigma)$ that outputs $\mathbf{b}' = \mathbf{R}^T \mathbf{b} + \mathbf{x}' \in \mathbb{Z}_q^n$ where $\mathbf{x}'$ is distributed statistically close to $D_{\mathbb{Z}^n, 2r\sigma}$.*

By combining the results in Lemma 23 and Lemma 22 we have the following corollary, which has been implicitly used in [35] and will be used to replace Lemma 3 in the security proof.

Corollary 1. *Let $n \geq 4$ be a power of 2, $q \geq 3$ a power of 3, and $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. Let integers $\ell \leq \bar{m}$ be integers. Let $\mathbf{b} \in R_q^{\bar{m}}$ be arbitrary and $\mathbf{x} \overset{\$}{\leftarrow} (D_{R,r})^{\bar{m}}$ with $r \geq \omega(\sqrt{\log n\bar{m}})$. Then, for randomly chosen matrix $\mathbf{R} \overset{\$}{\leftarrow} (D_{R,\omega(\sqrt{\log n\bar{m}})})^{\ell \times \bar{m}}$, there exists a PPT algorithm $\mathbf{ReRand}'(\mathbf{R}, \mathbf{b} + \mathbf{x}, r, \sqrt{n\bar{m}} \cdot \omega(\sqrt{\log n\bar{m}}))$ that outputs $\mathbf{b}' = \mathbf{Rb} + \mathbf{x}' \in R_q^\ell$, where $\mathbf{x}'$ is distributed statistically close to $(D_{R,r\sqrt{n\bar{m}} \cdot \omega(\sqrt{\log n\bar{m}})})^\ell$.*

Proof. The algorithm $\mathbf{ReRand}'(\mathbf{R}, \mathbf{b} + \mathbf{x}, r, \sigma)$ first runs the algorithm $\mathbf{v} \leftarrow \mathbf{ReRand}(\mathsf{Rot}(\mathbf{R})^T, \phi(\mathbf{b} + \mathbf{x}), r, \sigma) \in \mathbb{Z}^{n\ell}$ in Lemma 23, and then outputs $\phi^{-1}(\mathbf{v}) \in \mathbf{R}^\ell$. It suffices to show that the above $\mathbf{ReRand}'$ satisfy the requirement. Since $\mathbf{R} \overset{\$}{\leftarrow} (D_{R,\omega(\sqrt{\log n\bar{m}})})^{\ell \times \bar{m}}$, we have that $s_1(\mathsf{Rot}(\mathbf{R})^T) \leq \sqrt{n\bar{m}} \cdot \omega(\sqrt{\log n\bar{m}})$ by Lemma 22. By Lemma 23, we have $\mathbf{v} = \mathsf{Rot}(\mathbf{R})\phi(\mathbf{b}) + \mathbf{u}$ for some $\mathbf{u}$ is distributed statistically close to $D_{\mathbb{Z}^{n\ell}, r\sqrt{n\bar{m}} \cdot \omega(\sqrt{\log n\bar{m}})}$. Since $\phi(\mathbf{Rb}) = \mathsf{Rot}(\mathbf{R})\phi(\mathbf{b})$, we have that $\phi^{-1}(\mathbf{v}) = \mathbf{Rb} + \phi^{-1}(\mathbf{u})$. Let $\mathbf{b}' = \phi^{-1}(\mathbf{v}) \in R_q^\ell$ and $\mathbf{x}' = \phi^{-1}(\mathbf{u}) \in \mathbf{R}^\ell$, we have that $\mathbf{x}'$ is distributed statistically close to $(D_{R,r\sqrt{n\bar{m}} \cdot \omega(\sqrt{\log n\bar{m}})})^\ell$, which completes the proof. □

Ring-LWE. Let n be a power of 2, and let q be a positive integer. Define $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. We tailor the ring-LWE to our particular setting, and refer to [44,50] for more information. Let $s \overset{\$}{\leftarrow} R_q$ and real $\alpha > 0$. Define $A_{s,\alpha}$ as the distribution of $(a, as + e) \in R_q \times R_q$, where $a \overset{\$}{\leftarrow} R_q$ and $e \overset{\$}{\leftarrow} D_{R,\alpha q}$ are randomly chosen. The (decisional) $\text{RLWE}_{\ell,q,\beta}$ problem was first considered in [63,44], which informally asks a PPT algorithm to distinguish the distribution $A_{s,\alpha}$ from uniform over $R_q \times R_q$ with $\ell = \text{poly}(n)$ samples. We have the following improved hardness result in [50].

