A Framework for Achieving KDM-CCA Secure Public-Key Encryption

Fuyuki Kitagawa and Keisuke Tanaka

Tokyo Institute of Technology, Tokyo, Japan
{kitagaw1,keisuke}@is.titech.ac.jp

Abstract

We propose a framework for achieving a public-key encryption (PKE) scheme that satisfies key dependent message security against chosen ciphertext attacks (KDM-CCA security) based on projective hash function. Our framework can be instantiated under the decisional diffie-hellman (DDH), quadratic residuosity (QR), and decisional composite residuosity (DCR) assumptions. The constructed schemes are KDM-CCA secure with respect to affine functions and compatible with the amplification method shown by Applebaum (EUROCRYPT 2011). Thus, they lead to PKE schemes satisfying KDM-CCA security for all functions computable by a-priori bounded size circuits. They are the first PKE schemes satisfying such a security notion in the standard model using neither non-interactive zero knowledge proof nor bilinear pairing.

The above framework based on projective hash function captures only KDM-CCA security in the single user setting. However, we can prove the KDM-CCA security in the multi user setting of our concrete instantiations by using their algebraic structures explicitly. Especially, we prove that our DDH based scheme satisfies KDM-CCA security in the multi user setting with the same parameter setting as in the single user setting.

Keywords: key dependent message security, chosen ciphertext security, projective hash function.

Contents

1 Introduction

1.1 Backgrounds

Key dependent message (KDM) security, introduced by Black, Rogaway, and Shrimpton [3], guarantees confidentiality of communication even if an adversary can get a ciphertext of secret keys. KDM security is useful for many practical applications including anonymous credential systems [7] and hard disk encryption systems (e.g., BitLocker [4]). KDM security is defined with respect to a function family $\mathcal{F}$. Let n denote the number of keys and $\mathbf{sk} = (\mathsf{sk}_1, \cdots, \mathsf{sk}_n)$ be secret keys. Informally, a PKE scheme is said to be $\mathcal{F}$-KDM secure if confidentiality of messages is protected even when an adversary can see a ciphertext of $f(\mathbf{sk})$ under the k-th public key for any $f \in \mathcal{F}$ and $k \in \{1, \cdots, n\}$. In this paper, we focus on constructing PKE schemes that satisfy KDM security against chosen ciphertext attacks, namely *KDM-CCA* security in the standard model.

Camenisch, Chandran, and Shoup [6] proposed the first KDM-CCA secure PKE based on the Naor-Yung paradigm [16]. They showed that for any function class $\mathcal{F}$, $\mathcal{F}$-KDM-CPA secure PKE can be transformed into $\mathcal{F}$-KDM-CCA secure one assuming non-interactive zero knowledge (NIZK) proof. They also showed a concrete instantiation based on the decisional diffie-hellman (DDH) assumption on a bilinear pairing. Subsequently, Hofheinz [10] showed more efficient KDM-CCA secure PKE. His scheme is circular secure (KDM-CCA secure with respect to selection functions) relying on both the DDH and decisional composite residuosity (DCR) assumptions on a bilinear pairing.

The first KDM-CCA secure PKE using neither NIZK proof nor bilinear pairing was proposed by Lu, Li, and Jia [14]. They claimed their scheme is KDM-CCA secure with respect to affine functions ($\mathcal{F}_{\mathsf{aff}}$-KDM-CCA secure) relying on both the DDH and DCR assumptions. However, a flow on their security proof was later discovered by Han, Liu, and Lyu [9]. Han et al. also showed a new $\mathcal{F}_{\mathsf{aff}}$-KDM-CCA secure PKE scheme based on the construction methodology of Lu et al. In addition, they constructed KDM-CCA secure PKE with respect to bounded degree polynomial functions. Their schemes are efficient and secure relying on both the DDH and DCR assumptions.

Despite the above previous efforts, it is still open whether we can construct KDM-CCA secure PKE based on a single computational assumption using neither NIZK proof nor bilinear pairing. All existing KDM-CCA secure PKE schemes without NIZK proof were proved to be secure relying on both the DDH and DCR assumptions. These schemes are proposed based on a specific algebraic structure and it is crucial to assume the hardness of both the DDH and DCR problems on the specific algebraic structure. Thus, it seems difficult to construct KDM-CCA secure PKE based on a single computational assumption using their techniques.

Moreover, it is also an open question whether we can construct KDM-CCA secure PKE with respect to all functions computable by bounded size circuits ($\mathcal{F}_{\mathsf{all}}$-KDM-CCA secure) using neither NIZK proof nor bilinear pairing. The only existing way to construct $\mathcal{F}_{\mathsf{all}}$-KDM-CCA secure PKE is to utilize the amplification method shown by Applebaum [2]. Applebaum showed if a PKE scheme is KDM-CCA secure with respect to projection functions, we can transform it into a $\mathcal{F}_{\mathsf{all}}$-KDM-CCA secure one, where projection function is a function whose each output bit depends on only a single bit of an input. Kitagawa, Matsuda, Hanaoka, and Tanaka [12] later showed we can perform such a transformation even if the underlying PKE is only KDM-CCA secure with respect to projection functions whose output is one bit.

Among existing KDM-CCA secure schemes, only Camenisch et al.'s scheme is compatible with those transformations. Thus, a construction of $\mathcal{F}_{\mathsf{all}}$-KDM-CCA secure PKE using neither NIZK proof nor bilinear pairing is not known so far.

1.2 Our Results

Based on the above back ground, we show the following results.

A framework achieving KDM-CCA security in the single user setting. First, we propose a framework to construct PKE that is $\mathcal{F}_{\mathsf{aff}}$-KDM-CCA secure in the single user setting based on *projective hash function*. Our framework can be instantiated based on the DDH, quadratic residuosity (QR), and DCR assumptions. More specifically, we obtain the following theorem.

Theorem 1 (Informal) *Under each of the DDH, QR, and DCR assumptions, there exists PKE that is $\mathcal{F}_{\mathsf{aff}}$-KDM-CCA secure in the single user setting.*

These schemes are also KDM-CCA secure with respect to projection functions of single-bit output thus compatible with the amplification method of Applebaum [2] and Kitagawa et al. [12]. Thus, we obtain the following corollary.

Corollary 1 (Informal) *Under each of the DDH, QR, and DCR assumptions, there exists PKE that is $\mathcal{F}_{\mathsf{all}}$-KDM-CCA secure in the single user setting.*

KDM-CCA secure PKE in the multi user setting. Then, we focus on KDM-CCA security in the multi user setting. Although the above framework based on projective hash function captures only KDM-CCA security in the single user setting, we can prove the KDM-CCA security in the multi user setting of our concrete instantiations by using their algebraic structures explicitly.

Our DDH based construction is an extension of the KDM-CPA secure scheme proposed by Boneh, Halevi, Hamburg, and Ostrovsky [4]. Similarly to Boneh et al., using the self-reducibility of the DDH problem, we can prove the KDM-CCA security in the multi user setting of our DDH based construction with the same parameter setting as in the single user setting. Especially, we formally prove the following theorem.

Theorem 2 (Informal) *Under the DDH assumption, there exists PKE that is $\mathcal{F}_{\mathsf{aff}}$-KDM-CCA secure in the multi user setting.*

Since the DDH based construction is compatible with the results by Applebaum [2] and Kitagawa et al. [12], we obtain the following corollary.

Corollary 2 (Informal) *Under the DDH assumption, there exists PKE that is $\mathcal{F}_{\mathsf{all}}$-KDM-CCA secure in the multi user setting.*

Our QR and DCR based constructions are extensions of the KDM-CPA secure scheme proposed by Brakerski and Goldwasser [5]. If we allow the length of a secret key to depend on the number of users, we can also prove the KDM-CCA security in the multi user setting of our DCR and QR based schemes using a technique similar to Brakerski and Goldwasser. We briefly explain how to prove it after the proof of multi user security of the DDH based scheme.

We summarize our results and previous results in Figure 1.

Scheme	Functions	Assumption	Free of pairing	Amplification	Flexible parameter
[6]	Affine	DDH		✓	✓
[10]	Circular	DDH+DCR			✓
[9]-1	Affine	DDH+DCR	✓		✓
[9]-2	polynomial	DDH+DCR	✓		✓
Ours 1	Affine	DDH	✓	✓	✓
Ours 2	Affine	QR	✓	✓	
Ours 3	Affine	DCR	✓	✓	

Figure 1: Comparison of KDM-CCA secure PKE schemes. "Amplification" indicates whether we can transform the scheme into $\mathcal{F}_{\mathsf{all}}$-KDM-CCA secure one using the results of Applebaum [2] and Kitagawa et al. [12]. "Flexible parameter" indicates whether we can prove KDM-CCA security in the multi user setting of the scheme without making the length of a secret key depend on the number of users.

1.3 Technical Overview

Our starting point is the constructions of PKE proposed by Wee [19] that is KDM secure in the single user setting (hereafter, KDM$^{(1)}$ security). He showed how to construct KDM$^{(1)}$-CPA secure PKE based on *homomorphic* projective hash function. His framework captures the previous constructions proposed by Boneh et al. [4] and Brakerski and Goldwasser [5].

Projective hash function was originally introduced by Cramer and Shoup [8] to construct PKE satisfying indistinguishability against chosen ciphertext attacks (IND-CCA security). Thus, we have a natural question whether we can construct KDM$^{(1)}$-CCA secure PKE based on projective hash function.

We answer the above question affirmatively with a simple construction. Below, we first review the construction proposed by Wee [19].

1.3.1 KDM$^{(1)}$-CPA Secure PKE Based on Homomorphic Projective Hash Function

We consider a group $\mathcal{C}$ and its subgroup $\mathcal{V}$ satisfying *the subgroup indistinguishability*, that is, uniform distributions over $\mathcal{C}$ and $\mathcal{V}$ are computationally indistinguishable. Based on $\mathcal{C}$ and $\mathcal{V}$, we define projective hash function as follows. A projective hash function is a family $\mathcal{H}$ of hash functions $\Lambda_{\mathsf{sk}} : \mathcal{C} \to \mathcal{K}$ indexed by a key $\mathsf{sk} \in \mathcal{SK}$, where $\mathcal{K}$ is a group. Let μ be a projection map defined over $\mathcal{SK}$. We require Λ_{sk} be *projective*, that is, for every $c \in \mathcal{V}$, the value of $\Lambda_{\mathsf{sk}}(c)$ is determined by only c and $\mathsf{pk} = \mu(\mathsf{sk})$. In addition, we require that there exist a public evaluation algorithm Pub that given pk, $c \in \mathcal{V}$, and a witness w of $c \in \mathcal{V}$, outputs $\Lambda_{\mathsf{sk}}(c)$. Below, we denote group operations of $\mathcal{C}$ and $\mathcal{K}$ by "$\cdot$" and "$+$", respectively.

Using a projective hash function $\mathcal{H}$, we can naturally construct a PKE scheme Π as follows. When generating a key pair $(\mathsf{pk}, \mathsf{sk})$, we sample random sk and compute $\mathsf{pk} = \mu(\mathsf{sk})$. When encrypting a message $m \in \mathcal{K}$, we first sample $c \xleftarrow{\mathsf{r}} \mathcal{V}$ with a witness w of $c \in \mathcal{V}$. Then, we compute $d \leftarrow \mathsf{Pub}(\mathsf{pk}, c, w) + m$ and set (c, d) as a ciphertext. When decrypting (c, d), we compute $m \leftarrow d - \Lambda_{\mathsf{sk}}(c)$.

Π is IND-CPA secure if $\mathcal{H}$ is *smooth*, that is, the value of $\Lambda_{\mathsf{sk}}(c)$ is statistically chose to uniform given $\mathsf{pk} = \mu(\mathsf{sk})$ and c, where $\mathsf{sk} \xleftarrow{\mathsf{r}} \mathcal{SK}$ and $c \xleftarrow{\mathsf{r}} \mathcal{C}$.[1] We prove the IND-CPA security of Π as follows. We first switch c^* used to encrypt the challenge message to $c^* \xleftarrow{\mathsf{r}} \mathcal{C}$ by using the subgroup indistinguishability. Then, the distribution of the resulting ciphertext is close to uniform due to the smoothness and thus IND-CPA security follows.

[1] More specifically, this property is called average-case smoothness in general.

KDM security from homomorphism. Wee [19] showed Π is also KDM$^{(1)}$-CPA secure if $\mathcal{H}$ is *homomorphic*, that is, for every $c_0, c_1 \in \mathcal{C}$, it holds that $\Lambda_{\mathsf{sk}}(c_0 \cdot c_1) = \Lambda_{\mathsf{sk}}(c_0) + \Lambda_{\mathsf{sk}}(c_1)$. More precisely, Π is KDM$^{(1)}$-CPA secure with respect to functions defined as $f_e(\mathsf{sk}) = \Lambda_{\mathsf{sk}}(e)$, where $e \in \mathcal{C}$. Note that this function class corresponds to the set of affine functions in his instantiations.

