Secure Hybrid Encryption In the Standard Model from Hard Learning Problems

Xavier Boyen[1], Malika Izabachène[2], Qinyi Li[3]

[1] QUT, Brisbane, Australia
[2] Cosmian, Paris, France
[3] Griffith University, Brisbane, Australia

Abstract. We present chosen-ciphertext secure hybrid encryption systems in the standard model from the learning with errors problem and low-noise learning parity with noise problem. The systems consist of public-key key encapsulation mechanisms that are not chosen-ciphertext secure. The systems are more efficient than the existing chosen-ciphertext secure hybrid encryption systems in the standard model based on the same hard learning problems.

Keywords: Hybrid encryption, CCA security, Standard model

1 Introduction

When encrypting large messages, public-key encryption (PKE) is mostly used as a key encapsulation mechanism (KEM) with a secret-key data encapsulation mechanism (DEM) to build hybrid encryption (HE) system. In a HE system, the KEM produces secret session keys for the DEM to encrypt the actual messages. The HE paradigm was proposed in [9], and it is known that if the KEM and the DEM are both adaptive-chosen ciphertext secure (CCA secure), then the HE system is CCA secure. Since CCA-secure DEM is relatively easy to obtain, the natural step of constructing a CCA-secure HE is to build a CCA-secure KEM or PKE. For example, starting from a weakly secure (i.e., chosen-plaintext secure or even oneway) PKE system, a CCA-secure KEM can be built in the random oracle model (ROM) (or the quantum random oracle model (QROM)) by applying to the Fujisaki-Okamoto transformation [13], or its variants, e.g., [15]. On the other hand, CCA-secure KEM is not necessary for a CCA-secure HE system. One of the prominent examples is the Kurosawa-Desmedt HE system [18], which is CCA secure, but its KEM is not ([2,16] showed that the KEM meets weaker security than the CCA security). CCA-secure HE systems with weakly secure KEMs are often more efficient than the CCA-secure HE systems with CCA-secure KEMs. For instance, the Kurosawa-Desmedt system is more efficient than the Cramer-Shoup system [8]. However, to the best of our knowledge, CCA-secure HE systems with similar feature (i.e., non-CCA-secure KEMs) under the post-quantum assumptions remain unknown.

1.1 Our Contributions

We present a post-quantum HE system in the standard model from the learning with errors (LWE) problem. The HE system is CCA secure, but its KEM is not, which gives the first post-quantum CCA-secure HE system with non-CCA-secure KEM. Our HE system's KEM is a non-adaptive CCA-secure PKE system provided in [19]. The DEM part constitutes a message authentication code (MAC) and a weakly secure secret-key encryption (SKE) system. Due to the KEM's simplicity, the HE system is more efficient than the existing HE systems from lattices in the standard model.

Our technique extends to a CCA-secure HE system from low-noise LPN, based on the Kiltz et al.'s LPN double trapdoor [17]. However, by avoiding the generic transformations, our HE system is more efficient than the HE system obtained from [17]. Our technique also gives simple CCA-secure KEMs from LWE (and low-noise LPN). This is done by combining the non-CCA-secure KEMs with a one-time secure MAC.

It is worth mentioning that our LWE-based HE system and KEM system can be adapted straightforwardly to the ring LWE setting) to be made more space-efficient, since their basis is the trapdoor function from [19], which has ring versions (also supported by practical implementations, e.g., [5,12]). This is not known to be the case for some previous constructions (e.g., [7]). Fig. 1 and

LWE constructions	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{ct}\rvert$	Enc. Time	Dec. Time	Ring?
[19]+[6]+CPA-DEM	$2nm\log q$	$2nm\log q + \lvert\mathsf{tag}\rvert + \lvert\mathsf{com}\rvert + \lvert\phi\rvert$	$t + t_{\mathsf{com}}$	$t' + t_{\mathsf{decom}}$	YES
[7]+CCA-DEM	$3nm\log q$	$2nm\log q + \lvert\mathsf{tag}\rvert + \lambda + \lvert\phi\rvert$	$t + t_{\mathsf{sis}}$	$t' + t_{\mathsf{sis}}$	Unknown
This work	$2nm\log q$	$2m\log q + \lvert\mathsf{tag}\rvert + \lvert\phi\rvert$	t	t'	YES

Table 1: Comparison among LWE-based HE systems

Fig. 2 summarise comparisons among post-quantum HE systems in the standard model. We compare our LWE-based HE system with two HE systems. The first one is the combination of the CCA1 PKE system in [19], Boneh-Katz transformation [6], and a chosen-plaintext secure secret-key encryption system as the DEM (we note that the MAC in the transformation is also used to authenticate the whole ciphertext, which allows avoiding a CCA-secure DEM). The other one is the combination of the CCA-secure KEM from [7] and a CCA-secure KEM (which can be built by using a chosen-plaintext secure secret-key encryption and a MAC). We also compare our LPN-based HE system with the HE system derived from applying Boneh-Katz transformation to the tag-based encryption system from [17] and a secret-key encryption system.

The comparisons assume the HE systems using the same LWE/LPN parameters (e.g., dimensions, noise ratios) and consider the KEM part encapsulates $\lambda \leq n$-bit session keys. $|\mathsf{tag}|$ denotes the size of tags output by the MAC (either for Boneh-Katz transformation or for CCA-secure DEM). $|\mathsf{com}|$ denotes the size of commitment from the commitment scheme used by Boneh-Katz transformation. $|\phi|$ denotes the size of encryption of messages produced by the secret-key encryption systems. We use t and τ (resp. t' and τ') to denote the encryption

LPN constructions	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{ct}\rvert$	Enc. Time	Dec. Time
[17]+[6] +CPA-DEM	$n(3m+\lambda)$	$3m+\lambda+\lvert\mathsf{tag}\rvert+\lvert\mathsf{com}\rvert+\lvert\phi\rvert$	$\tau+t_{\mathsf{com}}$	$\tau'+t_{\mathsf{decom}}$
This work	$n(3m+\lambda)$	$3m+\lambda+\lvert\mathsf{tag}\rvert+\lvert\phi\rvert$	τ	τ'

Table 2: Comparison between LPN-based HE systems

and decryption time of our LWE-based construction (resp. LPN-based construction). t_{com} and t_{decom} denote the time used to compute the commitments and decommitment in the Boneh-Katz transformation (using the commitment scheme recommended in [6], computing the commitment and the commitment contains computing a collision-resistant hash and evaluating a universal hash function). t_{sis} denotes the time of computing the function $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$ where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{x} \in \mathbb{Z}^m$. We also consider whether the system can be adapted to the ring LWE setting. Enabling realisation over rings may help to improve the space efficiency of the systems. The comparisons show our HE systems are either more efficient or more flexible (i.e., support ring LWE). We note that our HE system and the HE system based on Boneh-Katz transformation lose a factor of the number of decryption queries, in security reduction, i.e., it is not tight, while the construction obtained from [7] loses a constant factor.

1.2 Our Approach

We provide the intuition of our approach. The ciphertext of the KEM part of our LWE-based HE system contains LWE samples $\mathbf{c}_0$, $\mathbf{c}_1$ where

$$[\mathbf{c}_0^{*\mathsf{T}}|\mathbf{c}_1^{*\mathsf{T}}] = \mathbf{s}^{\mathsf{T}}[\mathbf{A}|\mathbf{A}_1 + H(\mathbf{c}_0^*)\mathbf{G}] + [\mathbf{e}_0^{\mathsf{T}}|\mathbf{e}_1^{\mathsf{T}}]$$

where $\mathbf{A}$, $\mathbf{A}_1 = \mathbf{A}\mathbf{R}$ are public matrices, $\mathbf{G}$ is the gadget matrix [19], H is a collision resistant hash function with full-rank difference property (see [3]). The session key $\mathbf{k}^*$ for the DEM, a random bit string, is embedded into the LWE secret vector $\mathbf{s}$, e.g., by $\mathbf{s} = \tilde{\mathbf{s}} + \mathbf{k}^*\lfloor q/2\rfloor$. Using the trapdoor $\mathbf{R}$, the private key, $\mathbf{k}^*$ can be recovered. The KEM is not CCA secure: Adding a small vector to $\mathbf{e}_1$ results in a correct ciphertext of $\mathbf{k}^*$. In addition to the KEM, our DEM uses a secret-key encryption (SKE) system to encrypt the actual message M and uses an authentication code (MAC) to authenticate the ciphertext:

$$\phi^* \leftarrow \mathsf{SKE.Enc}(dk, M) \quad \text{and} \quad \sigma^* \leftarrow \mathsf{MAC.Sign}(mk, \mathbf{c}_0||\mathbf{c}_1||\phi)$$

where MAC key mk and SKE key dk are derived from $\mathbf{k}^*$ via a key derivation function (KDF). In a nutshell, the construction is reminiscent of the Boneh-Katz transformation [6] that turns any selectively secure identity-based encryption into a CCA-secure PKE using a MAC and a commitment scheme. $\mathbf{c}_0^*$ constitutes LWE samples and, thus, can be seen as a statistically binding (by that the LWE problem has unique solutions) and computationally hiding (by the LWE assumption) commitment of the session key $\mathbf{k}^*$. The intuition of security, given

the ciphertext $\mathbf{c}_0^*, \mathbf{c}_1^*, \phi^*, \sigma^*$, is that (1) any decryption query with $\mathbf{c}_0 \neq \mathbf{c}_0^*$ is not helpful as the security reduction will use the technique from [3] to embed $\mathbf{c}_0^*$ into the $\mathbf{A}_1$, i.e., $\mathbf{A}_1 = \mathbf{AR} - H(\mathbf{c}_0^*)\mathbf{G}$, so it can decrypt using the trapdoor $\mathbf{R}$, and (2) if the adversary makes a decryption query with $\mathbf{c}_0 = \mathbf{c}_0^*$, since $\mathbf{c}_0^*$ uniquely determines $\mathbf{k}^*$ by the binding property of LWE, the adversary has to know $\mathbf{k}^*$ to forge the MAC or break the security of the SKE. However, $\mathbf{k}^*$ is hidden by LWE assumption.

2 Preliminaries

We use symbol "$\intercal$" for matrix/vector transpose, e.g., $\mathbf{A}^\intercal$ means the transpose of $\mathbf{A}$. We denote by $x \leftarrow X$ the process of sampling x according to the distribution X. Let $x \sim X$ denote sample x satisfies distribution X. We use $U(X)$ to denote the uniform distribution over the set X. We will be using standard asymptotic notations, e.g., O, Ω, ω. Let $\lambda \in \mathbb{N}$, the function $f : \mathbb{N} \to \mathbb{R}$ is said to be negligible if $f(\lambda) = \lambda^{-\omega(1)}$ and is written as $f(\lambda) = \mathsf{negl}(\lambda)$.