Lemma 24 (A special case of [50]). *Let n be a power of 2. Let real $\beta \in (0,1)$, and integers $\ell = \text{poly}(n), q \geq 2$ such that $\beta q > \omega(1)$. Define $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. Then, there exists a polynomial time quantum reduction from γ-SIVP in the worst case on ideal lattices to average-case $\text{RLWE}_{\ell,q,\alpha}$, where $\alpha = \beta n^{3/4} \cdot (\ell/\log(n\ell))^{1/4}$ and $\gamma \leq \max\{\omega(\sqrt{n\log n}/\beta), \sqrt{2n}\}$.*

It has been proven that the ring-LWE assumption still holds even if the secret s is chosen according to the error distribution $D_{R,\alpha q}$ rather than uniformly [8,44]. This variant is known as the *Hermite normal form* of ring-LWE, and is preferable for controlling the size of the error term [18,16].

q-ary Lattices over Rings and Trapdoors. Let n be a power of 2, and let q be an integer. Define $R = \mathbb{Z}[X]/(X^n + 1)$ and $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. Let $\mathbf{a} \in R_q^{\bar{m}}$, define two lattices $\Lambda_q^\perp(\mathbf{a}) = \{\mathbf{e} \in R^{\bar{m}} \text{ s.t. } \mathbf{a}^T\mathbf{e} = 0\}$ and $\Lambda_q(\mathbf{a}) = \{\mathbf{y} \in R^{\bar{m}} \text{ s.t. } \exists s \in R_q, \mathbf{a}s = \mathbf{y}\}$. The following lemma is implicit in the proof of [29, Lemma. 7].

Lemma 25 ([29]). *Let $n \geq 4$ be a power of 2, $q \geq 3$ a power of 3, and $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. Let integer $\bar{m} \geq 2\log_2 q + c$ for some constant $c > 0$. Then, for all but an at most $2^{-cn/2}$ fraction of $\mathbf{a} \overset{\$}{\leftarrow} (R_q)^{\bar{m}}$, the shortest vector (in the ℓ_∞ norm) of $\Lambda_q(\mathbf{a})$ has ℓ_∞ norm at least $q/12$.*

As shown in [45,29], the trapdoor notion in [45] can be easily extended to the ring setting. Formally, let k be an integer and let $q = 3^k$. Let $\mathbf{g} = (1, 3, \ldots, 3^{k-1})^T \in R_q^k$ be the public primitive vector.

Lemma 26 (implicit in [45]). *Let $n \geq 4$ be a power of 2, and let k be an integer. Let $q = 3^k$ and let $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. Let $\mathbf{I}_k$ is the $k \times k$ identity matrix over R_q. For any $\mathbf{a} \in R_q^m$, $\mathbf{R} \in R_q^{k \times (m-k)}$ and invertible $t \in R_q$ satisfying $(\mathbf{R}\|\mathbf{I}_k)\mathbf{a} = \mathbf{g}t$, there exists an efficient algorithm $\mathsf{Solve}'(\mathbf{a}, \mathbf{R}, \mathbf{y})$ that given any $\mathbf{y} = \mathbf{a}s + \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \in R_q^m$ satisfying $\|\mathbf{R}\mathbf{e}_1 + \mathbf{e}_2\|_\infty < q/6$, outputs $s \in R_q$, where $\mathbf{e}_1 \in R^{m-k}$ and $\mathbf{e}_2 \in R^k$.*

Proof. First, we note that given any $\mathbf{y}' = \mathbf{g}s' + \mathbf{e}' \in R_q^k$, we have $\phi(\mathbf{y}') = \mathsf{Rot}(\mathbf{g}')\phi(s') + \phi(\mathbf{e}')$. By the construction of $\mathbf{g}$, we have that $\Lambda_q^\perp(\mathbf{g})$ has a short basis $\mathbf{B}$ such that the norm of its Gram-Schmidt orthogonalization $\tilde{\mathbf{B}} = 3\mathbf{I}_{nk}$. By using the algorithms in [9,40,45,49], the basis $\mathbf{B}$ can be used to invert the function $\phi(\mathbf{y}') = \mathsf{Rot}(\mathbf{g})\phi(s') + \phi(\mathbf{e}')$ as long as $\|\mathbf{e}'\|_\infty < q/(2\|\tilde{\mathbf{B}}\|) = q/6$. Thus, there exists an algorithm $\mathsf{Invert}(\mathbf{y}')$ which takes as input $\mathbf{y}' = \mathbf{g}s' + \mathbf{e}' \in R_q^k$ satisfying $\|\mathbf{e}'\|_\infty < q/6$, outputs $s' \in R_q$.