If $\mathcal{H}$ is homomorphic, we can change the distribution of an encryption of $f_e(\mathsf{sk})$, that is $(c, \mathsf{Pub}(\mathsf{pk}, c, w) + \Lambda_{\mathsf{sk}}(e))$ as

$$
\begin{aligned}
& (c, \mathsf{Pub}(\mathsf{pk}, c, w) + \Lambda_{\mathsf{sk}}(e)) \,, \text{where } c \xleftarrow{\mathsf{r}} \mathcal{V} \\
=\, & (c, \Lambda_{\mathsf{sk}}(c) + \Lambda_{\mathsf{sk}}(e)) \,, \text{where } c \xleftarrow{\mathsf{r}} \mathcal{V} \quad \text{(by projective property)} \\
=\, & (c, \Lambda_{\mathsf{sk}}(c \cdot e)) \,, \text{where } c \xleftarrow{\mathsf{r}} \mathcal{V} \quad \text{(by homomorphism)} \\
\approx_{\mathsf{c}}\, & (c, \Lambda_{\mathsf{sk}}(c \cdot e)) \,, \text{where } c \xleftarrow{\mathsf{r}} \mathcal{C} \quad \text{(by subgroup indistinguishability)} \\
\approx_{\mathsf{s}}\, & \left(c \cdot e^{-1}, \Lambda_{\mathsf{sk}}(c)\right) \,, \text{where } c \xleftarrow{\mathsf{r}} \mathcal{C} \quad \text{(since } e \in \mathcal{C}\text{)} \\
\approx_{\mathsf{c}}\, & \left(c \cdot e^{-1}, \Lambda_{\mathsf{sk}}(c)\right) \,, \text{where } c \xleftarrow{\mathsf{r}} \mathcal{V} \quad \text{(by subgroup indistinguishability)} \\
=\, & \left(c \cdot e^{-1}, \mathsf{Pub}(\mathsf{pk}, c, w)\right) \,, \text{where } c \xleftarrow{\mathsf{r}} \mathcal{V} \quad \text{(by projective property)} \ ,
\end{aligned}
$$

where w denotes a witness of $c \in \mathcal{V}$, and $\approx_{\mathsf{c}}$ and $\approx_{\mathsf{s}}$ denote computational indistinguishability and statistical indistinguishability, respectively. This means that we can simulate an encryption of $f_e(\mathsf{sk})$ without sk. Then, based on the standard hybrid argument, we can prove the KDM$^{(1)}$-CPA security of Π using the smoothness of $\mathcal{H}$ similarly to the proof for the IND-CPA security of Π.

1.3.2 Extension to KDM$^{(1)}$-CCA Secure PKE

We can construct IND-CCA secure PKE by adding 2-*universal* projective hash function to the construction of Π. More precisely, we use a projective hash function $\hat{\mathcal{H}}$ consisting of hash functions $\hat{\Lambda}_{\hat{\mathsf{sk}}}$ indexed by $\hat{\mathsf{sk}} \in \hat{\mathcal{SK}}$ defined on $\mathcal{C}$ and $\mathcal{V}$.[2] Let $\hat{\mu}$ and $\hat{\mathsf{Pub}}$ be the projection map and public evaluation algorithm of $\hat{\mathcal{H}}$. We require that $\hat{\mathcal{H}}$ be 2-*universal*, that is, for every $\hat{\mathsf{pk}}$, $c, c^* \in \mathcal{C} \setminus \mathcal{V}$, and $\mathsf{K}, \mathsf{K}^* \in \mathcal{K}$, $\hat{\Lambda}_{\hat{\mathsf{sk}}}(c) = \mathsf{K}$ holds with only negligible probability under the condition that $\hat{\mathsf{pk}} = \hat{\mu}\left(\hat{\mathsf{sk}}\right)$ and $\hat{\Lambda}_{\hat{\mathsf{sk}}}(c^*) = \mathsf{K}^*$, where $\hat{\mathsf{sk}} \xleftarrow{\mathsf{r}} \hat{\mathcal{SK}}$.

We modify Π into IND-CCA secure Π' as follows. When generating a key pair, in addition to $(\mathsf{pk}, \mathsf{sk})$, we sample $\hat{\mathsf{sk}} \xleftarrow{\mathsf{r}} \hat{\mathcal{SK}}$ and compute $\hat{\mathsf{pk}} = \hat{\mu}\left(\hat{\mathsf{sk}}\right)$. A public key and secret key of Π' are $\left(\mathsf{pk}, \hat{\mathsf{pk}}\right)$ and $\left(\mathsf{sk}, \hat{\mathsf{sk}}\right)$, respectively. When encrypting a message, we first compute c and d in the same way as Π using pk. Then, we compute $\pi \leftarrow \hat{\mathsf{Pub}}\left(\hat{\mathsf{pk}}, c, w\right)$ and set (c, d, π) as the resulting ciphertext. When decrypting (c, d, π), we first check whether $\pi = \hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$ holds and if so decrypt a message in the same way as Π using sk. Otherwise, we output $\perp$.

Since $\hat{\mathcal{H}}$ is 2-universal, an adversary cannot compute $\hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$ correctly for $c \in \mathcal{C} \setminus \mathcal{V}$ even if he obtain a single hash value $\hat{\Lambda}_{\hat{\mathsf{sk}}}(c^*)$ for $c^* \in \mathcal{C} \setminus \mathcal{V}$ in the challenge ciphertext. In other words, the adversary cannot make a valid decryption query (c, d, π) for $c \in \mathcal{C} \setminus \mathcal{V}$. Then, from the projective property of $\mathcal{H}$, the adversary cannot obtain information of sk other than pk through decryption queries. Thus, we can reduce the IND-CCA security of Π' to the smoothness of $\mathcal{H}$.

[2] In the actual construction of IND-CCA secure PKE, we need to use 2-universal projective hash function defined on $\mathcal{C} \times \mathcal{K}$ and $\mathcal{V} \times \mathcal{K}$. Such a primitive is called *extended* projective hash function [8]. For simplicity, we ignore this issue here.

Problems for proving KDM$^{(1)}$-CCA security. Even if $\mathcal{H}$ is homomorphic, we cannot prove the KDM$^{(1)}$-CCA security of Π' straightforwardly. In the security game of KDM$^{(1)}$-CCA security, an adversary can obtain an encryption of $\hat{\mathsf{sk}}$ in addition to that of sk. Thus, we need to eliminate $\hat{\mathsf{sk}}$ from the view of the adversary to use the 2-universal property of $\hat{\mathcal{H}}$.

Moreover, if we can do that, there is another problem. Consider functions of the form $f_e\left(\mathsf{sk}, \hat{\mathsf{sk}}\right) = \Lambda_{\mathsf{sk}}(e) + f\left(\hat{\mathsf{sk}}\right)$, where $e \in \mathcal{C}$ and $f : \hat{\mathcal{SK}} \to \mathcal{K}$ is a function. If $\mathcal{H}$ is homomorphic, using a similar argument as Wee [19], we can simulate an encryption of $f_e\left(\mathsf{sk}, \hat{\mathsf{sk}}\right)$ by

$$\left(c \cdot e^{-1}, \mathsf{Pub}(\mathsf{pk}, c, w) + f\left(\hat{\mathsf{sk}}\right), \hat{\Lambda}_{\hat{\mathsf{sk}}}\left(c \cdot e^{-1}\right)\right) \ ,$$

where $c \in \mathcal{V}$ and w is an witness of $c \in \mathcal{V}$. Even if we can eliminate $f\left(\hat{\mathsf{sk}}\right)$ from the second component, the third component $\hat{\Lambda}_{\hat{\mathsf{sk}}}\left(c \cdot e^{-1}\right)$ incurs another problem. e is an element chosen by an adversary in the security game, and thus $c \cdot e^{-1}$ might not be included in $\mathcal{V}$. Thus, the adversary can obtain a hash value $\hat{\Lambda}_{\hat{\mathsf{sk}}}\left(c \cdot e^{-1}\right)$ for $c \cdot e^{-1} \notin \mathcal{V}$ through each KDM query f_e. In this case, we cannot rely on the 2-universal property of $\hat{\mathcal{H}}$ to argue about decryption queries made by the adversary if he makes multiple KDM queries. Therefore, we also need to eliminate $\hat{\Lambda}_{\hat{\mathsf{sk}}}\left(c \cdot e^{-1}\right)$ from the view of the adversary.

Our solution: Double layered encryption. We solve the above two problems at once by extending double layered encryption techniques originally used to expand the plaintext space of an IND-CCA secure PKE scheme [15, 11]. More precisely, by adding an outer encryption layer, we put the estimation of the probability that an adversary makes an "illegal" decryption query off till the end of the sequence of games where all information about the inner layer is eliminated from the challenge ciphertexts. We use an IND-CCA secure PKE scheme Π_{cca} as the outer layer encryption scheme. When encrypting a message, we first generate (c, d, π) in the same way as Π' and then encrypt them by Π_{cca}. We call the resulting PKE scheme Π_{kdm}.

Of course, if we just maintain a secret key csk of Π_{cca} as a part of a secret-key of Π_{kdm}, we cannot use the IND-CCA security of Π_{cca}. Thus, we add a modification. We maintain csk after encrypting by $\mathcal{H}$. More precisely, we modify the key generation procedure of Π_{kdm} as follows. We first generate $(\mathsf{pk}, \mathsf{sk})$ and $\left(\hat{\mathsf{pk}}, \hat{\mathsf{sk}}\right)$ in the same way as Π' and generate a key pair $(\mathsf{cpk}, \mathsf{csk})$ of Π_{cca}. Moreover, we sample $c^* \xleftarrow{\mathsf{r}} \mathcal{C}$ and compute $d^* \xleftarrow{\mathsf{r}} \Lambda_{\mathsf{sk}}(c^*) + \mathsf{csk}$.[3] The resulting public key and secret key of Π_{kdm} are $\left(\mathsf{pk}, \hat{\mathsf{pk}}, \mathsf{cpk}\right)$ and $\left(\mathsf{sk}, \hat{\mathsf{sk}}, c^*, d^*\right)$, respectively.

The overview of the security proof is as follows. Let $\mathcal{A}$ be an adversary for the KDM$^{(1)}$-CCA security of Π_{kdm}. We consider functions of the form

$$f_e\left(\mathsf{sk}, \hat{\mathsf{sk}}, c^*, d^*\right) = \Lambda_{\mathsf{sk}}(e) + f\left(\hat{\mathsf{sk}}, c^*, d^*\right) \ ,$$

where $e \in \mathcal{C}$ and $f : \hat{\mathcal{SK}} \times \mathcal{C} \times \mathcal{K} \to \mathcal{K}$ is a function. This set of functions includes affine functions in the actual instantiations.

1. We first change the security game so that we do not need sk to simulate KDM queries using the projective property and homomorphism of $\mathcal{H}$ and subgroup indistinguishability. Note that we do not need the smoothness of $\mathcal{H}$ to make this change as explained before.

   After this change, the answer to a KDM query f_e is of the form

   $$\mathsf{Enc}_{\mathsf{cpk}}\left(c \cdot e^{-1}, \mathsf{Pub}(\mathsf{pk}, c, w) + f\left(\hat{\mathsf{sk}}, c^*, d^*\right), \hat{\Lambda}_{\hat{\mathsf{sk}}}\left(c \cdot e^{-1}\right)\right) \ .$$

[3] Without loss of generality, we assume that the secret-key space of Π_{cca} is $\mathcal{K}$.

2. Then, we change the security game so that a decryption query CT made by $\mathcal{A}$ is replied with $\perp$ if $c \notin \mathcal{V}$, where $(c, d, \pi) \leftarrow \mathsf{Dec}_{\mathsf{csk}}(\mathsf{CT})$. The probability that this change affects the behavior of $\mathcal{A}$ is bounded by the probability that $\mathcal{A}$ makes a decryption query CT such that $c \notin \mathcal{V}$ and $\pi = \hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$, where $(c, d, \pi) \leftarrow \mathsf{Dec}_{\mathsf{csk}}(\mathsf{CT})$. We call such a decryption query a *bad decryption query*. Since $\hat{\mathsf{sk}}$ is contained in answers to KDM queries, we cannot estimate the probability at this point. However, as noted above, we can put the estimation off till the end of the sequence of games, and thus we continue the sequence.

3. By the previous change on how decryption queries are replied, we can use the smoothness of $\mathcal{H}$. We eliminate csk encrypted in (c^*, d^*) using the smoothness of $\mathcal{H}$.

4. Then, we can use IND-CCA security of Π_{cca}. We change the security game so that a KDM query made by $\mathcal{A}$ is replied with $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cpk}}(0)$. In this game, the advantage of $\mathcal{A}$ is 0.

To complete the security proof, we need to estimate the probability that $\mathcal{A}$ makes a bad decryption query. In the final game, $\hat{\mathsf{sk}}$ is hidden from the view of $\mathcal{A}$ and he cannot obtain any hash value $\hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$ for $c \notin \mathcal{V}$. Thus, the probability is negligible in the final game if $\hat{\mathcal{H}}$ is 2-universal. In fact, since the universal property of $\hat{\mathcal{H}}$ is sufficient for this argument, we use a universal projective hash function instead of a 2-universal one in the actual construction. Then, the remaining problem is whether the probability that $\mathcal{A}$ makes a bad decryption query changes during the sequence of games.

The probability does not change by the third step since the view of $\mathcal{A}$ before the third step is statistically close to that after the third step from the smoothness of $\mathcal{H}$. In addition, if we can efficiently detect a bad decryption query made by $\mathcal{A}$, we can prove that the probability does not change by the fourth step based on the IND-CCA security of Π_{cca}. For the purpose, in this work, we require there exist a trapdoor that enables us to efficiently check the membership of $\mathcal{V}$ for projective hash function. We can complete the security proof under the existence of such a trapdoor.

1.3.3 Instantiations

We instantiate the above framework based on the DDH, QR, and DCR assumptions by extending the instantiations of KDM$^{(1)}$-CPA secure PKE by Wee [19]. Therefore, the DDH based construction is also an extension of that proposed by Boneh et al. [4], and the QR and DCR based constructions are also extensions of those proposed by Brakerski and Goldwasser [5]. In all constructions, we can make a trapdoor for checking the membership of $\mathcal{V}$. We briefly review the DDH based instantiation.