Let X and Y be two random variables over some finite set S. The statistical distance between X and Y is defined as $\Delta(X, Y) = \frac{1}{2}\sum_{s \in S} |\Pr[X = s] - \Pr[Y = s]|$. Let X_λ and Y_λ be ensembles of random variables indexed by the security parameter λ. We say that X and Y are $\mathsf{negl}(\lambda)$-statistically close (or simply statistically close) if $\Delta(X_\lambda, Y_\lambda) = \mathsf{negl}(\lambda)$. We use the following lemma in our security proofs.

Lemma 1 (Special case of Lemma 4.4 of [20]). *For $\mathbf{x} \leftarrow D_{\mathbb{Z}^m, s}$, $\Pr[\|\mathbf{x}\| > s\sqrt{m}] < 1 - 2^{-\Omega(m)}$.*

Lemma 2 (Proposition 5.1 of [14]). *Let $q \geq 2$. For all but a $2q^{-n}$ fraction of all $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and for any $s \geq \omega(\sqrt{\log n})$, the distribution of $\mathbf{Ae} \bmod q$ is statistically close to uniform over $\mathbb{Z}_q^n$, where $\mathbf{e} \sim D_{\mathbb{Z}^m, s}$.*

We will use the super-increasing vector $\mathbf{g}^\intercal = (1, 2, 4, \ldots, 2^{k-1})$, for $k = \lceil \log_2 q \rceil$ and extend it to form a "gadget" matrix $\mathbf{G} = diag(\mathbf{g}^\intercal, \ldots, \mathbf{g}^\intercal) \in \mathbb{Z}_q^{n \times nk}$ as in [19]. Here we use a base 2 but other choices of base can be used. We formulate the following lemma which is directly derived from the Theorem 4.1 and Theorem 5.4 and of [19].

Lemma 3. *Let $w = n\lceil \log q \rceil$. Let $\mathbf{F} = [\mathbf{A}|\mathbf{AR} + \mathbf{HG}]$ where $\mathbf{R} \in \mathbb{Z}^{m \times w}$, $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$ is invertible in $\mathbb{Z}_q$, and $\mathbf{G} \in \mathbb{Z}_q^{n \times w}$ is the gadget matrix. Given $\mathbf{b}^\intercal = \mathbf{s}^\intercal\mathbf{F} + \mathbf{e}^\intercal$ where $\mathbf{e}^\intercal = [\mathbf{e}_0^\intercal|\mathbf{e}_1^\intercal]$, there exists an efficient algorithm $\mathsf{Invert}(\mathbf{R}, \mathbf{F}, \mathbf{b})$ that outputs $\mathbf{s}$ and $\mathbf{e}$ when $\|\mathbf{e}_1^\intercal - \mathbf{e}_0^\intercal\mathbf{R}\|_\infty < q/4$.*

Definition 1 ((Normal Form) Learning-With-Errors Problem). *Let λ be the security parameter, $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$ and an error distribution $\chi = \chi(n)$ over $\mathbb{Z}_q$. The advantage of an adversary $\mathcal{A}$ for the (normal-form) $\mathsf{NLWE}_{n,m,q,\chi}$ problem, denoted by $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{NLWE}_{n,m,q,\chi}}(\lambda)$, is defined as*

$$|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{b}^\intercal) = 1]|$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow \chi^n$, $\mathbf{e} \leftarrow \chi^m$. The $\mathsf{NLWE}_{n,m,q,\chi}$ problem is hard if $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{NLWE}_{n,m,q,\chi}}(\lambda) \leq \mathsf{negl}(\lambda)$ for all p.p.t adversary $\mathcal{A}$.

We note the normal-form LWE problem is equivalent to the standard form of the LWE problem. A series of works have established the hardness of the LWE problem. We refer to [23,22] for details.

2.1 Definitions of Cryptographic Primitives

Public-Key Encryption (PKE). A PKE system $\mathsf{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ consists of three algorithms. The probabilistic key generation algorithm $\mathsf{PKE.Gen}(1^\lambda)$ takes as input a security parameter λ, returns a key pair $(\mathsf{pk}, \mathsf{sk})$. The probabilistic encryption algorithm $\mathsf{PKE.Enc}(\mathsf{pk}, M)$ returns a ciphertext ct. The deterministic decryption algorithm $\mathsf{PKE.Dec}(\mathsf{pk}, \mathsf{sk}, \mathsf{ct})$ recovers the message M, or returns $\perp$, indicating decryption fails. The correctness of PKE requires that for all $\lambda \in \mathbb{N}$, all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$,

$$\Pr[\mathsf{PKE.Dec}(\mathsf{pk}, \mathsf{sk}, \mathsf{PKE.Enc}(\mathsf{pk}, M)) = K] \geq 1 - \mathsf{negl}(\lambda)$$

where the probability is over the randomness of $\mathsf{PKE.Gen}$ and $\mathsf{PKE.Enc}$.

Definition 2. *We say* PKE *is chosen-ciphertext-attack secure (CCA-secure) if for all PPT adversary* $\mathcal{A}$*, the advantage function*

$$\mathsf{Adv}^{\mathrm{ind-cca}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = \left|\mathsf{Exp}^{\mathrm{ind-cca}}_{\mathsf{PKE},\mathcal{A}}\lambda) - 1/2\right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Exp}^{\mathrm{ind-cca}}_{\mathsf{PKE},\mathcal{A}}\lambda)$ *is defined in Fig. 1. In the experiment, the adversary is not allowed to query* ct^* *to the oracle* $\mathcal{O}$ *in step 4.*

Experiment $\mathsf{Exp}^{\mathrm{ind-cca}}_{\mathsf{PKE},\mathcal{A}}\lambda)$:

1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$
2. $(M_0, M_1) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pk})$
3. $b \leftarrow U(\{0,1\})$, $\mathsf{ct}^* \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, M_b)$
4. $b' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pk}, \mathsf{ct}^*, \mathsf{pk})$
5. Return 1 if $b' = b$; Otherwise, return 0.

Oracle $\mathcal{O}(\mathsf{ct})$:

1. Return $\mathsf{PKE.Dec}(\mathsf{pk}, \mathsf{sk}, \mathsf{ct})$

Fig. 1: CCA Security Definitions for PKE

Secret-Key Encryption (SKE). A SKE system $\mathsf{SKE} = (\mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ with key space $\mathcal{K}_{\mathsf{ske}}$ and ciphertext space $\mathcal{C}_{\mathsf{ske}}$ (typically $\mathcal{K}_{\mathsf{ske}} = \{0,1\}^\lambda$ for the security λ) consists of two algorithms. The deterministic encryption algorithm $\mathsf{SKE.Enc}(dk, M)$ uses a key $dk \in \mathcal{K}_{\mathsf{ske}}$ to encrypt message M into a ciphertext ϕ. The deterministic decryption algorithm $\mathsf{SKE.Dec}(dk, \phi)$ recovers message M, or return $\perp$, indicating decryption fails. We require that for all $dk \in \mathcal{K}_{\mathsf{ske}}$ and message M, $\mathsf{SKE.Dec}(dk, \mathsf{SKE.Enc}(dk, M)) \rightarrow M$.

Definition 3. *We say a secret-key encryption scheme* SKE *is one-time secure if for all PPT adversary* $\mathcal{A}$ *the advantage function*

$$\mathsf{Adv}^{\mathrm{ot-ind}}_{\mathsf{SKE},\mathcal{A}}(\lambda) = \left|\Pr[\mathsf{Exp}^{\mathrm{ot-ind}}_{\mathsf{SKE},\mathcal{A}}(\lambda) = 1] - 1/2\right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Exp}^{\mathrm{ot-ind}}_{\mathsf{SKE},\mathcal{A}}(\lambda)$ *is defined as in Fig. 2.*

Experiment $\mathsf{Exp}^{\mathrm{ot-ind}}_{\mathsf{SKE},\mathcal{A}}(\lambda)$:

1. $M \leftarrow \mathcal{A}(1^\lambda)$
2. $dk \leftarrow \mathcal{K}_{\mathsf{ske}}$, $b \leftarrow U(\{0,1\})$, $\phi_0 \leftarrow \mathsf{SKE.Enc}(dk, M)$, $\phi_1 \leftarrow \mathcal{C}_{\mathsf{ske}}$
3. $b' \leftarrow \mathcal{A}(1^\lambda, \phi_b)$
4. Return 1 if $b' = b$; Otherwise, return 0

Fig. 2: Security Experiments of SKE

Message Authentication Codes (MACs). In a MAC system $\mathsf{MAC} = (\mathsf{MAC.Sign}, \mathsf{MAC.Ver})$ with key space $\mathcal{K}_{\mathsf{mac}}$ (typically $\mathcal{K}_{\mathsf{mac}} = \{0,1\}^\lambda$ for the security parameter λ), the algorithm $\mathsf{MAC.Sign}(K, x)$ takes as input a key $K \in \mathcal{K}_{\mathsf{mac}}$, a message x and some random coins, and returns a tag σ. The deterministic algorithm $\mathsf{MAC.Ver}(K, \sigma, x)$ returns 1 if $\sigma \leftarrow \mathsf{MAC.Sign}(K, x)$, or outputs 0, otherwise.

Definition 4. *Let* λ *be the security parameter. We say a mac* MAC *is secure with one-time strong unforgeablility if for all PPT adversary* $\mathcal{A}$*, the advantage function*

$$\mathsf{Adv}^{\mathrm{ot-suf}}_{\mathsf{MAC},\mathcal{A}}(\lambda) = \Pr[\mathsf{Exp}^{\mathrm{ot-suf}}_{\mathsf{MAC},\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Exp}^{\mathrm{ot-suf}}_{\mathsf{MAC},\mathcal{A}}(\lambda)$ *is defined as in Fig. 3.*

Experiment $\mathsf{Exp}^{\mathrm{ot-suf}}_{\mathsf{MAC},\mathcal{A}}(\lambda)$:

1. $mk \leftarrow \mathcal{K}_{\mathsf{mac}}$, $x \leftarrow \mathcal{A}(1^\lambda)$
2. $\sigma \leftarrow \mathsf{MAC.Sign}(mk, x)$
3. $(x', \sigma') \leftarrow \mathcal{A}(1^\lambda, \sigma)$
4. If $\mathsf{MAC.Ver}(K, x', \sigma')$ and $(x, \sigma) \neq (x', \sigma')$, return 1
5. Otherwise, return 0

Fig. 3: Security Experiments of MAC

Collision Resistant Hashing. Let $H : \{0,1\}^* \to \{0,1\}^\ell$ be a hash function (where ℓ is a function of the security parameter).

Definition 5. *We say that H is collision resistant if for all p.p.t algorithms $\mathcal{A}$, the advantage,*

$$\mathsf{Adv}^{\mathrm{coll}}_{\mathcal{A}}(\lambda) = \Pr[\mathcal{A}(1^\lambda, H) \to (x, x') : \ x \neq x \text{ and } H(x) = H(x')] \leq \mathsf{negl}(\lambda)$$

where $x \leftarrow \{0,1\}^*$ and λ is the security parameter.

Key Derivation Functions. Our constructions use key derivation functions (KDFs) to expand short random keys to longer pseudorandom keys for message authentication codes and secret-key encryption. Basically, a KDF is a pseudo-random number generator.