Now, given $\mathbf{y} = \mathbf{a}s + \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \in R_q^m$, the $\mathsf{Solve}'(\mathbf{a}, \mathbf{R}, \mathbf{y})$ algorithm first computes $\mathbf{y}' = (\mathbf{R}\|\mathbf{I}_k)\mathbf{y} = \mathbf{g}ts + \mathbf{R}\mathbf{e}_1 + \mathbf{e}_2$. Then, it runs the above invert algorithm to recover $s' = \mathsf{Invert}(\mathbf{y}')$. Finally, it returns $s = s't^{-1}$. Note that if the $\|\mathbf{R}\mathbf{e}_1 + \mathbf{e}_2\|_\infty < q/6$, we must have $s' = ts \in R_q$ by the correctness of Invert. Since t is invertible by assumption, we have that $s = s't^{-1}$ is exactly the desired solution, which completes the proof. $\square$

We also need the following lemma for constructing standard model PKE from ring-LWE, which may be of independent interest.

Lemma 27. *Let $n \geq 4$ be a power of 2, $q \geq 3$ a power of 3, and $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. Let integer $\bar{m} \geq 2\log_2 q + c$ for some constant $c > 0$. Then, for all but an at most $2^{-cn/2}$ fraction of $\mathbf{a} \xleftarrow{\$} (R_q)^{\bar{m}}$, for any $\mathbf{u} \in R_q^{\bar{m}}$ there exists at most one pair $(s, \mathbf{e}) \in R_q \times R^{\bar{m}}$ such that $\|\mathbf{e}\|_\infty < q/24$ and $\mathbf{u} = \mathbf{a}s + \mathbf{e}$.*

Proof. The proof is similar to that of Lemma 6. For any $\mathbf{u} \in R_q^{\bar{m}}$, we assume that there exist two tuples $(s, \mathbf{e}) \neq (s', \mathbf{e}') \in R_q \times R^{\bar{m}}$, such that $\|\mathbf{e}\|_\infty, \|\mathbf{e}'\|_\infty < q/24$ and $\mathbf{u} = \mathbf{a}s + \mathbf{e} = \mathbf{a}s' + \mathbf{e}'$. Letting $\tilde{s} = s - s'$ and $\tilde{\mathbf{e}} = \mathbf{e}' - \mathbf{e}$, we have that $\mathbf{a}\tilde{s} = \tilde{\mathbf{e}}$ for some $\tilde{s} \neq 0$ and $\|\tilde{\mathbf{e}}\|_\infty < q/12$. By Lemma 25, for all but an at most 2^{-cn} fraction of $\mathbf{a} \in R_q^{\bar{m}}$, the vector $\mathbf{a}\tilde{s}$ has ℓ_∞ norm $\|\mathbf{a}\tilde{s}\|_\infty \geq q/12$ for any non-zero $\tilde{s} \in R_q$. This shows that there exists at most one pair $(s, \mathbf{e}) \in R_q \times R^{\bar{m}}$ such that $\|\mathbf{e}\|_\infty < q/24$ and $\mathbf{u} = \mathbf{a}s + \mathbf{e}$. $\square$

A.2 The Construction

Let $n \geq 4$ be a power of 2, and let $q = 3^k$ be a power of 3. Let $R_q = \mathbb{Z}_q[X]/(X^n+1)$. Let integer $\bar{m} \geq 2\log_2 q + c$ for some constant $c > 0$, and $m = \bar{m} + k$. Let $(\mathsf{encode}_d, \mathsf{decode}_d)$ be the pair of encode/decode algorithm parameterized by (n, q, d). Let $\mathbb{F}_{2^\kappa}$ be a finite field of order 2^κ. Let $\mathsf{H} : \{0,1\}^* \to \mathbb{F}_{2^\kappa}\backslash\{0\}$ be a collision-resistant hash function, where $\kappa \leq n/2$. Let $\phi : R \to \mathbb{Z}^n$ be the bijective coefficient embedding, which can naturally map an element $x \in \{0,1\}^\kappa$ as a degree $\kappa - 1$ polynomial $\phi^{-1}(x) \in R$. Our CCA-secure PKE from ring-LWE is given as follows.