The DDH based instantiation. In the DDH based instantiation, we set

$$\mathcal{C} = \mathbb{G}^\ell \text{ and } \mathcal{V} = \{(g_1^r, \ldots, g_\ell^r) \,|r \in \mathbb{Z}_p\} \quad ,$$

where $\mathbb{G}$ is a cyclic group of order p, $g_1, \ldots, g_\ell$ are random generators of $\mathbb{G}$, and ℓ is a parameter determined in the analysis. The uniform distribution over $\mathcal{C}$ and $\mathcal{V}$ are computationally indistinguishable based on the DDH assumption on $\mathbb{G}$. Moreover, the discrete logarithms α_i such that $g_i = g^{\alpha_i}$ for every $i \in [\ell]$ can be used as a trapdoor to efficiently decide the membership of $\mathcal{V}$, where g is another generator and $[\ell]$ denotes $\{1, \ldots, \ell\}$.

We construct homomorphic projective hash function $\mathcal{H}$ exactly in the same way as Wee [19]. A secret key sk is randomly chosen $s = s_1 \cdots s_\ell \in \{0,1\}^\ell$. The corresponding public key is $g_0 = \prod_{i \in [\ell]} g_i^{s_i}$. When hashing $c = (c_1, \ldots, c_\ell) \in \mathcal{C}$, we compute $\prod_{i \in [\ell]} c_i^{s_i}$. We see that this

construction satisfies the projective property and homomorphism. Moreover, we can prove the (average-case) smoothness of it based on the leftover hash lemma by taking ℓ appropriately.

We construct a universal projective hash function $\hat{\mathcal{H}}$ as follows. A secret key $\hat{\mathsf{sk}}$ is randomly chosen $(x_1, \ldots, x_\ell) \in \mathbb{Z}_p^\ell$. The corresponding public key is $\hat{g}_0 = \prod_{i\in[\ell]} g_i^{x_i}$. When hashing $c = (c_1, \ldots, c_\ell) \in \mathcal{C}$, we compute $\prod_{i\in[\ell]} c_i^{x_i}$. This construction can be seen as an extension of that proposed by Cramer and Shoup [8], and we can prove its projective property and universal property.

The QR and DCR based instantiations. In the QR based construction, we use the same $\mathcal{C}$, $\mathcal{V}$, and $\mathcal{H}$ as Wee [19]. However, in the QR based construction, we slightly modify how to mask csk in the key generation. Roughly speaking, this is because a hash value of $\mathcal{H}$ uniformly distributes over a group of order 2, and thus we need parallelization in order to mask csk using the smoothness of $\mathcal{H}$. In the modified version of construction, we avoid such parallelization. However, in the construction of a universal projective hash function $\hat{\mathcal{H}}$, we still need a parallelized construction similarly to IND-CCA secure PKE based on the QR assumption proposed by Cramer and Shoup [8]. When we consider CCA security, if the underlying group has a small prime factor, we need a parallelized construction.

In the DCR based construction, we also apply some modifications to the construction of $\mathcal{C}$, $\mathcal{V}$, and $\mathcal{H}$ used by Wee. In the construction of Wee, the underlying group has a small prime factor 2. Therefore, in a naive construction, we need parallelization. However, by defining hash functions so that every time we compute a hash value, we first perform a squaring, we can make the small factor useless to attack the scheme without parallelization. By this modification, the range of hash functions become a group whose order does not have a small prime factor and we can avoid parallelization.

Overhead of our constructions. The overhead of communicational complexity (that is, the size of public-keys and ciphertexts) of our KDM$^{(1)}$-CCA secure PKE schemes from its KDM$^{(1)}$-CPA secure counterparts [19] is very small in the DDH and DCR based constructions. A public-key and hash value of $\hat{\mathcal{H}}$ are just a single group element in the DDH and DCR based constructions. Moreover, we can use highly efficient IND-CCA secure PKE schemes [17, 13] as the outer layer scheme. In this case, the overhead of communicational complexity is only few group elements.

1.3.4 Extension to Multi User Setting

Although the above framework based on projective hash function captures only KDM$^{(1)}$-CCA security, we can prove the KDM-CCA security in the multi user setting of concrete instantiations.

As noted before, our DDH based construction is an extension of that proposed by Boneh et al. [4], and our QR and DCR based constructions are extensions of those proposed by Brakerski and Goldwasser [5]. In both works, they first show the KDM$^{(1)}$-CPA security of their schemes, and then prove its KDM-CPA security in the multi user setting by extending the proof for KDM$^{(1)}$-CPA security.

By using similar techniques, we can prove KDM-CCA security in the multi user setting of our schemes. Especially, we prove the KDM-CCA security in the multi user setting of our DDH based construction with the same parameter setting as in the single user setting. We also briefly explain how to prove the KDM-CCA security in the multi user setting of our QR and DCR based constructions after proving the multi user security of the DDH based construction.

2 Preliminaries

In this section, we define some notations and cryptographic primitives.

2.1 Notations

In this paper, $x \xleftarrow{\mathsf{r}} X$ denotes choosing an element from a finite set X uniformly at random, and $y \leftarrow \mathsf{A}(x)$ denotes assigning to y the output of an algorithm A on an input x. For bit strings x and y, $x\|y$ denotes the concatenation of x and y. For an integer ℓ, $[\ell]$ denotes the set of integers $\{1, \ldots, \ell\}$.

λ denotes a security parameter. PPT stands for probabilistic polynomial time. A function $f(\lambda)$ is a negligible function if $f(\lambda)$ tends to 0 faster than $\frac{1}{\lambda^c}$ for every constant $c > 0$. We write $f(\lambda) = \mathsf{negl}(\lambda)$ to denote $f(\lambda)$ being a negligible function.

2.2 Leftover Hash Lemma

We introduce the leftover hash lemma.

Lemma 1 (Leftover hash lemma) *Let X, Y, and Z are sets. Let $\mathcal{H} := \{h : X \to Y\}$ be a universal hash family. Let $\mathsf{aux} : X \to Z$ be a function. Then, the distributions $(h, h(x), \mathsf{aux}(x))$ and $(h, y, \mathsf{aux}(x))$ are $\sqrt{\frac{|Y||Z|}{4 \cdot |X|}}$-close, where $h \xleftarrow{\mathsf{r}} \mathcal{H}$, $x \xleftarrow{\mathsf{r}} X$, and $y \xleftarrow{\mathsf{r}} Y$.*

2.3 Public Key Encryption

We define public key encryption (PKE).

Definition 1 (Public key encryption) *A PKE scheme PKE is a three tuple $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ of PPT algorithms. Below, let $\mathcal{M}$ be the message space of PKE.*

- *The key generation algorithm KG, given a security parameter 1^λ, outputs a public key pk and a secret key sk.*
- *The encryption algorithm Enc, given a public key pk and message $m \in \mathcal{M}$, outputs a ciphertext CT.*
- *The decryption algorithm Dec, given a secret key sk and ciphertext c, outputs a message $\tilde{m} \in \{\bot\} \cup \mathcal{M}$.*

Correctness *We require $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m$ for every $m \in \mathcal{M}$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$.*

We introduce indistinguishability against chosen ciphertext attacks (IND-CCA security) for PKE.

Definition 2 (IND-CCA security) *Let PKE be a PKE scheme. We define the IND-CCA game between a challenger and an adversary $\mathcal{A}$ as follows. We let $\mathcal{M}$ be the message space of PKE.*

1. *First, the challenger chooses a challenge bit $b \xleftarrow{\mathsf{r}} \{0, 1\}$. Next, the challenger generates a key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$ and sends pk to $\mathcal{A}$.*

   At any step of the game, $\mathcal{A}$ can make decryption queries.

Decryption queries *$\mathcal{A}$ sends CT to the challenger. The challenger returns $m \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{CT})$ to $\mathcal{A}$.*

2. *$\mathcal{A}$ sends $\left(m^0, m^1\right) \in \mathcal{M}^2$ to the challenger. We require that $\left|m^0\right| = \left|m^1\right|$. The challenger computes $\mathsf{CT}^* \leftarrow \mathsf{Enc}\left(\mathsf{pk}, m^b\right)$ and returns CT to $\mathcal{A}$.*

   Below, $\mathcal{A}$ is not allowed to query CT^* as a decryption query.

3. *$\mathcal{A}$ outputs $b' \in \{0,1\}$.*

We say that PKE is IND-CCA secure if for any PPT adversary $\mathcal{A}$, we have $\mathsf{Adv}^{\mathsf{indcca}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right| = \mathsf{negl}(\lambda)$.

Next, we define key dependent message security against chosen ciphertext attacks (KDM-CCA security) for PKE.

Definition 3 (KDM-CCA security) *Let PKE be a PKE scheme, $\mathcal{F}$ function family, and n the number of keys. We define the $\mathcal{F}$-KDM$^{(n)}$-CCA game between a challenger and an adversary $\mathcal{A}$ as follows. Let $\mathcal{SK}$ and $\mathcal{M}$ be the secret key space and message space of PKE, respectively.*

1. *First, the challenger chooses a challenge bit $b \xleftarrow{\mathsf{r}} \{0,1\}$. Next, the challenger generates n key pairs $(\mathsf{pk}_k, \mathsf{sk}_k) \leftarrow \mathsf{KG}(1^\lambda)\,(k \in [n])$. The challenger sets $\mathbf{sk} := (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ and sends $(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$ to $\mathcal{A}$. Finally, the challenger prepares a list L_{kdm} which is initially empty.*

2. *$\mathcal{A}$ may adaptively make the following queries polynomially many times.*

   KDM queries *$\mathcal{A}$ sends $(j, f^0, f^1) \in [n] \times \mathcal{F} \times \mathcal{F}$ to the challenger. We require that f^0 and f^1 be functions such that $f : \mathcal{SK}^n \to \mathcal{M}$. The challenger returns $\mathsf{CT} \leftarrow \mathsf{Enc}\left(\mathsf{pk}_j, f^b(\mathbf{sk})\right)$ to $\mathcal{A}$. Finally, the challenger adds (j, CT) to L_{kdm}.*

   Decryption queries *$\mathcal{A}$ sends (j, CT) to the challenger. If $(j, \mathsf{CT}) \in L_{\mathsf{kdm}}$, the challenger returns $\perp$ to $\mathcal{A}$. Otherwise, the challenger returns $m \leftarrow \mathsf{Dec}\left(\mathsf{sk}_j, \mathsf{CT}\right)$ to $\mathcal{A}$.*

3. *$\mathcal{A}$ outputs $b' \in \{0,1\}$.*

We say that PKE is $\mathcal{F}$-KDM$^{(n)}$-CCA secure if for any PPT adversary $\mathcal{A}$, we have $\mathsf{Adv}^{\mathsf{kdmcca}}_{\mathsf{PKE},\mathcal{F},\mathcal{A},n}(\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right| = \mathsf{negl}(\lambda)$.

In addition, we say that PKE is $\mathcal{F}$-KDM-CCA secure if it is $\mathcal{F}$-KDM$^{(n)}$-CCA secure for any polynomial $n = n(\lambda)$.

Remark 1 (Difference with the previous definitions) In the original definition of KDM security defined by Black et al. [3], an adversary is required to distinguish an encryption of $f(\mathsf{sk})$ from that of some constant message such as 0, where f is a function chosen by the adversary.

In our definition of KDM-CCA security, an adversary chooses two functions (f^0, f^1) and is required to distinguish an encryption of $f^0(\mathsf{sk})$ from that of $f^1(\mathsf{sk})$. Such a definition was previously used by Alperin-sheriff and Peikert [1] when they defined KDM security for identity-based encryption to simplify their security proof. We also adopt this definition to simplify our security proofs.

These two types of definitions are equivalent if the function class $\mathcal{F}$ contains a constant function. This is the case for affine functions and projection functions that we focus on.

2.4 Projective Hash Function

We review the notion of projective hash function introduced by Cramer and Shoup [8] after introducing the notion of subset membership problem.

Definition 4 (Subset membership problem) *Let $\mathcal{C}$ be a group and $\mathcal{V}$ be a subgroup of $\mathcal{C}$. We say that subset membership problem $(\mathcal{C}, \mathcal{V})$ is hard if for any PPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathsf{smp}}_{(\mathcal{C},\mathcal{V}),\mathcal{A}}(\lambda) = \left|\Pr[\mathcal{A}(c) = 1 : c \xleftarrow{\mathsf{r}} \mathcal{C}] - \Pr[\mathcal{A}(c) = 1 : c \xleftarrow{\mathsf{r}} \mathcal{V}]\right| = \mathsf{negl}(\lambda) \ .$$

In this work, for a subset membership problem $(\mathcal{C}, \mathcal{V})$, we require that there exist a trapdoor that enables us to efficiently check the membership of $\mathcal{V}$. Moreover, we require that we can efficiently sample c from $\mathcal{V}$ with a witness of $c \in \mathcal{V}$.