Definition 6. *Let λ be the security parameter. Let $\mathcal{K}$ be a set with size $\{0,1\}^{\geq\lambda}$. We say a key derivation function $\mathsf{KDF} : \mathcal{K} \to \{0,1\}^\ell$ is secure if the advantage function*

$$\mathsf{Adv}^{\mathrm{ind}}_{\mathsf{KDF},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(1^\lambda, \mathsf{KDF}(k))] = 1 - \Pr[\mathcal{A}(1^\lambda, r) = 1]| \leq \mathsf{negl}(\lambda)$$

where $k \leftarrow U(\mathcal{K})$ and $r \leftarrow U(\{0,1\}^\ell)$.

3 CCA-Secure Hybrid Encryption from LWE

The system uses the following public parameters shared by all system instances.

1. We use $\mathsf{NLWE}_{n,m,q,D_{\mathbb{Z},\alpha q}}$ problem for some polynomial-size (in n) prime q. n, q, m, α are determined to ensure $\mathsf{NLWE}_{n,m,q,D_{\mathbb{Z},\alpha q}}$ problem is hard. Let $w = n\lceil \log q \rceil$ and $m \geq n \log q + \omega(\sqrt{\log n})$.
2. A full-rank difference encoding as defined in [3] $\mathsf{FRD} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ that for any $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^n$ with $\mathbf{x} \neq \mathbf{y}$, $\mathsf{FRD}(\mathbf{x}) - \mathsf{FRD}(\mathbf{y})$ is invertible over $\mathbb{Z}_q^{n\times n}$. In particular, $\mathsf{FRD}(\mathbf{x})$ is invertible over $\mathbb{Z}_q^{n\times n}$ if $\mathbf{x} \neq \mathbf{0}$.
3. A collision resistance hash function $H : \{0,1\}^* \to \mathbb{Z}_q^n \backslash \{\mathbf{0}\}$ where $\mathbf{0}$ is the zero matrix in $\mathbb{Z}_q^n$, a secret-key encryption system $\mathsf{SKE} = (\mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ with key space $\mathcal{K}_{\mathsf{ske}}$, message space $\mathcal{M}_{\mathsf{ske}}$, and ciphertext space $\mathcal{C}_{\mathsf{ske}}$, a secure message authentication code $\mathsf{MAC} = (\mathsf{MAC.Sign}, \mathsf{MAC.Ver})$ with key space $\mathcal{K}_{\mathsf{mac}}$. A key derivation function $\mathsf{KDF} : \{0,1\}^n \to \mathcal{K}_{\mathsf{ske}} \times \mathcal{K}_{\mathsf{mac}}$.

- $\mathsf{PKE.Gen}(1^\lambda)$:
  1. $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n\times m})$, $\mathbf{R} \leftarrow D^{m\times w}_{\mathbb{Z},\omega(\sqrt{\log n})}$
  2. $\mathbf{A}_1 \leftarrow \mathbf{AR}$.
  3. $\mathsf{pk} \leftarrow (\mathbf{A}, \mathbf{A}_1)$, $\mathsf{sk} \leftarrow \mathbf{R}$
- $\mathsf{PKE.Enc}(\mathsf{pk}, M)$:
  1. $\mathbf{k} \leftarrow U(\{0,1\}^n)$, $(dk, mk) \leftarrow \mathsf{KDF}(\mathbf{k})$
  2. $\bar{\mathbf{s}} \leftarrow D^n_{\mathbb{Z},\alpha q}$, $\mathbf{s} \leftarrow \mathbf{k}\lfloor q/2 \rfloor + \bar{\mathbf{s}}$.
  3. $\mathbf{e}_0 \leftarrow D^m_{\mathbb{Z},\alpha q}$, $\mathbf{c}_0^\intercal \leftarrow \mathbf{s}^\intercal\mathbf{A} + \mathbf{e}_0^\intercal$.
  4. $\mathbf{e}_1 \leftarrow D^w_{\mathbb{Z},s}$ where $s^2 = (\|\mathbf{e}_0\|^2 + m(\alpha q)^2) \cdot \omega(\sqrt{\log n})^2$.
  5. $\mathbf{c}_1^\intercal \leftarrow \mathbf{s}^\intercal(\mathbf{A}_1 + \mathsf{FRD}(H(\mathbf{c}_0)\mathbf{G}) + \mathbf{e}_1^\intercal$.
  6. $\phi \leftarrow \mathsf{SKE.Enc}(dk, M)$, $\sigma \leftarrow \mathsf{MAC.Sign}(mk, \mathbf{c}_0||\mathbf{c}_1||\phi)$.

7. $\mathsf{ct} \leftarrow (\mathbf{c}_0, \mathbf{c}_1, \phi, \sigma)$.

– $\mathsf{PKE.Dec}(\mathsf{pk}, \mathsf{sk}, \mathsf{ct})$:

1. Parse $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \phi, \sigma)$; Output $\perp$ if ct doesn't parse.
2. Recover $(\mathbf{s}, \mathbf{e}_0, \mathbf{e}_1) \leftarrow \mathsf{Invert}(\mathbf{R}, [\mathbf{A}|\mathbf{A}_1 + \mathsf{FRD}(H(\mathbf{c}_0))\mathbf{G}], [\mathbf{c}_0^\intercal|\mathbf{c}_1^\intercal])$.
3. If $\|\mathbf{e}_0\| > \alpha q\sqrt{m}$ or $\|\mathbf{e}_1\| > \alpha q\sqrt{2mw} \cdot \omega(\sqrt{\log n})$, output $\perp$.
4. Set $\mathbf{k}[i] \leftarrow 0$ if $\mathbf{s}[i]$ is closer to 0 or $\mathbf{k}[i] \leftarrow 1$ if $\mathbf{s}[i]$ is closer to $q/2$.
5. Output $\perp$ and aborts if $\|\mathbf{s} - \mathbf{k}\| > \alpha q\sqrt{n}$; Else, output $\mathbf{k}$ and continue.
6. $(dk, mk) \leftarrow \mathsf{KDF}(\mathbf{k})$.
7. Return $M \leftarrow \mathsf{SKE.Dec}(dk, \phi)$ if $1 \leftarrow \mathsf{MAC.Ver}(mk, \mathbf{c}_0||\mathbf{c}_1||\phi)$, or else $\perp$.

Correctness. We show that by the chosen parameters, given an honestly generated ciphertext $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \phi, \sigma)$, the algorithm $\mathsf{Invert}(\mathbf{R}, [\mathbf{A}|\mathbf{A}_1 + \mathsf{FRD}(\mathbf{c}_0)\mathbf{G}], [\mathbf{c}_0^\intercal|\mathbf{c}_1^\intercal])$ correctly returns the $\mathbf{s}$. The rest part of correctness readily follows from the correctness of SKE and MAC. Recall that $[\mathbf{c}_0^\intercal|\mathbf{c}_1^\intercal] = \mathbf{s}^\intercal[\mathbf{A}|\mathbf{A}_1 + \mathsf{FRD}(H(\mathbf{c}_0))\mathbf{G}] + [\mathbf{e}_0^\intercal|\mathbf{e}_1^\intercal]$ where $\mathbf{e}_0 \sim D^m_{\mathbb{Z},\alpha q}$ and $\mathbf{e}_1 \sim D^w_{\mathbb{Z},s}$ where $s^2 = (\|\mathbf{e}_0\|^2 + m(\alpha q)^2) \cdot \omega(\sqrt{\log n})^2$. According to Lemma 1, with overwhelming probability, $\|\mathbf{e}_0\| \leq \alpha q\sqrt{m}$ and $\|\mathbf{e}_1\| \leq s\sqrt{w}$. So, the error term is bounded

$$\begin{aligned}\|\mathbf{e}_1^\intercal - \mathbf{e}_0^\intercal\mathbf{R}\|_\infty \leq \|\mathbf{e}_1^\intercal - \mathbf{e}_0^\intercal\mathbf{R}\| \leq \|\mathbf{e}_1^\intercal\| + \|\mathbf{e}_0^\intercal\mathbf{R}\| \quad &\leq 2\alpha q \cdot O(\sqrt{w}) \cdot \omega(\sqrt{\log n}) \cdot \sqrt{3w} \\ &= q \cdot \alpha \cdot O(w) \cdot \omega(\sqrt{\log n})\end{aligned}$$

with overwhelming probability. For large enough $1/\alpha = O(w) \cdot \omega(\sqrt{\log n})$, the error term is smaller than $q/4$ as required by Lemma 3. With $\mathbf{s} = \mathbf{k}\lfloor q/2 \rfloor + \bar{\mathbf{s}}$, $\mathbf{k}$ can be recovered with overwhelming probability since $\|\bar{\mathbf{s}}\| \leq \alpha q\sqrt{n} < q/4$ with overwhelming probability.

3.1 Security Proof

Theorem 1. *Under the assumptions that the problem* $\mathsf{NLWE}_{n,m,q,D_{\mathbb{Z},\alpha q}}$ *is hard,* MAC, SKE *are secure w.r.t Definition 4 and 3, respectively, and* H *is collision resistant w.r.t Definition 5, the hybrid encryption system* PKE *is CCA secure w.r.t to Definition 2.*

Proof. Let λ be the security parameter. Let $\mathcal{A}$ be any PPT adversary who has advantage $\mathsf{Adv}^{\mathrm{ind-cca}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ against the proposed public-key (hybrid) encryption scheme. We show how to bound $\mathsf{Adv}^{\mathrm{ind-cca}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ by the hardness of the $\mathsf{NLWE}_{n,m,q,D_{\mathbb{Z},\alpha q}}$ problem, the one-time security of MAC and SKE, and the collision resistance of H.

We proceed with a sequence of security games. Fig. 4 describes how pk, sk, and the challenge ciphertext ct^* are generated, and Fig. 5 describes how decryption queries are responded in the security games. We denote by E_i that some event E happens in Game i. Each security game eventually outputs a binary value. We denote by S_i the event that Game i outputs 1 (which means that the adversary wins the chosen-ciphertext security game). Throughout the proof, we say a ciphertext is *valid* if it can be properly decrypted to some message.