- $\mathsf{KeyGen}(1^\kappa)$: randomly choose $\mathbf{a} \xleftarrow{\$} R_q^{\bar{m}}$ and $\mathbf{R} \xleftarrow{\$} (D_{R,\omega(\sqrt{\log n\bar{m}})})^{k \times \bar{m}}$. Then, compute $\mathbf{b} = -\mathbf{R}\mathbf{a}$, and return $(pk, sk) = ((\mathbf{a}, \mathbf{b}), \mathbf{R})$.
- $\mathsf{Enc}(pk, \mu \in \mathbb{F}_{2^\kappa})$: first randomly choose $x, y, z \xleftarrow{\$} \mathbb{F}_{2^\kappa}$, $s \xleftarrow{\$} D_{R,\alpha q}$, $\mathbf{e}_1 \xleftarrow{\$} (D_{R,\alpha q})^{\bar{m}}$, and $\mathbf{e}_2 \xleftarrow{\$} (D_{R,\gamma})^k$ where $\gamma = \alpha q\sqrt{n\bar{m}} \cdot \omega(\sqrt{\log n\bar{m}})$. Then, interpret the bit-concatenation of $(x, y, z) \in (\mathbb{F}_{2^\kappa})^3$ as a vector $\mathbf{v} = x\|y\|z \in \mathbb{Z}_d^n$, and compute

$$\tilde{s} = s + \phi^{-1}(\mathsf{encode}_d(\mathbf{v})), \qquad \mathbf{c}_1 = \mathbf{a}\tilde{s} + \mathbf{e}_1,$$
$$\mathbf{c}_2 = (\mathbf{b} + \mathbf{g}t)\tilde{s} + \mathbf{e}_2, \qquad c_3 = x + \mu,$$
$$c_4 = \tau y + z,$$

where $t = \phi^{-1}(\mathsf{H}(\mathbf{c}_1))$ and $\tau = \mathsf{H}(\mathbf{c}_2, c_3)$. Return the ciphertext $C = (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4) \in R_q^{m_1} \times R_q^k \times \mathbb{F}_{2^\kappa} \times \mathbb{F}_{2^\kappa}$.
- $\mathsf{Dec}(sk, (\mathbf{c}_1, \mathbf{c}_2, c_3, c_4))$: first compute $t = \phi^{-1}(\mathsf{H}(\mathbf{c}_1))$,

$$\mathbf{a}_t = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} + \mathbf{g}t \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix}.$$

Then, compute $\tilde{s} \leftarrow \mathsf{Solve}'(\mathbf{a}_t, \mathbf{R}, \mathbf{u})$, $\mathbf{v} = \mathsf{decode}_d(\tilde{s})$, and parse $\mathbf{v} = x\|y\|z$, where $(x, y, z) \in (\mathbb{F}_{2^\kappa})^3$. Let $\mathbf{e}_1 = \mathbf{c}_1 - \mathbf{a}\tilde{s}$ and $\mathbf{e}_2 = \mathbf{c}_2 - (\mathbf{b}+\mathbf{g}t)\tilde{s}$. If $\|\mathbf{e}_1\| > \alpha q\sqrt{n\bar{m}}$, or $\|\mathbf{e}_2\|_\infty > \gamma \cdot \omega(\sqrt{\log n})$, or $c_4 \neq \mathsf{H}(\mathbf{c}_2, c_3)y + z$, return $\perp$, else return $\mu = c_3 - x \in \mathbb{F}_{2^\kappa}$.