Definition 5 (Projective hash function) *A projective hash function $\mathcal{H}$ is a tuple $(\mathcal{C}, \mathcal{V}, \mathcal{K}, \mathcal{SK}, \mathcal{PK}, \Lambda, \mu)$.*

- *$\mathcal{C}$ and $\mathcal{K}$ are groups and $\mathcal{V}$ is a subgroup of $\mathcal{C}$. $\mathcal{SK}$ and $\mathcal{PK}$ are sets.*
- *The hash function Λ_{sk} indexed by $\mathsf{sk} \in \mathcal{SK}$, given $c \in \mathcal{C}$, outputs a hash value $\mathsf{K} \in \mathcal{K}$.*
- *The projection map μ, given $\mathsf{sk} \in \mathcal{SK}$, outputs $\mathsf{pk} \in \mathcal{PK}$.*

Projective property *We require that for any $\mathsf{sk} \in \mathcal{SK}$ and $c \in \mathcal{V}$, the value of $\Lambda_{\mathsf{sk}}(c)$ is determined only by c and $\mathsf{pk} = \mu(\mathsf{sk})$. In addition, we require that there exist a public evaluation algorithm Pub, given $\mathsf{pk} = \mu(\mathsf{sk})$, $c \in \mathcal{V}$, and a witness w that $c \in \mathcal{V}$, outputs $\Lambda_{\mathsf{sk}}(c)$, where $\mathsf{sk} \in \mathcal{SK}$.*

In addition, if $\mathcal{H}$ satisfies the following homomorphism, $\mathcal{H}$ is said to be a homomorphic projective hash family.

Homomorphism *Let "$\cdot$" and "$+$" denote operations in $\mathcal{C}$ and $\mathcal{K}$, respectively. For all $\mathsf{sk} \in \mathcal{SK}$ and $c_0, c_1 \in \mathcal{C}$, it holds that $\Lambda_{\mathsf{sk}}(c_0 \cdot c_1) = \Lambda_{\mathsf{sk}}(c_0) + \Lambda_{\mathsf{sk}}(c_1)$.*

We define two security notions for projective hash function.

Definition 6 (Average-case smoothness) *Let $\mathcal{H} = (\mathcal{C}, \mathcal{V}, \mathcal{K}, \mathcal{SK}, \mathcal{PK}, \Lambda, \mu)$ be a projective hash function. We say that $\mathcal{H}$ is average-case smooth if the distributions*

$$(\mathsf{pk}, c, \Lambda_{\mathsf{sk}}(c)) \text{ and } (\mathsf{pk}, c, \mathsf{K})$$

are statistically close, where $\mathsf{sk} \xleftarrow{\mathsf{r}} \mathcal{SK}$, $\mathsf{pk} = \mu(\mathsf{sk})$, $c \xleftarrow{\mathsf{r}} \mathcal{C}$, and $\mathsf{K} \xleftarrow{\mathsf{r}} \mathcal{K}$.

Definition 7 (Universal property) *Let $\mathcal{H} = (\mathcal{C}, \mathcal{V}, \mathcal{K}, \mathcal{SK}, \mathcal{PK}, \Lambda, \mu)$ be a projective hash function. We say that $\mathcal{H}$ is universal if for any $\mathsf{pk} \in \mathcal{PK}$, $c \in \mathcal{C} \setminus \mathcal{V}$, $\mathsf{K} \in \mathcal{K}$, we have*

$$\Pr_{\mathsf{sk} \xleftarrow{\mathsf{r}} \mathcal{SK}}[\Lambda_{\mathsf{sk}}(c) = \mathsf{K} | \mu(\mathsf{sk}) = \mathsf{pk}] = \mathsf{negl}(\lambda) \ .$$

3 KDM$^{(1)}$-CCA Secure PKE Based on Homomorphic Projective Hash Function

In this section, we show a framework for achieving KDM$^{(1)}$-CCA secure PKE based on homomorphic projective hash function.

Let $\mathcal{H} = (\mathcal{C}, \mathcal{V}, \mathcal{K}, \mathcal{SK}, \mathcal{PK}, \Lambda, \mu)$ be a homomorphic projective hash function with a public evaluation algorithm Pub. We denote the group operations of $\mathcal{C}$ and $\mathcal{K}$ by "$\cdot$" and "$+$", respectively. Let $\hat{\mathcal{H}} = \left(\mathcal{C}, \mathcal{V}, \hat{\mathcal{K}}, \hat{\mathcal{SK}}, \hat{\mathcal{PK}}, \hat{\Lambda}, \hat{\mu}\right)$ be a projective hash function with a public evaluation algorithm $\hat{\mathsf{Pub}}$. Let $\Pi_{\mathsf{cca}} = (\mathsf{KG}_{\mathsf{cca}}, \mathsf{Enc}_{\mathsf{cca}}, \mathsf{Dec}_{\mathsf{cca}})$ be a PKE scheme. We assume that the secret-key space of Π_{cca} is $\mathcal{K}$ for simplicity. Using these building blocks, we construct the following PKE scheme $\Pi_{\mathsf{kdm}} = (\mathsf{KG}_{\mathsf{kdm}}, \mathsf{Enc}_{\mathsf{kdm}}, \mathsf{Dec}_{\mathsf{kdm}})$. The message space of Π_{kdm} is $\mathcal{M}$. We use an invertible map $\phi : \mathcal{M} \to \mathcal{K}$ in the construction.

$\mathsf{KG}_{\mathsf{kdm}}(1^\lambda)$:

- Generate $\mathsf{sk} \xleftarrow{\mathsf{r}} \mathcal{SK}$ and compute $\mathsf{pk} \leftarrow \mu(\mathsf{sk})$.
- Generate $\hat{\mathsf{sk}} \xleftarrow{\mathsf{r}} \hat{\mathcal{SK}}$ and compute $\hat{\mathsf{pk}} \leftarrow \hat{\mu}\left(\hat{\mathsf{sk}}\right)$.
- Generate $(\mathsf{cpk}, \mathsf{csk}) \leftarrow \mathsf{KG}_{\mathsf{cca}}(1^\lambda)$.
- Generate $c^* \xleftarrow{\mathsf{r}} \mathcal{C}$ and compute $d^* \leftarrow \Lambda_{\mathsf{sk}}(c^*) + \mathsf{csk}$.
- Return $\mathsf{PK} := \left(\mathsf{pk}, \hat{\mathsf{pk}}, \mathsf{cpk}\right)$ and $\mathsf{SK} := \left(\mathsf{sk}, \hat{\mathsf{sk}}, c^*, d^*\right)$.

$\mathsf{Enc}_{\mathsf{kdm}}(\mathsf{PK}, m)$:

- Parse $\left(\mathsf{pk}, \hat{\mathsf{pk}}, \mathsf{cpk}\right) \leftarrow \mathsf{PK}$.
- Generate $c \xleftarrow{\mathsf{r}} \mathcal{V}$ with an witness w of $c \in \mathcal{V}$.
- Compute $\mathsf{K} \leftarrow \mathsf{Pub}(\mathsf{pk}, c, w)$ and $d \leftarrow \mathsf{K} + \phi(m)$.
- Compute $\pi \leftarrow \hat{\mathsf{Pub}}\left(\hat{\mathsf{pk}}, c, w\right)$.
- Return $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cca}}\left(\mathsf{cpk}, (c, d, \pi)\right)$.

$\mathsf{Dec}_{\mathsf{kdm}}(\mathsf{SK}, \mathsf{CT})$:

- Parse $\left(\mathsf{sk}, \hat{\mathsf{sk}}, c^*, d^*\right) \leftarrow \mathsf{SK}$.
- Compute $\mathsf{csk} \leftarrow d^* - \Lambda_{\mathsf{sk}}(c^*)$.
- Compute $(c, d, \pi) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{csk}, \mathsf{CT})$. If the decryption result is not in $\mathcal{C} \times \mathcal{K} \times \hat{\mathcal{K}}$, return $\perp$. Otherwise, compute as follows.
- If $\pi \neq \hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$, return $\perp$. Otherwise, return $m \leftarrow \phi^{-1}\left(d - \Lambda_{\mathsf{sk}}(c)\right)$.

Correctness. We have $\mathsf{Pub}(\mathsf{pk}, c, w) = \Lambda_{\mathsf{sk}}(c)$ and $\hat{\mathsf{Pub}}\left(\hat{\mathsf{pk}}, c, w\right) = \hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$ for $c \in \mathcal{V}$, where w is a witness of $c \in \mathcal{V}$. Then, the correctness of Π_{kdm} follows from that of Π_{cca}.

Π_{kdm} is KDM-CCA secure with respect to the function family $\mathcal{F}_{\mathsf{phf}}$ consisting of functions described as

$$f_e\left(\mathsf{sk}, \hat{\mathsf{sk}}, c^*, d^*\right) = \phi^{-1}\left(\Lambda_{\mathsf{sk}}(e) + \phi\left(f\left(\hat{\mathsf{sk}}, c^*, d^*\right)\right)\right) \quad ,$$

where $e \in \mathcal{C}$ and $f : \hat{\mathcal{SK}} \times \mathcal{C} \times \mathcal{K} \to \mathcal{M}$ is a function. This class corresponds to affine and projection functions in the instantiations. Formally, we prove the following theorem.

Theorem 3 *Let the subset membership problem $(\mathcal{C}, \mathcal{V})$ be hard. Let $\mathcal{H}$ be average-case smooth and $\hat{\mathcal{H}}$ universal. Let Π_{cca} be IND-CCA secure. Then, Π_{kdm} is $\mathcal{F}_{\mathsf{phf}}$-KDM$^{(1)}$-CCA secure.*

Proof of Theorem 3. Let $\mathcal{A}$ be an adversary that attacks the $\mathcal{F}_{\mathsf{phf}}$-KDM$^{(1)}$-CCA security of Π_{kdm}. We proceed the proof via a sequence of games. For every $t \in \{0, \ldots, 8\}$, let $\mathtt{SUC}_t$ be the event that $\mathcal{A}$ succeeds in guessing the challenge bit b in Game t.

Game 0: This is the original $\mathcal{F}_{\mathsf{phf}}$-KDM$^{(1)}$-CCA game regarding Π_{kdm}. We have $\mathsf{Adv}^{\mathsf{kdmcca}}_{\Pi_{\mathsf{kdm}},\mathcal{F}_{\mathsf{phf}},\mathcal{A},1}(\lambda) = \left|\Pr[\mathtt{SUC}_0] - \frac{1}{2}\right|$. The detailed description is as follows.

1. The challenger chooses a challenge bit $b \xleftarrow{\mathsf{r}} \{0,1\}$, and runs as follows.
   (a) Generate $\mathsf{sk} \xleftarrow{\mathsf{r}} \mathcal{SK}$ and compute $\mathsf{pk} \leftarrow \mu(\mathsf{sk})$.
   (b) Generate $\hat{\mathsf{sk}} \xleftarrow{\mathsf{r}} \hat{\mathcal{SK}}$ and compute $\hat{\mathsf{pk}} \leftarrow \hat{\mu}\left(\hat{\mathsf{sk}}\right)$.
   (c) Generate $(\mathsf{cpk}, \mathsf{csk}) \leftarrow \mathsf{KG}_{\mathsf{cca}}\left(1^\lambda\right)$.
   (d) Generate $c^* \xleftarrow{\mathsf{r}} \mathcal{C}$ and compute $d^* \leftarrow \Lambda_{\mathsf{sk}}(c^*) + \mathsf{csk}$.
   (e) Send $\mathsf{PK} := \left(\mathsf{pk}, \hat{\mathsf{pk}}, \mathsf{cpk}\right)$ to $\mathcal{A}$ and prepare a list L_{kdm}.
2. The challenger responds to queries made by $\mathcal{A}$.
   For a KDM query $((e^0, f^0), (e^1, f^1))$ made by $\mathcal{A}$, the challenger responds as follows.
   (a) Generate $c \xleftarrow{\mathsf{r}} \mathcal{V}$ with a witness w of $c \in \mathcal{V}$.
   (b) Compute $\mathsf{K} \leftarrow \mathsf{Pub}(\mathsf{pk}, c, w)$ and $d \leftarrow \mathsf{K} + \Lambda_{\mathsf{sk}}(e^b) + \phi\left(f^b\left(\hat{\mathsf{sk}}, c^*, d^*\right)\right)$.
   (c) Compute $\pi \leftarrow \hat{\mathsf{Pub}}\left(\hat{\mathsf{pk}}, c, w\right)$.
   (d) Return $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cca}}\left(\mathsf{cpk}, (c, d, \pi)\right)$ to $\mathcal{A}$ and add CT to L_{kdm}.
   For a decryption query CT made by $\mathcal{A}$, the challenger returns $\perp$ to $\mathcal{A}$ if $\mathsf{CT} \in L_{\mathsf{kdm}}$, and otherwise responds as follows.
   (a) Compute $(c, d, \pi) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{csk}, \mathsf{CT})$. If the decryption result is not in $\mathcal{C} \times \mathcal{K} \times \hat{\mathcal{K}}$, return $\perp$ to $\mathcal{A}$. Otherwise, responds as follows.
   (b) Return $\perp$ if $\pi \neq \hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$ and $m \leftarrow \phi^{-1}\left(d - \Lambda_{\mathsf{sk}}(c)\right)$ otherwise.
3. $\mathcal{A}$ outputs $b' \in \{0,1\}$.

Game 1: Same as Game 0 except that when $\mathcal{A}$ makes a KDM query, the challenger computes $\mathsf{K} \leftarrow \Lambda_{\mathsf{sk}}(c)$ and $\pi \leftarrow \hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$ instead of $\mathsf{K} \leftarrow \mathsf{Pub}(\mathsf{pk}, c, w)$ and $\pi \leftarrow \hat{\mathsf{Pub}}\left(\hat{\mathsf{pk}}, c, w\right)$, respectively.

Due to the projective property of $\mathcal{H}$ and $\hat{\mathcal{H}}$, this change is only conceptual and thus we have $|\Pr[\mathtt{SUC}_0] - \Pr[\mathtt{SUC}_1]| = 0$.

Game 2: Same as Game 1 except that when $\mathcal{A}$ makes a KDM query, the challenger generates $c \xleftarrow{\mathsf{r}} \mathcal{C}$.