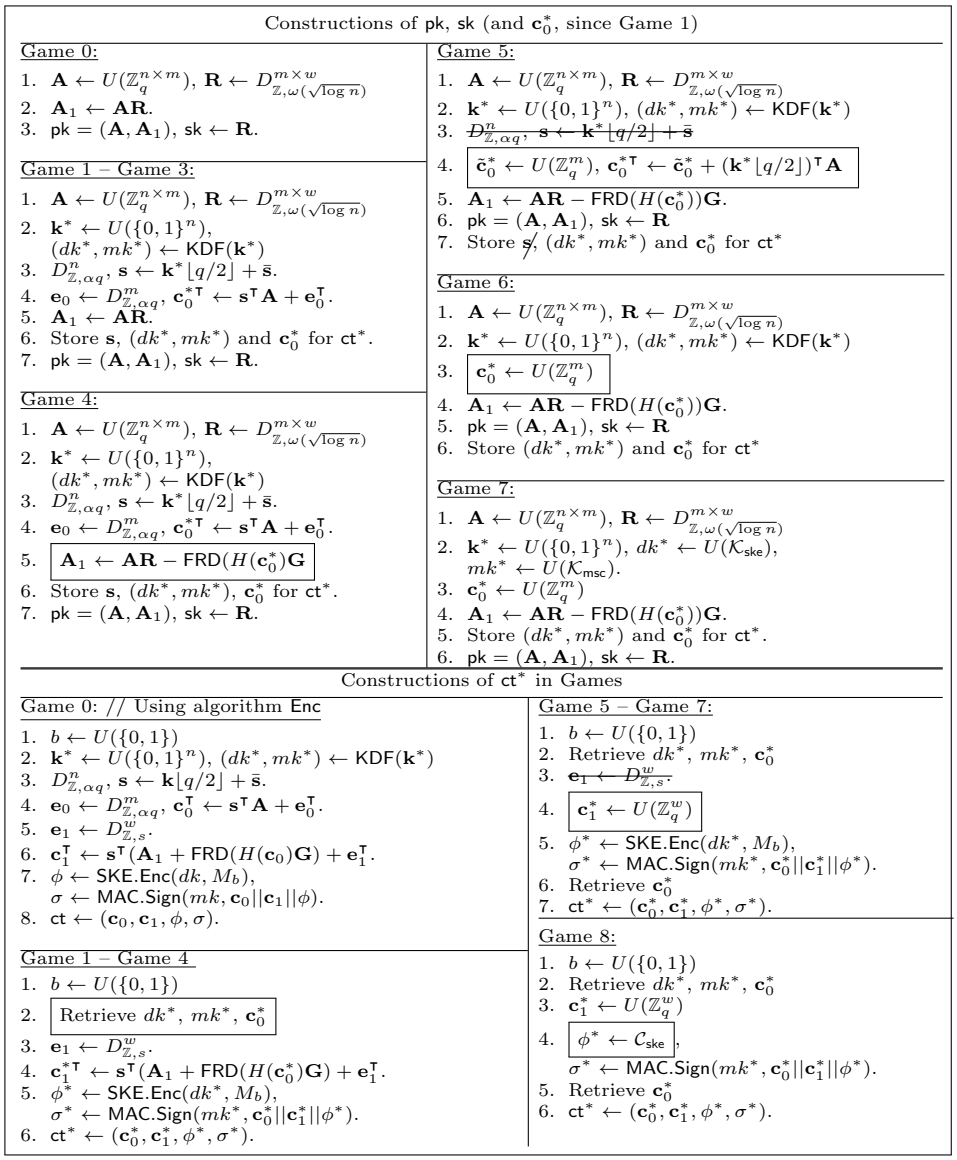

Fig. 4: Generating Keys (and $\mathbf{c}_0^*$) and ct^* in Games

Descriptions of decryption oracle $\mathcal{O}$ in Games

Game 0 – Game 1: // Real decryption algorithm Dec

1. Parse $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \phi, \sigma)$; Output $\perp$ if ct doesn't parse.
2. Recover $(\mathbf{s}, \mathbf{e}_0, \mathbf{e}_1) \leftarrow \mathsf{Invert}(\mathbf{R}, [\mathbf{A}|\mathbf{A}_1 + \mathsf{FRD}(H(\mathbf{c}_0))\mathbf{G}], [\mathbf{c}_0^\intercal|\mathbf{c}_1^\intercal])$.
3. If $\|\mathbf{e}_0\| > \alpha q\sqrt{m}$ or $\|\mathbf{e}_1\| > \alpha q\sqrt{2mw} \cdot \omega(\sqrt{\log n})$, output $\perp$.
4. Set $\mathbf{k}[i] \leftarrow 0$ if $\mathbf{s}[i]$ is closer to 0 or $\mathbf{k}[i] \leftarrow 1$ if $\mathbf{s}[i]$ is closer to $q/2$.
5. Output $\mathbf{k}$ if $\|\mathbf{s} - \mathbf{k}\| \leq \alpha q\sqrt{n}$; Otherwise output $\perp$.
6. $(dk, mk) \leftarrow \mathsf{KDF}(\mathbf{k})$.
7. Return $M \leftarrow \mathsf{SKE.Dec}(dk, \phi)$ if $1 \leftarrow \mathsf{MAC.Ver}(mk, \mathbf{c}_0||\mathbf{c}_1||\phi)$, or else $\perp$.

Game 2:

1. Parse $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \phi, \sigma)$; Output $\perp$ if ct doesn't parse.
2. Return $\perp$ if
   - $\boxed{\mathsf{ct}^* \text{ is not released and } H(\mathbf{c}_0) = H(\mathbf{c}_0^*).}$
   - $\boxed{\mathsf{ct}^* \text{ has been released, and } H(\mathbf{c}_0) = H(\mathbf{c}_0^*) \text{ where } \mathbf{c}_0 \neq \mathbf{c}_0^*.}$
3. (Same as Game 1, step 2 to Step 7)

Game 3 – Game 7:

1. Parse $\mathsf{ct} = (\mathbf{c}_0), \mathbf{c}_1, \phi, \sigma)$; Output $\perp$ if ct doesn't parse.
2. Return $\perp$ if
   - ct^* is not released and $H(\mathbf{c}_0) = H(\mathbf{c}_0^*)$.
   - ct^* has been released, and $H(\mathbf{c}_0) = H(\mathbf{c}_0^*)$ where $\mathbf{c}_0 \neq \mathbf{c}_0^*$.
3. $\boxed{\text{Return } \perp \text{ if the query } \mathsf{ct} \text{ is made after seeing } \mathsf{ct}^* \text{ where } c_0 = \mathbf{c}_0^*.}$[a]
4. (Same as Game 1, step 2 to Step 7)

[a] Even ct is a valid ciphertext.

Fig. 5: Descriptions of decryption oracle $\mathcal{O}$ in Games

Game 0. The first game, Game 0, follows the security experiment $\mathsf{Exp}^{\text{ind-cca}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$. That is, $\mathcal{A}$ is given a public key pk and starts making decryption queries to $\mathcal{O}$. The decryption queries are answered by $\mathsf{PKE.Dec}$. After that, $\mathcal{A}$ submits two messages M_0, M_1 of equal length. The challenge ciphertext $\mathsf{ct}^* = (\mathbf{c}_0^*, \mathbf{c}_1^*, \phi^*, \sigma^*)$ is constructed by $\mathsf{PKE.Enc}(\mathsf{pk}, M_b)$ for $b \leftarrow U(\{0,1\})$, and sent back to $\mathcal{A}$. Concretely, ct^* is computed as

$$\mathbf{c}_0^{*\intercal} = \mathbf{s}^{*\intercal}\mathbf{A} + \mathbf{e}_0^\intercal \quad ; \quad \mathbf{c}_1^{*\intercal} = \mathbf{s}^{*\intercal}(\mathbf{A}_1 + \mathsf{FRD}(H(\mathbf{c}_0^*))\mathbf{G}) + \mathbf{e}_1^\intercal$$

$$\phi^* = \mathsf{SKE.Enc}(dk^*, M_b) \quad ; \quad \sigma = \mathsf{MAC.Sign}(mk^*, \mathbf{c}_0^*||\mathbf{c}_1^*||\phi^*)$$

where $\mathbf{k}^* \leftarrow U(\{0,1\})$, $\mathbf{s} = \tilde{\mathbf{s}} + \mathbf{k}^*\lfloor q/2 \rfloor$ for $\tilde{\mathbf{s}} \leftarrow D^n_{\mathbb{Z},\alpha q}$, and $(dk^*, mk^*) \leftarrow \mathsf{KDF}(\mathbf{k}^*)$. $\mathcal{A}$ then continues making decryption queries ct to the oracle $\mathcal{O}$ with the restriction that $\mathsf{ct} \neq \mathsf{ct}^*$. The decryption queries $\mathsf{ct} \neq \mathsf{ct}^*$ to $\mathcal{O}$ are responded with $\mathsf{PKE.Dec}(\mathsf{pk}, \mathsf{sk}, \mathsf{ct})$. Finally, $\mathcal{A}$ outputs b'. The game returns 1 if $b' = b$, or 0, otherwise. By definition, we have

$$\Pr[S_0] = \Pr[\mathsf{Exp}^{\text{ind-cca}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = 1] \tag{1}$$

Game 1. Game 1 modifies Game 0 on the ways that pk, sk, and challenge ciphertext ct^* are generated, as specified in Fig. 4. Decryption queries in this game are responded to as in Game 0. It can be seen that the modification does not change $\mathcal{A}$'s view: the precomputed $\mathbf{k}^*$ and $\mathbf{c}_0^*$ are distributed as they are in Game 0.

They are independent of pk and unavailable to $\mathcal{A}$ until ct^* gets released. So,

$$\Pr[S_1] = \Pr[S_0]. \tag{2}$$

Game 2. As detailed in Fig. 5, Game 2 is identical to Game 1 except for decryption oracle $\mathcal{O}$, the two types of decryption queries are rejected with returning $\perp$ regardless whether they are valid ciphertexts: (1) $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \phi, \sigma)$ with $H(\mathbf{c}_0) = H(\mathbf{c}_0^*)$ made before seeing ct^* (where $\mathbf{c}_0$ can be the same as or different from $\mathbf{c}_0^*$); and (2) $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \phi, \sigma)$ with $\mathbf{c}_0 \neq \mathbf{c}_0^*$ and $H(\mathbf{c}_0) = H(\mathbf{c}_0^*)$. We prove the following Lemma.

Lemma 4. *Let Q_1 (resp. Q_2) be the maximun number of decryption queries made before (resp. after) seeing ct^*. We have for some adversary $\mathcal{B}_1$ against H.*

$$|\Pr[S_2] - \Pr[S_1]| \leq \frac{Q_1}{q^m} + Q_2 \cdot \mathsf{Adv}^{\mathrm{coll}}_{H,\mathcal{B}_1}(\lambda) \tag{3}$$

Proof. Assume Q_1, Q_2 be the number of decryption queries (both are polynomials in λ) that the adversary can make in the first decryption query phase (i.e., before seeing ct^*) and the second decryption query phase (i.e., after seeing ct^*), respectively. First of all, recall that Game 2 pre-computes $\mathbf{c}_0^*$ and releases it as a part of the challenge ciphertext ct^*, after the first decryption query phase is over. Therefore, $\mathbf{c}_0^* \in \mathbb{Z}_q^m$ is independent of $\mathcal{A}$'s view during the first decryption phase, and, $\mathcal{A}$ makes type-(1) query (in the first decryption query phase) with at most Q_1/q^m.