Correctness. Since $\mathbf{c}_1 = \mathbf{a}\tilde{s} + \mathbf{e}_1$ and $\mathbf{c}_2 = (\mathbf{b} + \mathbf{g}t)\tilde{s} + \mathbf{e}_2$, the algorithm $\mathsf{Solve}'(\mathbf{a}_t, \mathbf{R}, \mathbf{u})$ can correctly recover $s \in R_q$ if $\|\mathbf{R}\mathbf{e}_1 + \mathbf{e}_2\|_\infty < q/6$ by Lemma 26. Because $\tilde{s} = s + \mathsf{encode}_d(\mathbf{v})$, one can correctly recover $\mathbf{v} \in \mathbb{Z}_d^n$ if $\|s\|_\infty < \frac{q-(d-1)d}{2d}$ by Lemma 7. Since $s \xleftarrow{\$} D_{R,\alpha q}$, $\mathbf{e}_1 \xleftarrow{\$} (D_{R,\alpha q})^{\bar{m}}$ and $\mathbf{e}_2 \xleftarrow{\$} (D_{R,\gamma})^k$, we have that $\|s\|_\infty \leq \alpha q \cdot \omega(\sqrt{\log n})$, $\|\mathbf{e}_1\| \leq \alpha q \cdot \sqrt{n\bar{m}}$ and $\|\mathbf{e}_2\|_\infty \leq \gamma \cdot \omega(\sqrt{\log n})$ hold with overwhelming probability by Lemma 1. Using the fact that $\mathbf{R} \xleftarrow{\$} (D_{R,\omega(\sqrt{\log n\bar{m}})})^{k \times \bar{m}}$, we have that $\|\mathbf{R}\mathbf{e}_1 + \mathbf{e}_2\|_\infty \leq \alpha q \sqrt{n\bar{m}} \cdot \omega(\sqrt{\log n\bar{m}})^2$ holds with overwhelming probability. Besides, for the worst-case hardness of the ring-LWE problem, we need $\alpha q \geq n^{3/4} \cdot (\bar{m}/\log(n\bar{m}))^{1/4} \cdot \omega(1)$. We will also use Lemma 21 and Lemma 27 in the security proof, which require $\bar{m} \geq 2\log_2 q + c$ for some constant $c > 0$ and $\|\mathbf{e}_1\|_\infty < q/24$.

In all, for both the correctness and the security, it suffices to set the parameters $\alpha, \bar{m}, q$ such that :

$$\begin{aligned}
\bar{m} &= 2\log_2 q + c, & 1/\alpha &= \sqrt{n\bar{m}} \cdot \omega(\sqrt{\log n\bar{m}})^2, \\
\alpha q &= n^{3/4} \cdot (\bar{m}/\log(n\bar{m}))^{1/4} \cdot \omega(1),
\end{aligned} \tag{3}$$

which means that $\bar{m} = O(\log n), 1/\alpha = \tilde{O}(\sqrt{n})$ and $q = \tilde{O}(n^{1.25})$. As before, one may prefer to consider the concrete hardness against known attacks when choosing the parameters, and put aside the constraint $\alpha q \geq n^{3/4} \cdot (\bar{m}/\log(n\bar{m}))^{1/4} \cdot \omega(1)$ used for the worst-case reduction [44,50]. For example, one can set $n = 512, \bar{m} = 36, q = 3^{11}, \alpha q = 4.0$ for a decryption error rate less than 2^{-81}, and a security level about 149-bit given by the online LWE estimator [4].[1] In this setting, the public key and secret key sizes are about $nm\log q \approx 52.88$ KB (or 12.38 KB when all users share the first element $\mathbf{a}$ of the public key) and $\bar{m}nk(\log_2(\alpha q \cdot \omega(\sqrt{\log n})) + 1) \approx 148.5$ KB (or 32 KB when using a PRG to sample the secret key). For 128-bit security, we set $\kappa = 256$, the ciphertext for a 256-bit message is about $nm\log q + 512$ bits ≈ 52.94 KB. This parameter also allows us to compress $\mathbf{c}_2$ with a modulus as small as 16 (i.e., using $\mathsf{Switch}_{q,16}(\mathbf{c}_2)$ before applying the hash function $\tau = \mathsf{H}(\mathbf{c}_2, c_3)$ as discussed in Section 4), and reduce the ciphertext size to 44 KB.

Security. As for security, we have the following theorem.

Theorem 2. *Let integers $n, \bar{m}, q > 0$ and real $\alpha > 0$ satisfy Equation (3). If $\mathrm{RLWE}_{\bar{m},q,\alpha}$ is hard and H is a collision-resistant hash function, then the above PKE scheme is CCA2-secure in the standard model.*

Proof. Given the ring-version Lemma $20 \sim 27$, the theorem can be easily proven by adapting the proof of Theorem 1 to the ring setting. We omit the details. $\square$

---

[1] Note that the best known lattice algorithms do not make use of the ring structure [6].