We have $|\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]| = \mathsf{negl}(\lambda)$ by the hardness of the subset membership problem $(\mathcal{C}, \mathcal{V})$.

Game 3: Same as Game 2 except that when $\mathcal{A}$ makes a KDM query $((e^0, f^0), (e^1, f^1))$, the challenger generates $c \xleftarrow{\mathsf{r}} \mathcal{C}$ and uses $c' = c \cdot (e^b)^{-1}$ instead of c.

We have $|\Pr[\mathtt{SUC}_2] - \Pr[\mathtt{SUC}_3]| = 0$ since if c uniformly distributes over $\mathcal{C}$, then so does $c \cdot (e^b)^{-1}$.

By this change, the answer to a KDM query $((e^0, f^0), (e^1, f^1))$ in Game 3 is $\mathsf{Enc}_{\mathsf{cca}}(\mathsf{cpk}, (c', d, \pi))$, where

$$c' = c \cdot (e^b)^{-1}, d = \Lambda_{\mathsf{sk}}(c \cdot (e^b)^{-1}) + \Lambda_{\mathsf{sk}}(e^b) + \phi\left(f^b\left(\hat{\mathsf{sk}}, c^*, d^*\right)\right), \pi = \hat{\Lambda}_{\hat{\mathsf{sk}}}(c') \quad ,$$

and $c \xleftarrow{\mathsf{r}} \mathcal{C}$. Moreover, by the homomorphism of $\mathcal{H}$, $d = \Lambda_{\mathsf{sk}}(c) + \phi\left(f^b\left(\hat{\mathsf{sk}}, c^*, d^*\right)\right)$ holds.

Game 4: Same as Game 3 except that when $\mathcal{A}$ makes a KDM query, the challenger generates $c \xleftarrow{\mathsf{r}} \mathcal{V}$ with a witness w of $c \in \mathcal{V}$.

We have $|\Pr[\mathtt{SUC}_3] - \Pr[\mathtt{SUC}_4]| = \mathsf{negl}(\lambda)$ by the hardness of the subset membership problem $(\mathcal{C}, \mathcal{V})$.

Game 5: Same as Game 4 except that when $\mathcal{A}$ makes a KDM query, the challenger computes $d \leftarrow \mathsf{Pub}(\mathsf{pk}, c, w) + \phi\left(f^b\left(\hat{\mathsf{sk}}, c^*, d^*\right)\right)$. Note that the challenger still computes π with $\pi \leftarrow \hat{\Lambda}_{\hat{\mathsf{sk}}}(c')$.

Due to the projective property of $\mathcal{H}$, this change is only conceptual and thus we have $|\Pr[\mathtt{SUC}_4] - \Pr[\mathtt{SUC}_5]| = 0$.

At this point, sk is not needed to compute answers to KDM queries. More precisely, the answer to a KDM query $((e^0, f^0), (e^1, f^1))$ is $\mathsf{Enc}_{\mathsf{cca}}(\mathsf{cpk}, (c', d, \pi))$, where

$$c' = c \cdot (e^b)^{-1}, d = \mathsf{Pub}(\mathsf{pk}, c, w) + \phi\left(f^b\left(\hat{\mathsf{sk}}, c^*, d^*\right)\right), \pi = \hat{\Lambda}_{\hat{\mathsf{sk}}}(c') \quad ,$$

$c \xleftarrow{\mathsf{r}} \mathcal{V}$, and w is a witness of $c \in \mathcal{V}$.

Game 6: Same as Game 5 except how the challenger responds decryption queries made by $\mathcal{A}$. In this game, the challenger returns $\perp$ for a decryption query related to $c \notin \mathcal{V}$. More precisely, the challenger responds as follows.

For a decryption query CT made by $\mathcal{A}$, the challenger returns $\perp$ to $\mathcal{A}$ if $\mathsf{CT} \in L_{\mathsf{kdm}}$, and otherwise responds as follows.

1. Compute $(c, d, \pi) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{csk}, \mathsf{CT})$. If the decryption result is not in $\mathcal{V} \times \mathcal{K} \times \hat{\mathcal{K}}$, return $\perp$ to $\mathcal{A}$. Otherwise, respond as follows.
2. Return $\perp$ if $\pi \neq \hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$ and $m \leftarrow \phi^{-1}(d - \Lambda_{\mathsf{sk}}(c))$ otherwise.

We define the following event in Game t ($t = 5, \cdots, 8$).

$\mathtt{BDQ}_t$**:** $\mathcal{A}$ makes a decryption query $\mathsf{CT} \notin L_{\mathsf{kdm}}$ which satisfies $c \in \mathcal{C} \setminus \mathcal{V}$ and $\pi = \hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$, where $(c, d, \pi) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{csk}, \mathsf{CT})$. We call such a decryption query a "**bad decryption query**".

Games 5 and 6 are identical games unless $\mathcal{A}$ makes a bad decryption query in each game. Therefore, we have $|\Pr[\mathtt{SUC}_5] - \Pr[\mathtt{SUC}_6]| \leq \Pr[\mathtt{BDQ}_6]$.

Below, we let td be a trapdoor for efficiently deciding the membership of $\mathcal{V}$.

Game 7: Same as Game 6 except that the challenger generates $d^* \xleftarrow{\mathsf{r}} \mathcal{K}$.

By the previous change, $\mathcal{A}$ cannot obtain information of sk other than pk through decryption queries in Games 6 and 7. Moreover, as noted above, KDM queries are replied without using sk in Games 6 and 7. Thus, the view of $\mathcal{A}$ in Games 6 and 7 can be perfectly simulated by $(\mathsf{pk}, c^*, \Lambda_{\mathsf{sk}}(c^*))$ and (pk, c^*, d^*), respectively, where $\mathsf{sk} \xleftarrow{\mathsf{r}} \mathcal{SK}$, $\mathsf{pk} \leftarrow \mu(\mathsf{sk})$, $c^* \xleftarrow{\mathsf{r}} \mathcal{C}$, and $d^* \xleftarrow{\mathsf{r}} \mathcal{K}$. Therefore, we have $|\Pr[\mathtt{SUC}_6] - \Pr[\mathtt{SUC}_7]| = \mathsf{negl}(\lambda)$ and $|\Pr[\mathtt{BDQ}_6] - \Pr[\mathtt{BDQ}_7]| = \mathsf{negl}(\lambda)$ from the average-case smoothness of $\mathcal{H}$.[4]

csk is now eliminated from the view of $\mathcal{A}$. Thus, we can use IND-CCA security of Π_{cca}.

Game 8**:** Same as Game 7 except that when $\mathcal{A}$ makes a KDM query, the challenger computes $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cca}}\left(\mathsf{cpk}, \left(1_{\mathcal{C}}, 1_{\mathcal{K}}, 1_{\hat{\mathcal{K}}}\right)\right)$, where $1_{\mathcal{C}}$, $1_{\mathcal{K}}$, and $1_{\hat{\mathcal{K}}}$ are identity elements of $\mathcal{C}$, $\mathcal{K}$, and $\hat{\mathcal{K}}$, respectively.

From the IND-CCA security of Π_{cca}, we have $|\Pr[\mathtt{SUC}_7] - \Pr[\mathtt{SUC}_8]| = \mathsf{negl}(\lambda)$.

Moreover, since reduction algorithms for IND-CCA security of Π_{cca} can detect a bad decryption query made by $\mathcal{A}$ by utilizing decryption queries, td, and $\hat{\mathsf{sk}}$, we obtain $|\Pr[\mathtt{BDQ}_7] - \Pr[\mathtt{BDQ}_8]| = \mathsf{negl}(\lambda)$ from the IND-CCA security of Π_{cca}.

We see that the value of b is information theoretically hidden from the view of $\mathcal{A}$ in Game 8. Thus, we have $\left|\Pr[\mathtt{SUC}_8] - \frac{1}{2}\right| = 0$.

We estimate $\Pr[\mathtt{BDQ}_8]$. In Game 8, $\hat{\mathsf{sk}}$ is hidden from the view of $\mathcal{A}$ except $\hat{\mathsf{pk}}$. Moreover, $\mathcal{A}$ cannot obtain any hash value $\hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$ for $c \in \mathcal{C} \setminus \mathcal{V}$ since an answer to a KDM query is computed as $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cca}}\left(\mathsf{cpk}, \left(1_{\mathcal{C}}, 1_{\mathcal{K}}, 1_{\hat{\mathcal{K}}}\right)\right)$ in Game 8. Therefore, from the universal property of $\hat{\mathcal{H}}$, we obtain $\Pr[\mathtt{BDQ}_8] = \mathsf{negl}(\lambda)$.[5]

From the above arguments, we see that

$$
\begin{aligned}
\mathsf{Adv}^{\mathsf{kdmcca}}_{\Pi_{\mathsf{kdm}}, \mathcal{F}_{\mathsf{phf}}, \mathcal{A}, 1}(\lambda) &= \left|\Pr[\mathtt{SUC}_0] - \frac{1}{2}\right| \\
&= \sum_{t=0}^{7} |\Pr[\mathtt{SUC}_t] - \Pr[\mathtt{SUC}_{t+1}]| + \left|\Pr[\mathtt{SUC}_8] - \frac{1}{2}\right| \\
&= \sum_{t \in \{0,\ldots,7\}, t \neq 5} |\Pr[\mathtt{SUC}_t] - \Pr[\mathtt{SUC}_{t+1}]| + |\Pr[\mathtt{SUC}_5] - \Pr[\mathtt{SUC}_6]| \\
&= \sum_{t \in \{0,\ldots,7\}, t \neq 5} |\Pr[\mathtt{SUC}_t] - \Pr[\mathtt{SUC}_{t+1}]| + \sum_{t=6}^{7} |\Pr[\mathtt{BDQ}_t] - \Pr[\mathtt{BDQ}_{t+1}]| + \Pr[\mathtt{BDQ}_8] \\
&= \mathsf{negl}(\lambda) \ .
\end{aligned}
$$

Since the choice of $\mathcal{A}$ is arbitrary, Π_{kdm} is $\mathcal{F}_{\mathsf{phf}}$-KDM$^{(1)}$-CCA secure. □ (**Theorem 3**)

Remark 2 (Shrink secret keys) We do not need to require any structure and homomorphism for $\hat{\mathsf{sk}}$ and csk. Then, we can shrink them into a single pseudorandom function key K_{prf}

[4] In terms of reduction, we can construct a computationally unbounded adversary $\mathcal{B}$ that given (pk, c^*, d^*), distinguishes whether $d^* \leftarrow \Lambda_{\mathsf{sk}}(c^*)$ or $d^* \xleftarrow{\mathsf{r}} \mathcal{K}$ using $\mathcal{A}$. The only non-trivial part of the construction of $\mathcal{B}$ is how $\mathcal{B}$ responds to decryption queries made by $\mathcal{A}$. After Game 6, bad decryption queries made by $\mathcal{A}$ are replied with $\perp$. In addition, if a decryption query is not a bad decryption query, computationally unbounded $\mathcal{B}$ can reply to the decryption query correctly without using sk. This is done by extracting a witness related to the decryption query and computing the hash value with Pub.

[5] Similarly to the estimation of $|\Pr[\mathtt{SUC}_6] - \Pr[\mathtt{SUC}_7]|$, in terms of reduction, we can construct a computationally unbounded reduction that responds a decryption query from $\mathcal{A}$ correctly without knowing $\hat{\mathsf{sk}}$ by extracting a witness and using $\hat{\mathsf{Pub}}$.

and modify the construction so that $\Lambda_{\mathsf{sk}}(c^*)$ masks K_{prf}. Moreover, we can maintain c^* and $d^* = \Lambda_{\mathsf{sk}}(c^*) + K_{\mathsf{prf}}$ as a part of the corresponding public key. If we do so, the resulting secret key is just sk.

4 Instantiations

We show concrete instantiations of our framework described in Section 3.

4.1 Instantiation Based on the DDH Assumption

Definition 8 (DDH assumption) *Let $\mathbb{G}$ be a cyclic group of order p. Let g be a random generator of $\mathbb{G}$. For any PPT algorithm $\mathcal{A}$, we have*

$$|\Pr[\mathcal{A}(p, g, g^x, g^y, g^{xy}) = 1] - \Pr[\mathcal{A}(p, g, g^x, g^y, g^z) = 1] = \mathsf{negl}(\lambda)| \ ,$$

where $x, y, z \xleftarrow{\mathsf{r}} \mathbb{Z}_p$.

Let $\mathbb{G}$ be a cyclic group of order p and $g_1, \ldots, g_\ell$ random generators of $\mathbb{G}$, where ℓ is determined later. We define $\mathcal{C}$ and $\mathcal{V}$ as

$$\mathcal{C} = \mathbb{G}^\ell, \text{ and } \mathcal{V} = \{(g_1^r, \ldots, g_\ell^r) \,|r \in \mathbb{Z}_p\} \ .$$

$\mathcal{V}$ is a subgroup of $\mathcal{C}$ and subset membership problem $(\mathcal{C}, \mathcal{V})$ is hard under the DDH assumption on $\mathbb{G}$ [4, 19]. Let g be another generator of $\mathbb{G}$. Then, there exists $\alpha_i \in \mathbb{Z}_p^*$ such that $g^{\alpha_i} = g_i$ for every $i \in [\ell]$. The trapdoor for checking the membership of $\mathcal{V}$ is $\{\alpha_i\}_{i\in[\ell]}$.