Next, we construct an adversary $\mathcal{B}_1$ to break the collision resistance of H if the type-(2) query is very made. $calB_1$ receives the security parameter λ and a collision-resistance hash function $H : \mathbb{Z}_q^m \to \mathbb{Z}_q^n \setminus \{\mathbf{0}\}$. It runs follows the process specified in Fig. 4 (for Game 1, which is same for Game 2) to produce pk, sk and $\mathsf{ct}^* = (\mathbf{c}_0^*, \mathbf{c}_1^*, \phi^*, \sigma^*)$. It also follows Fig. 5 to respond decryption queries made before the release of ct^*. Whenever the adversary $\mathcal{A}$ makes a type-(2) decryption query $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \phi, \sigma)$, such that $\mathbf{c}_0 \neq \mathbf{c}_0^*$ and $H(\mathbf{c}_0) = H(\mathbf{c}_0^*)$, $\mathcal{B}$ aborts the Game and return $(\mathbf{c}_0, \mathbf{c}_0^*)$ as a collision for H. Since $\mathcal{A}$ can make at most Q_2 decryption queries after seeing the challenge ciphertext, we have $\mathcal{A}$ makes a type-(2) query with probability at most $Q_2 \cdot \mathsf{Adv}^{\mathrm{coll}}_{H,\mathcal{B}_1}(\lambda)$. We conclude the proof by the Difference Lemma, i.e., [24], Lemma 1. □

Looking ahead, after Game 2, any decryption query in which $\mathbf{c}_0 = \mathbf{c}_0^*$ in the pre-challenge decryption query phase is rejected, and any decryption query with $H(\mathbf{c}_0) \neq H(\mathbf{c}_0^*)$ after the challenge ciphertext has released are rejected.

Game 3. Game 3 is identical to Game 2 except that the decryption oracle $\mathcal{O}$ is implemented slightly differently. Let $\mathsf{ct}^* = (\mathbf{c}_0^*, \mathbf{c}_1^*, \phi^*, \sigma^*)$ be the challenge ciphertext in which the session key $\mathbf{k}^* \in \{0,1\}^n$ is encapsulated in $\mathbf{c}_0^*$ for generating dk^* and mk^*. We focus on the decryption queries of the form $\mathsf{ct} = (\mathbf{c}_0^*, \mathbf{c}_1, \phi, \sigma)$ that are issued after the challenge ciphertext ct^* is released. In Game 3, the oracle rejects (by returning $\perp$) such decryption queries. Fig. 5

specifies in more details how the queries are simulated in Game 3.[4] We define Valid the event that $\mathcal{A}$ submits a valid decryption query $\mathsf{ct} = (\mathbf{c}_0^*, \mathbf{c}_1, \phi, \sigma)$ ($\mathbf{c}_1$, ϕ, and σ are arbitrary) after seeing ct^*. In Game 3, we cannot determine exactly the probability of the event Valid, i.e., $\Pr[\mathsf{Valid}_3]$. However, we can bound it considering two sub-events:

- NoBind: a session key $\mathbf{k} \neq \mathbf{k}^*$ is associated to $\mathbf{c}_0^*$, i.e. $\mathbf{c}_0^* = (\tilde{\mathbf{s}} + \mathbf{k}\lfloor q/2 \rfloor)^\intercal \mathbf{A} + \mathbf{e}_0^\intercal$ for some $\tilde{\mathbf{s}} \sim D_{\mathbb{Z},\alpha q}^n$ and $\mathbf{e}_0 \sim D_{\mathbb{Z},\alpha q}^m$.
- Forge: the session key $\mathbf{k}^*$ associated to ct^* has also been used for ct. And since the query is valid, we have that $1 \leftarrow \mathsf{MAC.Ver}(dk^*, \sigma, \mathbf{c}_0^* || \mathbf{c}_1 || \phi)$ where $(dk^*, mk^*) \leftarrow \mathsf{KDF}(\mathbf{k}^*)$.

Note that we have $\Pr[\mathsf{Valid}_3] \leq \Pr[\mathsf{NoBind}_3] + \Pr[\mathsf{Forge}_3]$. It is the fact that over the random choice of matrix $\mathbf{A}$ and $\mathbf{e}_0$, $\mathbf{c}_0^{*\intercal} = \mathbf{s}^\intercal \mathbf{A} + \mathbf{e}_0^\intercal$ uniquely determines $\mathbf{s}$, and, thus $\mathbf{k}^*$, (see e.g., [25], Lemma 6). Therefore, $\Pr[\mathsf{NoBind}_3]$ is negligible. Meanwhile, we can see that Game 3 and Game 2 are identical until Valid occurs. Hence, we have

$$|\Pr[S_3] - \Pr[S_2]| \leq \Pr[\mathsf{Valid}_3] \leq \Pr[\mathsf{Forge}_3] + \mathsf{negl}(\lambda) \tag{4}$$

for some negligible function $\mathsf{negl}(\lambda)$.

Game 4. Game 4 is identical to Game 3 except it slightly modifies the construction of matrix $\mathbf{A}_1$ when defining pk, as specified in Fig. 4. Notice that the distributions of $\mathbf{A}_1$ in Game 4 ($\mathbf{A}_1 = \mathbf{AR} - \mathsf{FRD}(H(\mathbf{c}_0^*))\mathbf{G}$) and Game 3 ($\mathbf{A}_1 = \mathbf{AR}$) are both statistically close to $U(\mathbb{Z}_q^{n \times w})$. So, the distributions of pk in Game 4 and Game 3 are statistically close. The decryption oracle in Game 4 can handle the same set of decryption queries as Game 3, except for decryption query of the form $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \phi, \sigma)$ where $H(\mathbf{c}_0) = H(\mathbf{c}_0^*)$. This is because for any such decryption query, $[\mathbf{A}|\mathbf{A}_1 + \mathsf{FRD}(H(\mathbf{c}_0))\mathbf{G}] = [\mathbf{A}|\mathbf{AR} + (\mathsf{FRD}(H(\mathbf{c}_0)) - \mathsf{FRD}(\mathbf{c}_0^*))\mathbf{G}]$ where $\mathsf{FRD}(H(\mathbf{c}_0)) - \mathsf{FRD}(\mathbf{c}_0^*)$ is invertible over $\mathbb{Z}_q^{n \times n}$. Thus, the trapdoor $\mathbf{R}$ can be used, same as in Dec, to recover the encryption randomness $\mathbf{s}, \mathbf{e}_0, \mathbf{e}_1$, and, then the message. However, such a decryption query has already been handled by returning $\perp$ by the implementation of $\mathcal{O}$. Therefore, the modification introduced in Game 4 does not statistically change $\mathcal{A}$'s view, and hence,

$$|\Pr[S_4] - \Pr[S_3]| \leq \mathsf{negl}(\lambda) \quad \text{and} \quad |\Pr[\mathsf{Forge}_4] - \Pr[\mathsf{Forge}_3]| \leq \mathsf{negl}(\lambda) \tag{5}$$

where the function $\mathsf{negl}(\lambda)$ accounts for the negligible statistical errors.

Game 5. In Game 5, we further modify the way that pk and ct^* are generated, as specified by Fig. 4. The decryption queries in Game 5 are responded as in Game 4. In particular, we chose pk with the same distribution of pk in Game 5 is identical to that of pk in Game 4. Note that in Game 4, the ciphertext components $\mathbf{c}_0^*$, $\mathbf{c}_1^*$ are LWE samples while, in Game 5, they are distributed

[4] Note that the decryption queries ct made before the release of ct^* where $H(\mathbf{c}_0) = H(\mathbf{c}_0^*)$ are already responded with $\perp$, and $\mathsf{ct} \neq \mathsf{ct}^*$ as querying ct^* is not allowed.

uniformly random. We can show $\Pr[S_4]$ and $\Pr[S_5]$, $\Pr[\mathsf{Forge}_4]$ and $\Pr[\mathsf{Forge}_5]$ are close under the LWE assumption via Lemma 5.

Lemma 5. *For Game 4 and Game 5 defined as per Fig. 4 and Fig. 5, we have*

$$|\Pr[S_5] - \Pr[S_4]| \leq \mathsf{Adv}^{\mathsf{NLWE}_{n,m,q,D_{\mathbb{Z},\alpha q}}}_{\mathcal{B}_2}(\lambda) \tag{6}$$

$$|\Pr[\mathsf{Forge}_5] - \Pr[\mathsf{Forge}_4]| \leq 2 \cdot \mathsf{Adv}^{\mathsf{NLWE}_{n,m,q,D_{\mathbb{Z},\alpha q}}}_{\mathcal{B}'_2}(\lambda) \tag{7}$$

for some adversary $\mathcal{B}_2$ and $\mathcal{B}'_2$ against the LWE problem.

Proof. We show that $|\Pr[S_5] - \Pr[S_4]$ is negligible if $\mathsf{NLWE}_{n,m,q,D_{\mathbb{Z},\alpha q}}$ assumption holds. We do this by constructing an efficient algorithm $\mathcal{B}_2$ who interacts with the PKE adversary $\mathcal{A}$. $\mathcal{B}_2$ receives an instance of $\mathsf{LWE}_{n,m,q,D_{\mathbb{Z},\alpha q}}$ $(\mathbf{B}, \mathbf{b}^\intercal) \in (\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m)$. It decides that if $\mathbf{b}$ is uniformly random (independent of $\mathbf{B}$) or there exists $\mathbf{x} \sim D^n_{\mathbb{Z},\alpha q}$ and $\mathbf{y} \sim D^m_{\mathbb{Z},\alpha q}$ such that $\mathbf{b}^\intercal = \mathbf{x}^\intercal\mathbf{B} + \mathbf{y}^\intercal$. $\mathcal{B}_2$ works as follows.

- It follows Game 5 (as specified in Fig. 4) to generate pk, sk except that it sets $\mathbf{A} \leftarrow \mathbf{B}$, $\tilde{\mathbf{c}}^*_0 \leftarrow \mathbf{b}$.
- It follows Game 5 (as specified in Fig. 4) to generate ct^* except for $\mathbf{c}^*_1$ (note that $\mathbf{c}^*_0$ has been constructed when generating pk). To construct $\mathbf{c}^*_1$, $\mathcal{B}_2$ samples $\mathbf{v} \leftarrow D^w_{\mathbb{Z},\alpha q\sqrt{m}\cdot\omega(\sqrt{\log n})}$ and sets $\mathbf{c}^{*\intercal}_1 \leftarrow \mathbf{c}^{*\intercal}_0\mathbf{R} + \mathbf{v}^\intercal$.
- It follows Game 5 (as specified in Fig. 5) for answering decryption queries.
- Finally, $\mathcal{B}_2$ outputs 1 when the security game outputs 1 (i.e., $\mathcal{A}$ outputs $b' = b$). Otherwise $\mathcal{B}_2$ outputs 0.

We analyse the reduction. First, recall that the distributions of pk in Game 4 and Game 5 are identical. Second, the same set of decryption queries are handled in Game 4 and Game 5, and the responses of decryption queries in the two games have the same distribution. Moreover, we can see that the distributions of ϕ^* and σ^* are identical in Game 4 and Game 5. Finally, if the challenge $(\mathbf{A}, \mathbf{b})$ is random, the challenge ciphertext ct^* distributes as in Game 5. Hence, we have $\Pr[S_5] = \Pr[\mathcal{B}_2(\mathbf{B}, \mathbf{b}^\intercal) = 1]$.