For $(\mathcal{C}, \mathcal{V})$ defined above, we construct two projective hash functions $\mathcal{H} = (\mathcal{C}, \mathcal{V}, \mathcal{K}, \mathcal{SK}, \mathcal{PK}, \Lambda, \mu)$ and $\hat{\mathcal{H}} = \left(\mathcal{C}, \mathcal{V}, \hat{\mathcal{K}}, \hat{\mathcal{SK}}, \hat{\mathcal{PK}}, \hat{\Lambda}, \hat{\mu}\right)$. The construction of $\mathcal{H}$ is the same as that of projective hash function based on the DDH assumption proposed by Wee [19] thus is a generalization of the KDM-CPA secure PKE scheme proposed by Boneh et al. [4].

4.1.1 Construction of $\mathcal{H}$

We define $\mathcal{SK} = \{0,1\}^\ell$, $\mathcal{PK} = \mathbb{G}$, and $\mathcal{K} = \mathbb{G}$. For every $\mathsf{sk} = s_1 \cdots s_\ell \in \{0,1\}^\ell$ and $c = (c_1, \ldots, c_\ell) \in \mathcal{C}$, we also define μ and Λ as

$$\mu(\mathsf{sk}) = \prod_{i\in[\ell]} g_i^{s_i} \quad \text{and} \quad \Lambda_{\mathsf{sk}}(c) = \prod_{i\in[\ell]} c_i^{s_i} \ .$$

Projective property. For every $\mathsf{sk} = s_1 \cdots s_\ell \in \{0,1\}^\ell$, $\mathsf{pk} = \prod_{i\in[\ell]} g_i^{s_i}$, and $c = (g_1^r, \ldots, g_\ell^r)$, where $r \in \mathbb{Z}_p$, we define the public evaluation algorithm Pub as

$$\mathsf{Pub}(\mathsf{pk}, c, r) = \mathsf{pk}^r \ .$$

We see that

$$\mathsf{pk}^r = \left(\prod_{i\in[\ell]} g_i^{s_i}\right)^r = \prod_{i\in[\ell]} (g_i^r)^{s_i} = \Lambda_{\mathsf{sk}}(c)$$

and thus $\mathcal{H}$ satisfies projective property.

Homomorphism. For every $\mathsf{sk} = s_1 \cdots s_\ell \in \{0,1\}^\ell$, $c = (c_1, \ldots, c_\ell) \in \mathcal{C}$, and $c' = (c'_1, \ldots, c'_\ell) \in \mathcal{C}$, we have

$$\Lambda_{\mathsf{sk}}(c) \cdot \Lambda_{\mathsf{sk}}(c') = \prod_{i \in [\ell]} c_i^{s_i} \cdot \prod_{i \in [\ell]} (c'_i)^{s_i} = \prod_{i \in [\ell]} (c_i \cdot c'_i)^{s_i} = \Lambda_{\mathsf{sk}}(c \cdot c')$$

and thus $\mathcal{H}$ is homomorphic.

Average-case smoothness. The leftover hash lemma implies that the following two distributions

$$\left(c, \prod_{i \in [\ell]} c_i^{s_i}, \prod_{i \in [\ell]} g_i^{s_i} \right) \text{ and } \left(c, \mathsf{K}, \prod_{i \in [\ell]} g_i^{s_i} \right)$$

are $\sqrt{\frac{p \cdot p}{4 \cdot 2^\ell}}$-close, where $\mathsf{sk} = s_1 \cdots s_\ell \xleftarrow{\mathsf{r}} \{0,1\}^\ell$, $c = (c_1, \ldots, c_\ell) \xleftarrow{\mathsf{r}} \mathcal{C}$, and $\mathsf{K} \xleftarrow{\mathsf{r}} \mathcal{K}$. Therefore, by setting $\ell = 3 \log p$, $\mathcal{H}$ satisfies average-case smoothness.

4.1.2 Construction of $\hat{\mathcal{H}}$

We define $\hat{\mathcal{SK}} = \mathbb{Z}_p^\ell$, $\hat{\mathcal{PK}} = \mathbb{G}$, and $\hat{\mathcal{K}} = \mathbb{G}$. For every $\hat{\mathsf{sk}} = (x_1, \ldots, x_\ell) \in \hat{\mathcal{SK}}$ and $c = (c_1, \ldots, c_\ell) \in \mathcal{C}$, we also define $\hat{\mu}$ and $\hat{\Lambda}$ as

$$\hat{\mu}\left(\hat{\mathsf{sk}}\right) = \prod_{i \in [\ell]} g_i^{x_i} \text{ and } \hat{\Lambda}_{\hat{\mathsf{sk}}}(c) = \prod_{i \in [\ell]} c_i^{x_i} \ .$$

Projective property. For every $\hat{\mathsf{sk}} = (x_1, \ldots, x_\ell) \in \mathbb{Z}_p^\ell$, $\hat{\mathsf{pk}} = \prod_{i \in [\ell]} g_i^{x_i}$, and $c = (g_1^r, \ldots, g_\ell^r)$, where $r \in \mathbb{Z}_p$, we define the public evaluation algorithm $\hat{\mathsf{Pub}}$ as

$$\hat{\mathsf{Pub}}\left(\hat{\mathsf{pk}}, c, r\right) = \hat{\mathsf{pk}}^r \ .$$

Similarly to $\mathcal{H}$, we see that $\hat{\mathcal{H}}$ satisfies projective property.

Universal property. For every $\hat{\mathsf{pk}} \in \hat{\mathcal{PK}}$, $c = \left(g_1^{r_1}, \ldots, g_\ell^{r_\ell}\right) \in \mathcal{C} \setminus \mathcal{V}$, and $\pi \in \hat{\mathcal{K}}$, we consider the following probability

$$\Pr_{x_i \xleftarrow{\mathsf{r}} \mathbb{Z}_p} \left[\prod_{i \in [\ell]} c_i^{x_i} = \pi \middle| \prod_{i \in [\ell]} g_i^{x_i} = \hat{\mathsf{pk}} \right] \ . \tag{1}$$

Let g be a generator of $\mathbb{G}$. There exists $\alpha_i \in \mathbb{Z}_p^*$ such that $g_i = g^{\alpha_i}$ for every $i \in [\ell]$. Then, the above probability of Equation 1 is the same as

$$\Pr_{x_i \xleftarrow{\mathsf{r}} \mathbb{Z}_p} \left[\sum_{i \in [\ell]} \alpha_i x_i r_i \bmod p = \log_g \pi \middle| \sum_{i \in [\ell]} \alpha_i x_i \bmod p = \log_g \hat{\mathsf{pk}} \right] \ .$$

Since $c \notin \mathcal{V}$, there exist $i_1, i_2 \in [\ell]$ such that $r_{i_1} \neq r_{i_2}$. Therefore, two equations

$$\sum_{i \in [\ell]} \alpha_i x_i r_i \bmod p = \log_g \pi \text{ and } \sum_{i \in [\ell]} \alpha_i x_i \bmod p = \log_g \hat{\mathsf{pk}}$$

are linearly independent. Thus, the probability of Equation 1 is $\frac{1}{p}$. This means that $\hat{\mathcal{H}}$ is universal.

4.1.3 Associated Function Class

The message space of the DDH based construction is $\{0,1\}$. We define $\phi(m) = g^m$. Since we restrict the message space to $\{0,1\}$, we can compute ϕ^{-1} in the brute-force manner. Let $\mathcal{F}_{\mathsf{ddh}}$ be a function family consisting of functions described as

$$\begin{aligned} f_e\left(\mathsf{sk}, \hat{\mathsf{sk}}, c^*, d^*\right) &= \phi^{-1}\left(\Lambda_{\mathsf{sk}}\left(g^{e_1}, \ldots, g^{e_\ell}\right) \cdot \phi\left(f\left(\hat{\mathsf{sk}}, c^*, d^*\right)\right)\right) \\ &= \phi^{-1}\left(g^{\sum_{i\in[\ell]} e_i s_i + f\left(\hat{\mathsf{sk}}, c^*, d^*\right)}\right) \\ &= \sum_{i\in[\ell]} e_i s_i + f\left(\hat{\mathsf{sk}}, c^*, d^*\right) \ , \end{aligned}$$

where $e_i \in \{0,1\}$ for every $i \in [\ell]$ and f is a function of single-bit output such that $\sum_{i\in[\ell]} e_i s_i + f\left(\hat{\mathsf{sk}}, c^*, d^*\right) \in \{0,1\}$ for all sk, $\hat{\mathsf{sk}}$, c^*, and d^*. The DDH based construction is $\mathcal{F}_{\mathsf{ddh}}$-KDM$^{(1)}$-CCA secure. In the construction, we can maintain $\hat{\mathsf{sk}}$, c^*, and d^* as bit strings. In this case, the above function class includes projection functions of single-bit output.

Remark 3 (Extension to affine functions) We can construct a DDH based PKE scheme that is KDM-CCA secure with respect to affine functions by applying the following modifications to the above construction. We set the message space as $\mathbb{G}$. Let $\mathsf{SK} \in \{0,1\}^L$ be a bit string that is a concatenation of sk and bit representation of $\left(\hat{\mathsf{sk}}, c^*, d^*\right)$. We maintain a secret-key $\mathsf{SK} = s_1 \cdots s_L$ as $(g^{s_1}, \ldots, g^{s_L})$. Then, the construction is $\mathcal{F}_{\mathsf{aff}}$-KDM$^{(1)}$-CCA secure, where $\mathcal{F}_{\mathsf{aff}}$ is a function class consisting of functions described as

$$f(g^{s_1}, \ldots, g^{s_L}) = \left(\prod_{i\in[L]} (g^{s_i})^{a_i}\right) \cdot a_0 \ ,$$

where $a_0 \in \mathbb{G}$ and $a_1, \ldots, a_L \in \mathbb{Z}_p$. This class is exactly the class of affine functions defined by Boneh et al. [4].

4.2 Instantiation Based on the QR Assumption

Definition 9 (QR assumption) *Let $N = PQ$ be a Blum integer for λ-bit safe primes $P, Q \equiv 3 \pmod 4$ such that $P = 2p+1$ and $Q = 2q+1$ for primes p and q. Let $n = pq$. We use $\mathbb{J}_N$ to denote the set of elements in $\mathbb{Z}_N^*$ with Jacobi symbol 1. We also use $\mathbb{QR}_N$ to denote the set of quadratic residues modulo N. $\mathbb{J}_N$ and $\mathbb{QR}_N$ are multiplicative groups of order $\frac{\varphi(N)}{2} = 2n$ and $\frac{\varphi(N)}{4} = n$, respectively.*

We say that the QR assumption holds if for any PPT algorithm $\mathcal{A}$, we have

$$\left|\Pr[\mathcal{A}(N, y) = 1] - \Pr[\mathcal{A}(N, y') = 1] = \mathsf{negl}(\lambda)\right| \ ,$$

where $y \xleftarrow{\mathsf{r}} \mathbb{J}_N$ and $y' \xleftarrow{\mathsf{r}} \mathbb{QR}_N$.

We define N, $\mathbb{J}_N$, and $\mathbb{QR}_N$ as in Definition 9. Then, we can decompose $\mathbb{J}_N$ as an internal direct product $\langle -1\rangle \otimes \mathbb{QR}_N$, and $\mathbb{QR}_N$ is a cyclic group, where $\langle -1\rangle$ is the subgroup of $\mathbb{Z}_N^*$ generated by $-1 \bmod N$. Let $g_1, \ldots, g_\ell$ be random generators of $\mathbb{QR}_N$, where ℓ is determined later. We can generate a random generator g of $\mathbb{QR}_N$ by generating $\mu \xleftarrow{\mathsf{r}} \mathbb{Z}_N^*$ and setting $g = \mu^2 \bmod N$. Then, g is a generator of $\mathbb{QR}_N$ with overwhelming probability.

We define $\mathcal{C}$ and $\mathcal{V}$ as

$$\mathcal{C} = \left\{ \left((-1)^{d_1} \cdot g_1^r, \ldots, (-1)^{d_\ell} \cdot g_\ell^r \right) \middle| d_1, \ldots, d_\ell \in \mathbb{Z}_2, r \in \mathbb{Z}_n \right\} \ , \text{ and}$$
$$\mathcal{V} = \{ (g_1^r, \ldots, g_\ell^r) \,|r \in \mathbb{Z}_n \} \ .$$

$\mathcal{V}$ is a subgroup of $\mathcal{C}$ and subset membership problem of $(\mathcal{C}, \mathcal{V})$ is hard under the QR assumption [5, 19]. Let g be another generator of $\mathbb{QR}_N$. Then, there exists $\alpha_i \in \mathbb{Z}_n^*$ such that $g^{\alpha_i} = g_i$ for every $i \in [\ell]$. The trapdoor for checking the membership of $\mathcal{V}$ is P, Q, and $\{\alpha_i\}_{i \in [\ell]}$.

When sampling a random element $c = (c_1, \ldots, c_\ell)$ from $\mathcal{V}$, we randomly choose $r \xleftarrow{\mathsf{r}} \mathbb{Z}_{\frac{N-1}{4}}$ and set $c_i \leftarrow g_i^r$ for every $i \in [\ell]$. The distribution of c is statistically close to the uniform distribution over $\mathcal{V}$. Moreover, r is a witness of $c \in \mathcal{V}$. We can sample a random element from $\mathcal{C}$ in a similar fashion.

For $(\mathcal{C}, \mathcal{V})$ defined above, we construct two projective hash functions $\mathcal{H} = (\mathcal{C}, \mathcal{V}, \mathcal{K}, \mathcal{SK}, \mathcal{PK}, \Lambda, \mu)$ and $\hat{\mathcal{H}} = \left(\mathcal{C}, \mathcal{V}, \hat{\mathcal{K}}, \hat{\mathcal{SK}}, \hat{\mathcal{PK}}, \hat{\Lambda}, \hat{\mu}\right)$. The construction of $\mathcal{H}$ is the same as that of projective hash function based on the QR assumption proposed by Wee [19] thus is a generalization of the KDM-CPA secure PKE scheme proposed by Brakerski and Goldwasser [5].