On the other hand, if $\mathbf{b}^\intercal = \mathbf{x}^\intercal\mathbf{B} + \mathbf{y}^\intercal$, we implicitly set $\tilde{\mathbf{s}} \leftarrow \mathbf{x}$ and $\mathbf{e}_0 \leftarrow \mathbf{y}$, and we have

$$\begin{aligned}\mathbf{c}^{*\intercal}_0 &= \tilde{\mathbf{c}}^*_0 + (\mathbf{k}^*\lfloor q/2\rfloor)^\intercal\mathbf{A} = \tilde{\mathbf{s}}^\intercal\mathbf{A} + \mathbf{e}^\intercal_0 + (\mathbf{k}^*\lfloor q/2\rfloor)^\intercal\mathbf{A}\\ &= \mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal_0\end{aligned}$$

where $\mathbf{s} = \tilde{\mathbf{s}} + (\mathbf{k}^*\lfloor q/2\rfloor)$ and

$$\begin{aligned}\mathbf{c}^{*\intercal}_1 = \mathbf{c}^{*\intercal}_0\mathbf{R} + \mathbf{v}^\intercal &= (\mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal_0)\mathbf{R} + \mathbf{v}^\intercal = \mathbf{s}^\intercal(\mathbf{AR}) + (\mathbf{e}^\intercal_0\mathbf{R} + \mathbf{v}^\intercal)\\ &= \mathbf{s}^\intercal(\mathbf{A}_1 + \mathsf{FRD}(H(\mathbf{c}^*_0))\mathbf{G}) + \mathbf{e}^\intercal_1\end{aligned}$$

By adapting Theorem 3.1 of [21] and Corollary 3.10 of [23], conditioned on $\mathbf{A}_1$, $\mathbf{e}_1$ has a distribution that is statistically close to $D^w_{\mathbb{Z},s}$, where $s^2 = (\|\mathbf{e}_0\|^2 + m(\alpha q)^2)\cdot\omega(\sqrt{\log n})^2$. Therefore, we have $\Pr[S_4] = \Pr[\mathcal{B}_2(\mathbf{B}, \mathbf{x}^\intercal\mathbf{B} + \mathbf{y}^\intercal) = 1]$, the Inequality 6 follows.

Next, we show $\Pr[\mathsf{Forge}_4]$ and $\Pr[\mathsf{Forge}_5]$ are close. Let $\mathcal{B}'_2$ be an LWE-problem solver interacting with the PKE scheme adversary $\mathcal{A}$. $\mathcal{B}'_2$ receives an instance of $\mathsf{LWE}_{n,m,q,D_{\mathbb{Z},\alpha q}}$ $(\mathbf{B}, \mathbf{b}^\intercal) \in (\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m)$. It needs to decide that if $\mathbf{b}$ is uniformly random (independent of $\mathbf{B}$) or there exists $\mathbf{x} \sim D^n_{\mathbb{Z},\alpha q}$ and $\mathbf{y} \sim D^m_{\mathbb{Z},\alpha q}$ such that $\mathbf{b}^\intercal = \mathbf{x}^\intercal\mathbf{B} + \mathbf{y}^\intercal$. $\mathcal{B}'_2$ works as follows:

- It follows Game 5 (as specified in Fig. 4) to generate pk, sk except that it sets $\mathbf{A} \leftarrow \mathbf{B}$, $\tilde{\mathbf{c}}^*_0 \leftarrow \mathbf{b}$.
- It follows Game 5 (as specified in Fig. 4 to generate ct^* except for $\mathbf{c}^*_1$. To construct $\mathbf{c}^*_1$, $\mathcal{B}_2$ samples $\mathbf{v} \leftarrow D^w_{\mathbb{Z},\alpha q\sqrt{m}\cdot\omega(\sqrt{\log n})}$ and set $\mathbf{c}^{*\intercal}_1 \leftarrow \mathbf{c}^{*\intercal}_0\mathbf{R} + \mathbf{v}^\intercal$.
- Decryption queries made before the release of ct^* are answered by following the Game 5 specification (i.e., Fig. 5). For any decryption query $\mathsf{ct} = (\mathbf{c}^*_0, \mathbf{c}_1, \phi, \sigma) \neq \mathsf{ct}^*$, check if $\mathsf{MAC.Ver}(mk^*, \sigma, \mathbf{c}_0||\mathbf{c}_1||\phi) = 1$. If so, abort the experiment and output 1. Otherwise, return $\perp$ to $\mathcal{A}$. [5]
- Eventually, if $\mathcal{A}$ halts and $\mathcal{B}'_2$ has not previously aborted the experiment, $\mathcal{B}'_2$ outputs a random bit.

Notice that if the given challenge $(\mathbf{B}, \mathbf{b}^\intercal)$ are LWE samples, $\mathcal{B}'_2$ simulates Game 4 and aborts when Forge_4 happens (i.e., $\mathsf{MAC.Ver}(mk^*, \sigma, \mathbf{c}^*_0||\mathbf{c}_1||\phi) = 1$). So, $\Pr[\mathcal{B}'_2(\mathbf{B}, \mathbf{b}^\intercal) = 1] = \Pr[\mathcal{B}'_2(\mathbf{B}, \mathbf{b}^\intercal) = 1|\neg\mathsf{Forge}_4]\cdot\Pr[\neg\mathsf{Forge}_4] + \Pr[1 \leftarrow \mathcal{B}'_2|\mathsf{Forge}_4]\cdot\Pr[\mathsf{Forge}_4] = \frac{1}{2} + \frac{1}{2}\cdot\Pr[\mathsf{Forge}_4]$. Similarly, we have $\Pr[\mathcal{B}'_2(\mathbf{B}, \mathbf{x}^\intercal\mathbf{B} + \mathbf{y}^\intercal) = 1] = \frac{1}{2} + \frac{1}{2}\cdot\Pr[\mathsf{Forge}_5]$. This leads to the Inequality 7. □

Game 6. In Game 6, we modify the way that $\mathbf{c}^*_0$ in ct^* are generated (as shown in Fig. 4). Specifically, $\mathbf{c}^*_0 \leftarrow U(\mathbb{Z}^m_q)$. The other parts of ct^* and decryption queries are handled as in Game 5. Notice that in both Game 5 and Game 6, $\mathbf{c}^*_0$ are uniformly random over $\mathbb{Z}^m_q$ (recall in Game 5, $\mathbf{c}^*_0 \leftarrow \tilde{\mathbf{c}}^*_0 + \mathbf{A}^\intercal(\mathbf{k}^*\lfloor q/2\rfloor)$ for $\tilde{\mathbf{c}}^*_0 \leftarrow U(\mathbb{Z}^m_q)$ and $\tilde{\mathbf{c}}^*_0$ is not used in anywhere else). Hence, the distributions of the two games are identical, and, we have

$$\Pr[S_6] = \Pr[S_5] \quad \text{and} \quad \Pr[\mathsf{Forge}_6] = \Pr[\mathsf{Forge}_5] \tag{8}$$

Game 7. In Game 7, we modify the way of generating the symmetric encryption key dk^* and the MAC key mk^*, as specified in Fig. 4. In particular, dk^* and mk^* are chosen uniformly at random from the key spaces of SKE and MAC (as opposed to computed by $\mathsf{KDF}(\mathbf{k}^*)$). The other parts are exactly the same as Game 6. Since $\mathbf{k}^*$ is independent of ct^*, this change is not noticeable to the adversary by the security of KDF, as stated in the following lemma.

Lemma 6. *For Game 6 and Game 7 defined as per Fig. 4 and Fig. 5, we have*

$$|\Pr[S_7] - \Pr[S_6]| \leq \mathsf{Adv}^{\mathrm{ot-ind}}_{\mathsf{KDF},\mathcal{B}_3}(\lambda) \tag{9}$$

$$|\Pr[\mathsf{Forge}_7] - \Pr[\mathsf{Forge}_6]| \leq 2\cdot\mathsf{Adv}^{\mathrm{ind}}_{\mathsf{KDF},\mathcal{B}'_3}(\lambda) \tag{10}$$

for some adversary $\mathcal{B}_3$ and $\mathcal{B}'_3$ against the LWE problem.

[5] Note that any decryption query made before the release of ct^* with $H(\mathbf{c}_0) = H(\mathbf{c}^*_0)$ is excluded since Game 2.

Proof. We first show that $\Pr[S_7]$ and $\Pr[S_6]$ are computationally close. Let $\mathcal{B}_3$ be an adversary against KDF. First, $\mathcal{B}_3$ receives a string $r \in \{0,1\}^{|\mathcal{K}_{\mathsf{ske}}|+|\mathcal{K}_{\mathsf{mac}}|}$ and it needs tell if r is random or $r = \mathsf{KDF}(\mathbf{k}^*)$ for some $\mathbf{k}^* \in \{0,1\}^n$. $\mathcal{B}_3$ proceeds with as follows

- It follows Game 6 to generate pk, sk, $\mathbf{c}_0^*$ (as specified in Fig. 4) except that in step 2, it simply sets the first $|\mathcal{K}_{\mathsf{ske}}|$ bits (resp. the last $|\mathcal{K}_{\mathsf{mac}}$—) bits of r to be dk^* (resp. mk^*). Then, it sets $\mathbf{A} \leftarrow \mathbf{B}$, $\tilde{\mathbf{c}}_0^* \leftarrow \mathbf{b}$.
- It follows Game 7 specification to generate ct^* (as specified in Fig. 4) .
- It follows Game 7 specification (Fig. 5) to answer decryption queries.
- Finally, $\mathcal{B}_2$ outputs 1 when the security game outputs 1 (i.e., $\mathcal{A}$ outputs $b' = b$). Otherwise $\mathcal{B}_2$ outputs 0.

We can see that $\Pr[\mathcal{B}_3(1^\lambda, \mathsf{KDF}(\mathbf{k}^*)) = 1] = \Pr[S_6]$ and $\Pr[\mathcal{B}_3(1^\lambda, r)] = \Pr[S_7]$ where r is random, and thus we get the inequality 9.

Similarly, we can show that Forge_6 and Forge_7 happen with essentially the same probability provided KDF is secure. To this end, we build an efficient algorithm $\mathcal{B}_3'$ that breaks KDF. $\mathcal{B}_3'$ proceeds as follows:

- It follows Game 6 to generate pk, sk, $\mathbf{c}_0^*$ (as specified in Fig. 4) except that in step 2, it simply sets the first $|\mathcal{K}_{\mathsf{ske}}|$ bits (resp. the last $|\mathcal{K}_{\mathsf{mac}}|$) bits of r to be dk^* (resp. mk^*) . Then, it sets $\mathbf{A} \leftarrow \mathbf{B}$, $\tilde{\mathbf{c}}_0^* \leftarrow \mathbf{b}$.
- It follows Game 7 to generate ct^* (as specified in Fig. 4).
- Decryption queries made before the release of ct^* are answered by following Game 7 specification. For any decryption query $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \phi, \sigma) \neq \mathsf{ct}^*$ made after the release of ct^*
  1. if $\mathbf{c}_0 \neq \mathbf{c}_0^*$ (which implies $H(\mathbf{c}_0) \neq H(\mathbf{c}_0^*)$), answer it by following Game 7 specification (Fig. 5).
  2. For $\mathbf{c}_0 = \mathbf{c}_0^*$, check if $\mathsf{MAC.Ver}(mk^*, \sigma, \mathbf{c}_0||\mathbf{c}_1||\phi) = 1$. If so, abort the experiment and output 1. Otherwise, return $\perp$ to $\mathcal{A}$.
- Finally, $\mathcal{B}_3'$ outputs 1 when the security game outputs 1 (i.e., $\mathcal{A}$ outputs $b' = b$). Otherwise $\mathcal{B}_3'$ outputs 0.

We can see that $\Pr[\mathcal{B}_3'(1^\lambda, r \leftarrow \mathsf{KDF}(\mathbf{k}^*))] = \frac{1}{2} + \frac{1}{2} \cdot \Pr[\mathsf{Forge}_6]$ and $\Pr[\mathcal{B}_3'(1^\lambda, r \leftarrow U)] = \frac{1}{2} + \frac{1}{2} \cdot \Pr[\mathsf{Forge}_7]$ for random r. This shows the inequality 10. □

We note that in Game 7, since mk^* is sampled at random and independent of $\mathbf{c}_0^*$, $\mathbf{c}_1^*$ and ϕ^*, we can easily bound $\Pr[\mathsf{Forge}_7]$ by the security of MAC. We prove the following Lemma.

Lemma 7. *Let Q_2 be the maximun number of decryption queries that the adversary $\mathcal{A}$ can make after seeing ct^* in Game 7. Then we have*

$$\Pr[\mathsf{Forge}_7] \leq Q_2 \cdot \mathsf{Adv}^{\mathrm{ot-suf}}_{\mathsf{MAC},\mathcal{B}_4}(\lambda) \tag{11}$$

for some adversary $\mathcal{B}_4$ against the unforgeability of MAC.

Proof. Let $Q_2 = Q_2(\lambda)$ be the the upper bound on the number of decryption query made by $\mathcal{A}$ *after* seeing the challenge ciphertext ct^*. We constructed an efficient MAC-breaking algorithm $\mathcal{B}_4$ which works as follows:

- $\mathcal{B}_4$ chooses a random index j from $\{1, 2, ..., Q_2\}$.
- It follows Game 7 to generate pk, sk, $\mathbf{c}_0^*$ (as specified in Fig. 4) except for mk^* (which is possessed by the MAC challenger). Note that dk^* remains randomly chosen as in Game 7.
- For decryption queries $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \phi, \sigma) \neq \mathsf{ct}^*$ made by $\mathcal{A}$ before the release of the challenge ciphertext. we know that $H(\mathbf{c}_0) \neq H(\mathbf{c}_0^*)$. So, such decryption queries are answered in the usual way using the algorithm Invert with the gadget trapdoor as in Game 7.
- It follows Game 7 to generate ct^* (as specified in Fig. 4) except for σ^* (i.e., $\mathcal{B}_4$ generates $\mathbf{c}_1^*$, ϕ^* together with the already existed $\mathbf{c}_0^*$). To get σ^*, $\mathcal{B}_4$ sends "message" $\mathbf{c}_0^*||\mathbf{c}_1^*||\phi^*$ to its challenger and gets back σ^*. Then it releases $\mathsf{ct}^* = (\mathbf{c}_0^*, \mathbf{c}_1^*, \phi^*, \sigma^*)$ to $\mathcal{A}$.
- For ith after-challenge decryption query where $1 \leq i \leq j-1$, $\mathcal{B}_4$ proceeds as specified by Game 7. For the ith after-challenge decryption query $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \phi, \sigma)$, $\mathcal{B}_4$ submits $(\mathbf{c}_0||\mathbf{c}_1||\phi)$ to its MAC challenger and halts.

It can be seen that the probability of $\mathcal{B}_4$ in outputting a valid MAC forgery is at least $\Pr[\mathsf{Forge}_7]/Q_2$ which shows $\Pr[\mathsf{Forge}_7] \leq Q_2 \cdot \mathsf{Adv}^{\mathrm{ot-suf}}_{\mathsf{MAC},\mathcal{B}_4}(\lambda)$. □

Game 8. Game 8 is identical to Game 7 except that the challenge ciphertext components ϕ^* are chosen randomly (as specified in Fig. 4). Since dk^* is independently and randomly chosen, a straightforward reduction shows that Game 6 and Game 7 are computationally indistinguishability. In particular,

$$|\Pr[S_8] - \Pr[S_7]| \leq \mathsf{Adv}^{\mathrm{ot-ind}}_{\mathsf{SKE},\mathcal{B}_5}(\lambda). \tag{12}$$

for some adversary against SKE. Finally, we can see that in Game 8, the challenge ciphertext ct^* is independent of the bit value b. So, the adversary has no advantage in winning the game. So,

$$\Pr[S_8] = 1/2 \tag{13}$$

Combining inequalities (1) to (13) with triangle inequality, we obtain the bound

$$\begin{aligned}\mathsf{Adv}^{\mathrm{ind-cca}}_{\mathsf{PKE},\mathcal{A}}(\lambda) \leq \frac{Q_1}{q^m} + Q_2 \cdot \mathsf{Adv}^{\mathrm{coll}}_{H,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{NLWE}_{n,m,q,D_{\mathbb{Z},\alpha q}}}_{\mathcal{B}_2}(\lambda) \\ + 2 \cdot \mathsf{Adv}^{\mathsf{NLWE}_{n,m,q,D_{\mathbb{Z},\alpha q}}}_{\mathcal{B}_2'}(\lambda) + \mathsf{Adv}^{\mathrm{ot-ind}}_{\mathsf{SKE},\mathcal{B}_3}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathrm{ot-ind}}_{\mathsf{SKE},\mathcal{B}_3}(\lambda) \\ + Q_2 \cdot \mathsf{Adv}^{\mathrm{ot-suf}}_{\mathsf{MAC},\mathcal{B}_4}(\lambda) + \mathsf{Adv}^{\mathrm{ot-ind}}_{\mathsf{SKE},\mathcal{B}_5}(\lambda) + \mathsf{negl}(\lambda)\end{aligned}$$

where all terms are negligible based on our assumptions.

4 CCA-Secure Hybrid Encryption from Low-Noise LPN

This section shows how to combine our idea with the tag-based encryption system from low-noise learning parity with noise problem due to Kiltz at al. [17] to obtain a CCA-secure hybrid encryption system. Following Kiltz et al. [17],

we denote by Ber_p be the Bernouli distribution with parameter $0 \geq p \leq 1/2$, i.e. $x \leftarrow \mathsf{Ber}_p$ is the random variable over $\{0,1\}$ with $\Pr[x=1]=p$. The LPN problem and its variant extended Knapsack LPN problem used in our system are as defined in Section 2.2, [17]. They have low noise rate $p \approx 1/\sqrt{n}$. We denote by $|\mathbf{a}|$ the hamming weight of vector $\mathbf{a} \in \mathbb{Z}_2^n$.

The hybrid encryption system uses the following public parameters.

1. The security parameter λ, the dimension of LPN problem n, and parameter $m \geq 2n$. A constant c with $0 < c < 1/4$ that defines: (1) the Bernoulli parameter $p = \sqrt{c/m}$, and (2) the bound $\beta = 2\sqrt{cm}$ for checking decryption consistency, and (2) a binary linear error-correcting code with generation matrix $\hat{\mathbf{G}}_1 : \mathbb{Z}_2^{n\times m}$ which corrects up to αm errors for some $4c < \alpha < 1$. Another error-correcting code with generation matrix $\hat{\mathbf{G}}_2 \in \mathbb{Z}_2^{n\times \ell}$ (where $\mathbb{Z}_2^\ell$ for $\ell \geq \lambda$ is the session key space for the KEM part).
2. A full-rank difference encoding (see [3,17]) $\mathsf{FRD} : \mathbb{Z}_2^n \to \mathbb{Z}_2^{n\times n}$ that for any $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_2^n$ with $\mathbf{x} \neq \mathbf{y}$, $\mathsf{FRD}(\mathbf{x}) - \mathsf{FRD}(\mathbf{y})$ is invertible over $\mathbb{Z}_2^{n\times n}$. In particular, $\mathsf{FRD}(\mathbf{x})$ is invertible over $\mathbb{Z}_2^{n\times n}$ if $\mathbf{x} \neq \mathbf{0}$.
3. A collision resistance hash function $H : \{0,1\} \to \mathbb{Z}_2^n \setminus \{\mathbf{0}\}$ where $\mathbf{0}$ is the zero vector of $\mathbb{Z}_2^n$, a secret-key encryption system $\mathsf{SKE} = (\mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ with key space $\mathcal{K}_{\mathsf{ske}}$, message space $\mathcal{M}_{\mathsf{ske}}$, and ciphertext space $\mathcal{C}_{\mathsf{ske}}$, a secure message authentication code $\mathsf{MAC} = (\mathsf{MAC.Sign}, \mathsf{MAC.Ver})$ with key space $\mathcal{K}_{\mathsf{mac}}$. A key derivation function $\mathsf{KDF} : \{0,1\}^\ell \to \mathcal{K}_{\mathsf{ske}} \times \mathcal{K}_{\mathsf{mac}}$.