4.2.1 Construction of $\mathcal{H}$

We define $\mathcal{SK} = \{0,1\}^\ell$, $\mathcal{PK} = \mathbb{QR}_N$, and $\mathcal{K} = \mathbb{J}_N$. For every $\mathsf{sk} = s_1 \cdots s_\ell \in \{0,1\}^\ell$ and $c = (c_1, \ldots, c_\ell) \in \mathcal{C}$, we also define μ and Λ as

$$\mu(\mathsf{sk}) = \prod_{i \in [\ell]} g_i^{s_i} \quad \text{and} \quad \Lambda_{\mathsf{sk}}(c) = \prod_{i \in [\ell]} c_i^{s_i} \ .$$

Projective property. Let $\mathsf{sk} = s_1 \cdots s_\ell \in \{0,1\}^\ell$, $\mathsf{pk} = \prod_{i \in [\ell]} g_i^{s_i}$, and $c = (g_1^r, \ldots, g_\ell^r)$, where $r \in \mathbb{Z}_n$. We define the public evaluation algorithm Pub as

$$\mathsf{Pub}(\mathsf{pk}, c, r) = \mathsf{pk}^r \ .$$

We see that

$$\mathsf{pk}^r = \left(\prod_{i \in [\ell]} g_i^{s_i} \right)^r = \prod_{i \in [\ell]} (g_i^r)^{s_i} = \Lambda_{\mathsf{sk}}(c)$$

and thus $\mathcal{H}$ satisfies projective property.

Homomorphism. For every $\mathsf{sk} = s_1 \cdots s_\ell \in \{0,1\}^\ell$, $c = (c_1, \ldots, c_\ell) \in \mathcal{C}$, and $c' = (c'_1, \ldots, c'_\ell) \in \mathcal{C}$, we have

$$\Lambda_{\mathsf{sk}}(c) \cdot \Lambda_{\mathsf{sk}}(c') = \prod_{i \in [\ell]} c_i^{s_i} \cdot \prod_{i \in [\ell]} (c'_i)^{s_i} = \prod_{i \in [\ell]} (c_i \cdot c'_i)^{s_i} = \Lambda_{\mathsf{sk}}(c \cdot c')$$

and thus $\mathcal{H}$ is homomorphic.

Average-case smoothness. For an element $e = (-1)^d \cdot g^r \in \mathbb{J}_N$, we define $e \bmod \mathbb{QR}_N = (-1)^d$. Similarly to Wee [19], we can prove a weaker property that $\Lambda_{\mathsf{sk}}(c) \bmod \mathbb{QR}_N$ uniformly distributes over $\langle -1 \rangle$ using leftover hash lemma.

In our construction, to mask a secret key of Π_{cca} relying on this property, we need to use several instances of $\mathcal{H}$ in parallel. However, we have another option which is more efficient than the parallel construction. Below, we show it.

Without loss of generality, we assume that the secret-key space of Π_{cca} is $\mathbb{QR}_N$. When mask csk in the key generation, we generate $c^* = (c_1^*, \ldots, c_\ell^*) \xleftarrow{\mathsf{r}} \mathbb{QR}_N^\ell$ and computes $d^* \leftarrow \left(\prod_{i\in[\ell]} (c_i^*)^{s_i}\right) \cdot \mathsf{csk}$.

The leftover hash lemma implies that the following two distributions

$$\left(c^*, \prod_{i\in[\ell]} (c_i{}^*)^{s_i}, \mathsf{pk} = \prod_{i\in[\ell]} g_i^{s_i}\right) \text{ and } \left(c^*, \mathsf{K}, \mathsf{pk} = \prod_{i\in[\ell]} g_i^{s_i}\right)$$

are $\sqrt{\frac{n\cdot n}{4\cdot 2^\ell}}$-close, where $s_1 \cdots s_\ell \xleftarrow{\mathsf{r}} \{0,1\}^\ell$ and $\mathsf{K} \xleftarrow{\mathsf{r}} \mathbb{QR}_N$. Therefore, by setting $\ell = 3\log n$, we can statistically hide csk from the view of an adversary in the proof of KDM$^{(1)}$-CCA security.

4.2.2 Construction of $\hat{\mathcal{H}}$

We define $\hat{\mathcal{SK}} = \mathbb{Z}_{2n}^{\lambda\times\ell}$,[6] $\hat{\mathcal{PK}} = \mathbb{QR}_N^\lambda$, and $\hat{\mathcal{K}} = \mathbb{J}_N^\lambda$. For every

$$\hat{\mathsf{sk}} = \begin{pmatrix} x_{11}, \ldots, x_{1\ell} \\ \vdots \\ x_{\lambda 1}, \ldots, x_{\lambda\ell} \end{pmatrix} \in \hat{\mathcal{SK}}$$

and $c = (c_1, \ldots, c_\ell) \in \mathcal{C}$, we also define $\hat{\mu}$ and $\hat{\Lambda}$ as

$$\hat{\mu}\left(\hat{\mathsf{sk}}\right) = \begin{pmatrix} \prod_{i\in[\ell]} g_i^{x_{1i}} \\ \vdots \\ \prod_{i\in[\ell]} g_i^{x_{\lambda i}} \end{pmatrix} \quad \text{and} \quad \hat{\Lambda}_{\hat{\mathsf{sk}}}(c) = \begin{pmatrix} \prod_{i\in[\ell]} c_i^{x_{1i}} \\ \vdots \\ \prod_{i\in[\ell]} c_i^{x_{\lambda i}} \end{pmatrix} .$$

Projective property. Let

$$\hat{\mathsf{sk}} = \begin{pmatrix} x_{11}, \ldots, x_{1\ell} \\ \vdots \\ x_{\lambda 1}, \ldots, x_{\lambda\ell} \end{pmatrix} \in \hat{\mathcal{SK}} \ , \hat{\mathsf{pk}} = \begin{pmatrix} \hat{\mathsf{pk}}_1 \\ \vdots \\ \hat{\mathsf{pk}}_\lambda \end{pmatrix} = \begin{pmatrix} \prod_{i\in[\ell]} g_i^{x_{1i}} \\ \vdots \\ \prod_{i\in[\ell]} g_i^{x_{\lambda i}} \end{pmatrix} \in \hat{\mathcal{PK}}$$

and $c = (g_1^r, \ldots, g_\ell^r)$, where $r \in \mathbb{Z}_n$. We define the public evaluation algorithm $\hat{\mathsf{Pub}}$ as

$$\hat{\mathsf{Pub}}\left(\hat{\mathsf{pk}}, c, r\right) = \begin{pmatrix} \left(\hat{\mathsf{pk}}_1\right)^r \\ \vdots \\ \left(\hat{\mathsf{pk}}_\lambda\right)^r \end{pmatrix} .$$

Similarly to $\mathcal{H}$, we see that $\hat{\mathcal{H}}$ satisfies projective property.

[6] In the actual construction, we sample $\hat{\mathsf{sk}}$ from $\mathbb{Z}_{\frac{N-1}{2}}^{\lambda\times\ell}$ to sample $\hat{\mathsf{sk}}$ without knowing n. The uniform distributions over $\mathbb{Z}_{2n}^{\lambda\times\ell}$ and $\mathbb{Z}_{\frac{N-1}{2}}^{\lambda\times\ell}$ are statistically close.

Universal property. We need to prove that the universal property holds not only for all $c \in \mathcal{C} \setminus \mathcal{V}$ but also all $c \in \mathbb{J}_N^\ell \setminus \mathcal{C}$. This is because we cannot efficiently check the membership of $\mathcal{C}$. Note that we can check the membership of $\mathbb{J}_N$ by computing Jacobi symbol with respect to N, and Jacobi symbol with respect to N can be computed without factorizations of N, that is P and Q [18, Section 12.3].

For every $c = (c_1, \ldots, c_\ell) \in \mathbb{J}_N^\ell \setminus \mathcal{C}$, we define $\hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$ in the same way as above. For every $\hat{\mathsf{pk}} = \left(\hat{\mathsf{pk}}_1, \ldots, \hat{\mathsf{pk}}_\lambda\right) \in \hat{\mathcal{PK}}$, $c = (c_1, \ldots, c_\ell) \in \mathbb{J}_N^\ell$, and $(\pi_1, \ldots, \pi_\lambda) \in \hat{\mathcal{K}}$, we consider the following probability

$$\Pr_{x_{ji} \xleftarrow{r} \mathbb{Z}_{2n}} \left[\prod_{i \in [\ell]} c_i^{x_{ji}} = \pi_j \text{ for all } j \in [\lambda] \middle| \prod_{i \in [\ell]} g_i^{x_{ji}} = \hat{\mathsf{pk}}_j \text{ for all } j \in [\lambda] \right]$$
$$= \prod_{j \in [\lambda]} \Pr_{x_{ji} \xleftarrow{r} \mathbb{Z}_{2n}} \left[\prod_{i \in [\ell]} c_i^{x_{ji}} = \pi_j \middle| \prod_{i \in [\ell]} g_i^{x_{ji} \bmod n} = \hat{\mathsf{pk}}_j \right] . \tag{2}$$

We first consider the case where at least one element of $c = (c_1, \ldots, c_\ell)$ is not in $\mathbb{QR}_N$. Suppose that $c_{i^*} \in \mathbb{J}_N \setminus \mathbb{QR}_N$ for some $i^* \in [\ell]$.

For two elements $e_0, e_1 \in \mathbb{J}_N$, we write $e_0 \equiv e_1 \bmod \mathbb{QR}_N$ to denote that $e_0 \bmod \mathbb{QR}_N = e_1 \bmod \mathbb{QR}_N$. For two elements $e_0, e_1 \in \mathbb{J}_N$, if $e_0 = e_1$ holds, then so does $e_0 \equiv e_1 \bmod \mathbb{QR}_N$. Thus, the probability of Equation 2 is bounded by

$$\prod_{j \in [\lambda]} \Pr_{x_{ji} \xleftarrow{r} \mathbb{Z}_{2n}} \left[\prod_{i \in [\ell]} c_i^{x_{ji}} \equiv \pi_j \bmod \mathbb{QR}_N \middle| \prod_{i \in [\ell]} g_i^{x_{ji} \bmod n} = \hat{\mathsf{pk}}_j \right] .$$

For every $i \in [\ell]$ and $j \in [\lambda]$, $c_i^{x_{ji}} \bmod \mathbb{QR}_N$ is determined by only $x_{ji} \bmod 2$ and independent of $x_{ji} \bmod n$ from the Chinese Remainder Theorem since 2 and $n = pq$ are relatively prime. Therefore, the above probability is

$$\prod_{j \in [\lambda]} \Pr_{x_{ji} \xleftarrow{r} \mathbb{Z}_{2n}} \left[\prod_{i \in [\ell]} c_i^{x_{ji}} \equiv \pi_j \bmod \mathbb{QR}_N \right] .$$

Since $c_{i^*} \notin \mathbb{QR}_N$, we can write $c_{i^*} = -g^{r_{i^*}}$, where $r_{i^*} \in \mathbb{Z}_n$. For every $j \in [\lambda]$, we have

$$c_{i^*}^{x_{ji^*}} = (-1)^{x_{ji^*} \bmod 2} \cdot (g^{r_{i^*}})^{x_{ji^*} \bmod n} .$$

Then, the above probability is

$$\prod_{j \in [\lambda]} \Pr_{x_{ji} \xleftarrow{r} \mathbb{Z}_{2n}} \left[(-1)^{x_{ji^*} \bmod 2} = \pi_j \cdot \left(\prod_{i \in [\ell], i \neq i^*} c_i^{x_{ji}} \right)^{-1} \bmod \mathbb{QR}_N \right] = \frac{1}{2^\lambda} .$$

Thus, in this case, the probability of Equation 2 is negligible in λ.

We next consider the case where all elements of $c = (c_1, \ldots, c_\ell) \notin \mathcal{V}$ are in $\mathbb{QR}_N$. In this case, we can write $c = \left(g_1^{r_1}, \ldots, g_\ell^{r_\ell}\right)$, where $r_1, \ldots, r_\ell \in \mathbb{Z}_n$ and there exist $i_1, i_2 \in [\ell]$ such that $r_{i_1} \neq r_{i_2}$. Let g be a generator of $\mathbb{QR}_N$. Since g_i is a generator of $\mathbb{QR}_N$, there exists $\alpha_i \in \mathbb{Z}_n^*$ such that $g_i = g^{\alpha_i}$ for every $i \in [\ell]$. The probability of Equation 2 is 0 if $\pi_j \notin \mathbb{QR}_N$ for some $j \in [\lambda]$, and thus we consider the case where $\pi_j \in \mathbb{QR}_N$ for every $j \in [\lambda]$. Then, the probability of Equation 2 is the same as

$$\prod_{j \in [\lambda]} \Pr_{x_{ji} \xleftarrow{r} \mathbb{Z}_n} \left[\sum_{i \in [\ell]} \alpha_i r_i x_{ji} \equiv \log_g \pi_j \bmod n \middle| \sum_{i \in [\ell]} \alpha_i x_{ji} \equiv \log_g \hat{\mathsf{pk}}_j \bmod n \right] .$$

Since $r_{i_1} \not\equiv r_{i_2} \bmod n$, either $r_{i_1} \not\equiv r_{i_2} \bmod p$ or $r_{i_1} \not\equiv r_{i_2} \bmod q$ holds. Without loss of generality, we assume that $r_{i_1} \not\equiv r_{i_2} \bmod p$. Since p and q are primes, the above probability is bounded by

$$\prod_{j\in[\lambda]} \Pr_{x_{ji}\xleftarrow{\mathsf{r}}\mathbb{Z}_n} \left[\sum_{i\in[\ell]} \alpha_i r_i x_{ji} \equiv \log_g \pi_j \bmod p \middle| \sum_{i\in[\ell]} \alpha_i x_{ji} \equiv \log_g \hat{\mathsf{pk}}_j \bmod p \right] .$$

Since $r_{i_1} \not\equiv r_{i_2} \bmod p$, two equations

$$\sum_{i\in[\ell]} \alpha_i r_i x_{ji} \equiv \log_g \pi_j \bmod p \text{ and } \sum_{i\in[\ell]} \alpha_i x_{ji} \equiv \log_g \hat{\mathsf{pk}}_j \bmod p$$

are linearly independent. Thus, the above probability is $\frac{1}{p^\lambda}$.