- $\mathsf{PKE.Gen}(1^\lambda)$:
  1. $\mathbf{A} \leftarrow U(\mathbb{Z}_2^{n\times m})$, $\mathbf{A}_2 \leftarrow U(\mathbb{Z}_2^{n\times m})$, $\mathbf{U} \leftarrow U(\mathbb{Z}_2^{n\times \ell})$.
  2. $\mathbf{T} \leftarrow \mathsf{Ber}_p^{m\times m}$, $\mathbf{A}_1 \leftarrow \mathbf{AT}$.
  3. $\mathsf{pk} \leftarrow (\mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{U})$, $\mathsf{sk} \leftarrow \mathbf{T}$.
- $\mathsf{PKE.Enc}(\mathsf{pk}, M)$:
  1. $\mathbf{k} \leftarrow U(\{0,1\}^\ell)$, $(dk, mk) \leftarrow \mathsf{KDF}(\mathbf{k})$.
  2. $\mathbf{s} \leftarrow U(\mathbb{Z}_2^n)$, $\mathbf{T}_1, \mathbf{T}_2 \leftarrow \mathsf{Ber}_p^{m\times m}$, $\mathbf{e}_0 \leftarrow \mathsf{Ber}_p^m$.
  3. $\mathbf{e}_1^\intercal \leftarrow \mathbf{e}_0^\intercal \mathbf{T}_1$, $\mathbf{e}_2 \leftarrow \mathbf{e}_0^\intercal \mathbf{T}_2$, $\mathbf{e}_3 \leftarrow \mathsf{Ber}_p^\ell$.
  4. $[\mathbf{c}_0^\intercal | \mathbf{c}_2^\intercal | \mathbf{c}_3^\intercal] \leftarrow \mathbf{s}^\intercal [\mathbf{A} | \mathbf{A}_2 | \mathbf{U}] + [\mathbf{e}_0^\intercal | \mathbf{e}_2^\intercal | \mathbf{e}_3^\intercal + \mathbf{k}^\intercal \hat{\mathbf{G}}_2]$
  5. $\mathbf{c}_1^\intercal \leftarrow \mathbf{s}^\intercal (\mathbf{A}_1 + \mathsf{FRD}(H(\mathbf{c}_0 || \mathbf{c}_2 || \mathbf{c}_3))\hat{\mathbf{G}}_1) + \mathbf{e}_1^\intercal$
  6. $\phi \leftarrow \mathsf{SKE.Enc}(dk, M)$, $\sigma \leftarrow \mathsf{MAC.Sign}(mk, \mathbf{c}_0 || \mathbf{c}_1 || \mathbf{c}_2 || \mathbf{c}_3 || \phi)$.
  7. Return $\mathsf{ct} \leftarrow (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, , \phi, \sigma)$.
- $\mathsf{PKE.Dec}(\mathsf{pk}, \mathsf{sk}, \mathsf{ct})$:
  1. Parse $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2.\mathbf{c}_3, \phi, \sigma)$; Output $\perp$ if ct doesn't parse.
  2. Let $\mathbf{H} \leftarrow \mathsf{FRD}(H(\mathbf{c}_0 || \mathbf{c}_2 || \mathbf{c}_3)) \in \mathbb{Z}_2^{n\times n}$; Set $\tilde{\mathbf{c}}_0^\intercal \leftarrow -\mathbf{c}_0^\intercal \mathbf{T} + \mathbf{c}_1^\intercal = (-\mathbf{e}_0^\intercal \mathbf{T} + \mathbf{e}_1^\intercal) + \mathbf{s}^\intercal \mathbf{H} \hat{\mathbf{G}}_1$; Use the error correction property of $\hat{\mathbf{G}}_1$ to reconstruct $\mathbf{s}^\intercal \mathbf{H}$ (from the error $-\mathbf{e}_0^\intercal \mathbf{T} + \mathbf{e}_1^\intercal$), and, then recover $\mathbf{s}^\intercal \leftarrow \mathbf{s}^\intercal \mathbf{H} \mathbf{H}^{-1}$.
  3. If $|\mathbf{c}_0^\intercal - \mathbf{s}^\intercal \mathbf{A}| \leq \beta$, $|\mathbf{c}_1^\intercal - \mathbf{s}^\intercal (\mathbf{A}_1 + \mathbf{H}\hat{\mathbf{G}}_1)| \leq \alpha m/2$, $|\mathbf{c}_2^\intercal - \mathbf{s}^\intercal \mathbf{A}| \leq \alpha m/2$, compute $\tilde{\mathbf{c}}_2^\intercal \leftarrow \mathbf{c}_3^\intercal - \mathbf{s}^\intercal \mathbf{U} = \mathbf{e}_3^\intercal + \mathbf{k}^\intercal \hat{\mathbf{G}}_2$ and use the error correction property of $\hat{\mathbf{G}}_2$ to recover $\mathbf{k}$ and $\mathbf{e}_3$; Otherwise, abort and return $\perp$.
  4. If $|\mathbf{e}_3| > \beta$, abort and return $\perp$; Otherwise, set $(dk, mk) \leftarrow \mathsf{KDF}(\mathbf{k})$ and continue.

5. Output $M \leftarrow \mathsf{SKE.Dec}(dk, \phi)$ if $1 \leftarrow \mathsf{MAC.Ver}(mk, \mathbf{c}_0||\mathbf{c}_1||\mathbf{c}_2||\mathbf{c}_3||\phi, \sigma)$; Otherwise output $\perp$.

Discussions. The system is based on the selective-tag weak CCA-secure encryption system by Kiltz et al. [17] (KMP system), except with the following differences. First, the system uses a hash of the concatenation of ciphertext components $\mathbf{c}_0, \mathbf{c}_2, \mathbf{c}_3$ as a tag for the tag-based trapdoor function of the KMP system, while the tag is chosen uniformly random from $\mathbb{Z}_2^n$ (and then mapped into a matrix in $\mathbb{Z}_2^{n \times n}$). Second, $\mathbf{k}$ in [17] is the message being encrypted. In our case, the session key is encapsulated in the KEM part. Then, the session key is used to derive keys for the secret-key encryption and the MAC. In summary, the "public-key" part of our system is essentially the same as the KMP system, and hence, its correctness follows directly from the correctness of the KMP encryption system (refer to [17] for details), and the correctness of the secret-key encryption and the MAC systems.

The security of the above HE system follows from the security of our LWE-based HE system. There are three differences between these two systems. The first one is that the LPN-based one has an extra matrix $\mathbf{A}_2 \in \mathbb{Z}_2^{n \times m}$, which is needed for the double trapdoor technique from [17]. Due to the low noise rate of the private key $\mathbf{T}$, the public matrix $\mathbf{A}_1 = \mathbf{AT}$ is only computationally indistinguishable from random (based on the Knapsack LPN problem). So, in the security proof, $\mathbf{T}$ cannot be used for responding to decryption queries when we need to argue the distribution of $\mathbf{A}$ is computationally uniform. The matrix $\mathbf{A}_2$ helps to bring in an extra trapdoor, by setting $[\mathbf{A}|\mathbf{A}_2] = [\mathbf{A}|\mathbf{A}\tilde{\mathbf{T}} + \hat{\mathbf{G}}_1]$, to bridge this gap. The second difference is we add LPN samples $\mathbf{c}_3$ for hiding the session key $\mathbf{k}$, whereas we directly encode the session key into the LWE "witness" $\mathbf{s} \in \mathbb{Z}_q^n$. We do this with the LPN-based constriction to follow the KMP construction [17]. The last one is the construction of noise terms $\mathbf{e}_1, \mathbf{e}_2$. We follow the KMP system to compute the noise terms $\mathbf{e}_1$, $\mathbf{e}_2$ from the noise $\mathbf{e}_0$ (whereas all LWE error terms are vectors chosen independently from some Gaussian distributions). The hardness of the leaky knapsack LPN problem is used to handle such a correlation in [17], which applies to our case as well.

The security proof of our LPN system is very similar to the security proof of our LWE-based HE system, thanks to the similarities of Kiltz et al.'s double-trapdoor and Micciancio-Peikert lattice trapdoor. Our LWE-based system crucially relies on the fact that the LWE trapdoor function's tag, $H(\mathbf{c}_0)$, is a secure commitment of the session key. Similarly, the LPN trapdoor function's "tag" $H(\mathbf{c}_0||\mathbf{c}_2||\mathbf{c}_3)$ is also an binding and hiding commitment of the session key: Given a random $\mathbf{A} \in \mathbb{Z}_2^{n \times m}$, and $\mathbf{e} \leftarrow \mathsf{Ber}_p^m$, $\mathbf{c}_0^\intercal = \mathbf{s}^\intercal\mathbf{A} + \mathbf{e}_0^\intercal$, a noisy random linear code w.r.t $\mathbf{A}$, determines $\mathbf{s}$ w.h.p over the choice of $\mathbf{A}$ (see [11], Theorem 2.3). So, $\mathbf{c}_3^\intercal = \mathbf{s}^\intercal\mathbf{A} + \mathbf{e}_3^\intercal + \mathbf{k}^\intercal\hat{\mathbf{G}}_2$ with $\mathbf{e}_3 \leftarrow \mathsf{Ber}_p^m$ determines $\mathbf{k}$ due to the error correction and unique decoding properties of $\hat{\mathbf{G}}_2$. The SKE and MAC in the two systems are used in the same way. Therefore, we can apply exactly the same strategy that used in proving the LWE-based construction to the LPN-based system. Due to the space limitation and high similarity with the proof of Theorem 1, we omit the security proof of the LPN-based HE system.

5 Conclusion

Based on the LWE problem and the low-noise LPN problem, we have presented two CCA-secure hybrid encryption systems in the standard model. The systems give the first post-quantum examples of CCA-secure hybrid encryption systems with non-CCA secure KEMs. Our systems are efficient. For instance, the KEM part of the LWE-based HE system is simply the lattice trapdoor function from [19], which is supported by efficient implementations (e.g., [5,12]). Our systems do not fit into the theoretic frameworks established in [1] and [16]. We leave providing a theoretic framework that explains our constructions as future work.

References

1. Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In *CRYPTO 2005*, 205–222.
2. Masayuki Abe, Rosario Gennaro, and Kaoru Kurosawa. Tag-KEM/DEM: A new framework for hybrid encryption. *Journal of Cryptology*, 21(1):97–130, 2008.
3. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT 2010*, 553–572.
4. Michael Alekhnovich. More on average case vs approximation complexity. In *FOCS 2003. Proceedings.*, 298–307.
5. Pauline Bert, Pierre-Alain Fouque, Adeline Roux-Langlois, and Mohamed Sabt. Practical implementation of ring-SIS/LWE based signature and IBE. In *PQCrypto 2018*, 271–291.
6. Dan Boneh and Jonathan Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In *CT-RSA 2005*, 87–103.
7. Xavier Boyen, Malika Izabachène, and Qinyi Li. A simple and efficient CCA-secure lattice KEM in the standard model. In *SCN 2020*, 321–337.
8. Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO 1998*, 13–25.
9. Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.
10. Nico Döttling. Low noise LPN: KDM secure public key encryption and sample amplification. In *PKC 2015*, 604–626.
11. Nico Marcel Döttling. *Cryptography based on the Hardness of Decoding.* PhD thesis, Karlsruhe Institute of Technology, 2014.
12. Rachid El Bansarkhani. LARA: a design concept for lattice-based encryption. In *Financial Cryptography and Data Security 2019*, 377–395.
13. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of cryptology*, 26(1):80–101, 2013.
14. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC 2008*, 197–206.
15. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the fujisaki-okamoto transformation. In *TCC 2017*, 341–371.
16. Dennis Hofheinz and Eike Kiltz. Secure hybrid encryption from weakened key encapsulation. In *CRYPTO 2007*, 553–571. Springer, 2007.

17. Eike Kiltz, Daniel Masny, and Krzysztof Pietrzak. Simple chosen-ciphertext security from low-noise LPN. In *PKC 2014*, 1–18.
18. Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO*, 426–442.
19. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, 700–718.
20. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, April 2007.
21. Chris Peikert. An efficient and parallel gaussian sampler for lattices. In *CRYPTO 2010*, 80–97.
22. Chris Peikert et al. A decade of lattice cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, 2016.
23. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC 2005*, STOC '05, 84–93
24. Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004.
25. Jiang Zhang, Yu Yu, Shuqin Fan, and Zhenfeng Zhang. Improved lattice-based CCA2-secure PKE in the standard model. Cryptology ePrint Archive, Report 2019/149, 2019. `https://eprint.iacr.org/2019/149`.