From the above, for every $c \in \mathbb{J}_N^\ell \setminus \mathcal{V}$, the probability of Equation 2 is negligible in λ.

4.2.3 Associated Function Class

The message space of the QR based construction is $\{0,1\}$. We define $\phi(m \in \{0,1\}) = (-1)^m$. Let $\mathcal{F}_{\mathsf{qr}}$ be a family of functions described as

$$\begin{aligned} f_e\left(\mathsf{sk}, \hat{\mathsf{sk}}, c^*, d^*\right) &= \phi^{-1}\left(\Lambda_{\mathsf{sk}}\left((-1)^{e_1}, \ldots, (-1)^{e_\ell}\right) + \phi\left(f\left(\hat{\mathsf{sk}}, c^*, d^*\right)\right)\right) \\ &= \phi^{-1}\left((-1)^{\sum_{i\in[\ell]} e_i s_i + f\left(\hat{\mathsf{sk}}, c^*, d^*\right)}\right) \\ &= \left(\sum_{i\in[\ell]} e_i s_i + f\left(\hat{\mathsf{sk}}, c^*, d^*\right)\right) \bmod 2 \ , \end{aligned}$$

where $e_i \in \mathbb{Z}_2$ for every $i \in [\ell]$ and f is a function of single-bit output. The QR based construction is $\mathcal{F}_{\mathsf{qr}}$-KDM$^{(1)}$-CCA secure. In the construction, we can maintain $\hat{\mathsf{sk}}$, c^*, and d^* as bit strings. In this case, the above function class includes affine functions and projection functions.

4.3 Instantiation Based on the DCR Assumption

Definition 10 (DCR assumption) *Let $N = PQ$ be a Blum integer for λ-bit safe primes $P, Q \equiv 3 \bmod 4$ such that $P = 2p+1$ and $Q = 2q+1$ for primes p and q. Let $n = pq$. We can decompose $\mathbb{Z}_{N^2}^*$ as an internal direct product $G_N \otimes \langle -1\rangle \otimes G_n \otimes G_2$, where $\langle -1\rangle$ is the subgroup of $\mathbb{Z}_{N^2}^*$ generated by $-1 \bmod N^2$, and G_N, G_n, and G_2 are cyclic groups of order N, n, and 2, respectively. Let $T = 1 + N \in \mathbb{Z}_{N^2}^*$. T has order N, and thus it generates G_N.*

We say that the DCR assumption holds if for any PPT algorithm $\mathcal{A}$, we have

$$|\Pr[\mathcal{A}(N, y) = 1] - \Pr[\mathcal{A}(N, y') = 1]| = \mathsf{negl}(\lambda) \ ,$$

where $y \xleftarrow{\mathsf{r}} G_N \otimes \langle -1\rangle \otimes G_n$ and $y' \xleftarrow{\mathsf{r}} \langle -1\rangle \otimes G_n$.

We define N, G_N, G_n, $\langle -1\rangle$, and T as in Definition 10. Let $g_1, \ldots, g_\ell$ be random generators of G_n, where ℓ is determined later. We can generate a random generator g of G_n by generating $\mu \xleftarrow{\mathsf{r}} \mathbb{Z}_{N^2}^*$ and setting $g = \mu^{2N} \bmod N^2$. Then, g is a generator of G_n with high probability.

We define $\mathcal{C}$ and $\mathcal{V}$ as

$$\begin{aligned} \mathcal{C} = \Big\{ &\left(T^{d_1} \cdot (-1)^{\gamma_1} \cdot g_1^r, \ldots, T^{d_\ell} \cdot (-1)^{\gamma_\ell} \cdot g_\ell^r\right) \\ &\qquad | d_1, \ldots, d_\ell \in \mathbb{Z}_N, \gamma_1, \ldots, \gamma_\ell \in \mathbb{Z}_2, r \in \mathbb{Z}_n \Big\} \text{, and} \\ \mathcal{V} = \{&((-1)^{\gamma_1} \cdot g_1^r, \ldots, (-1)^{\gamma_\ell} \cdot g_\ell^r) \,|\gamma_1, \ldots, \gamma_\ell \in \mathbb{Z}_2, r \in \mathbb{Z}_n\} \ . \end{aligned}$$

$\mathcal{V}$ is a subgroup of $\mathcal{C}$ and subset membership problem of $(\mathcal{C}, \mathcal{V})$ is hard under the DCR assumption. As shown by previous works [5, 19], two distributions $\left\{T^{d_i} \cdot (-g_i)^r\right\}_{i\in[\ell]}$ and $\{(-g_i)^r\}_{i\in[\ell]}$ are computationally indistinguishable under the DCR assumption, where $d_i \xleftarrow{\mathsf{r}} \mathbb{Z}_N$ for every $i \in [\ell]$ and $r \xleftarrow{\mathsf{r}} \mathbb{Z}_{2n}$. We see that uniform distributions over $\mathcal{C}$ and $\mathcal{V}$ are also computationally indistinguishable under the DCR assumption.

Let g be another generator of G_n. Then, there exists $\alpha_i \in \mathbb{Z}_n^*$ such that $g^{\alpha_i} = g_i$ for every $i \in [\ell]$. The trapdoor for checking the membership of $\mathcal{V}$ is P, Q, and $\{\alpha_i\}_{i\in[\ell]}$.

When sampling a random element $c = (c_1, \ldots, c_\ell)$ from $\mathcal{V}$, we randomly choose $r \xleftarrow{\mathsf{r}} \mathbb{Z}_{\frac{N-1}{4}}$ and $\gamma_i \xleftarrow{\mathsf{r}} \mathbb{Z}_2$ for every $i \in [\ell]$, and set $c_i \leftarrow (-1)^{\gamma_i} \cdot g_i^r$ for every $i \in [\ell]$. The distribution of c is statistically close to the uniform distribution over $\mathcal{V}$. Moreover, r is a witness of $c \in \mathcal{V}$. We can sample a random element from $\mathcal{C}$ in a similar fashion.

For $(\mathcal{C}, \mathcal{V})$ defined above, we construct two projective hash functions $\mathcal{H} = (\mathcal{C}, \mathcal{V}, \mathcal{K}, \mathcal{SK}, \mathcal{PK}, \Lambda, \mu)$ and $\hat{\mathcal{H}} = \left(\mathcal{C}, \mathcal{V}, \hat{\mathcal{K}}, \hat{\mathcal{SK}}, \hat{\mathcal{PK}}, \hat{\Lambda}, \hat{\mu}\right)$. The construction of $\mathcal{H}$ is a slightly modified version of projective hash function based on the DCR assumption proposed by Wee [19] thus is a generalization of the KDM-CPA secure PKE scheme proposed by Brakerski and Goldwasser [5]. For the reason we need a modification, see Remark 4 after the constructions.

4.3.1 Construction of $\mathcal{H}$

We define $\mathbb{QR}_{N^s} = G_{N^{s-1}} \otimes G_n$ and $\mathbb{J}_{N^s} = G_{N^{s-1}} \otimes \langle -1 \rangle \otimes G_n = \langle -1 \rangle \otimes \mathbb{QR}_{N^s}$. We define $\mathcal{SK} = \{0,1\}^\ell$, $\mathcal{PK} = G_n$, and $\mathcal{K} = \mathbb{QR}_{N^s}$. For every $\mathsf{sk} = s_1 \cdots s_\ell \in \{0,1\}^\ell$ and $c = (c_1, \ldots, c_\ell) \in \mathcal{C}$, we also define μ and Λ as

$$\mu(\mathsf{sk}) = \prod_{i\in[\ell]} g_i^{2s_i} \quad \text{and} \quad \Lambda_{\mathsf{sk}}(c) = \prod_{i\in[\ell]} c_i^{2s_i} \quad .$$

Projective property. Let $\mathsf{sk} = s_1 \cdots s_\ell \in \{0,1\}^\ell$, $\mathsf{pk} = \prod_{i\in[\ell]} g_i^{2s_i}$, and $c = ((-1)^{\gamma_1} \cdot g_1^r, \ldots, (-1)^{\gamma_\ell} \cdot g_\ell^r)$, where $r \in \mathbb{Z}_n$ and $\gamma_i \in \mathbb{Z}_2$ for every $i \in [\ell]$. We define the public evaluation algorithm Pub as

$$\mathsf{Pub}(\mathsf{pk}, c, r) = \mathsf{pk}^r \quad .$$

We see that

$$\mathsf{pk}^r = \left(\prod_{i\in[\ell]} g_i^{2s_i}\right)^r = \prod_{i\in[\ell]} (g_i^r)^{2s_i} = \prod_{i\in[\ell]} ((-1)^{\gamma_i} \cdot g_i^r)^{2s_i} = \Lambda_{\mathsf{sk}}(c)$$

and thus $\mathcal{H}$ satisfies projective property.

Homomorphism. For every $\mathsf{sk} = s_1 \cdots s_\ell \in \{0,1\}^\ell$, $c = (c_1, \ldots, c_\ell) \in \mathcal{C}$, and $c' = (c'_1, \ldots, c'_\ell) \in \mathcal{C}$, we have

$$\Lambda_{\mathsf{sk}}(c) \cdot \Lambda_{\mathsf{sk}}(c') = \prod_{i\in[\ell]} c_i^{2s_i} \cdot \prod_{i\in[\ell]} (c'_i)^{2s_i} = \prod_{i\in[\ell]} (c_i \cdot c'_i)^{2s_i} = \Lambda_{\mathsf{sk}}(c \cdot c')$$

and thus $\mathcal{H}$ is homomorphic.

Average-case smoothness. Similarly to Wee [19], we prove a weaker property that is sufficient for our construction.

For an element $e = T^d \cdot g^r \in \mathbb{QR}_{N^s}$, we define $e \bmod G_n = T^d$. Let $c = (c_1, \ldots, c_\ell) = \left(T^{d_1} \cdot (-1)^{\gamma_1} \cdot g_1^r, \ldots, T^{d_\ell} \cdot (-1)^{\gamma_\ell} \cdot g_\ell^r\right)$, where $d_1, \ldots, d_\ell \in \mathbb{Z}_N$, $\gamma_1, \ldots, \gamma_\ell \in \mathbb{Z}_2$, and $r \in \mathbb{Z}_n$. We have

$$\Lambda_{\mathsf{sk}}(c) \bmod G_n = \prod_{i \in [\ell]} \left(T^{d_i} \cdot g_i^r\right)^{2s_i} \bmod G_n = T^{2\sum_{i \in [\ell]} d_i s_i \bmod N} \ .$$

The leftover hash lemma implies that the following two distributions

$$\left(c, \sum_{i \in [\ell]} d_i s_i \bmod N, \prod_{i \in [\ell]} g_i^{2s_i}\right) \text{ and } \left(c, \mathsf{K}, \prod_{i \in [\ell]} g_i^{2s_i}\right)$$

are $\sqrt{\frac{N \cdot n}{4 \cdot 2^\ell}}$-close, where $\mathsf{sk} = s_1 \cdots s_\ell \xleftarrow{\mathsf{r}} \{0,1\}^\ell$, $c = (c_1, \ldots, c_\ell) \xleftarrow{\mathsf{r}} \mathcal{C}$, and $\mathsf{K} \xleftarrow{\mathsf{r}} \mathbb{Z}_N$. Moreover, if K is uniformly at random over $\mathbb{Z}_N$, then so does $2\mathsf{K} \bmod N$. Therefore, by setting $\ell = 3 \log N$, the distribution of $\Lambda_{\mathsf{sk}}(c) \bmod G_n$ is statistically close to uniform over G_N.

4.3.2 Construction of $\hat{\mathcal{H}}$

We define $\hat{\mathcal{SK}} = \mathbb{Z}_{Nn}^\ell$,[7] $\hat{\mathcal{PK}} = G_n$, and $\hat{\mathcal{K}} = \mathbb{QR}_{N^s}$. For every $\hat{\mathsf{sk}} = (x_1, \ldots, x_\ell) \in \hat{\mathcal{SK}}$ and $c = (c_1, \ldots, c_\ell) \in \mathcal{C}$, we also define $\hat{\mu}$ and $\hat{\Lambda}$ as

$$\hat{\mu}\left(\hat{\mathsf{sk}}\right) = \prod_{i \in [\ell]} g_i^{2x_i} \ \text{ and } \ \hat{\Lambda}_{\hat{\mathsf{sk}}}(c) = \prod_{i \in [\ell]} c_i^{2x_i} \ .$$

Projective property. For every $\hat{\mathsf{sk}} = (x_1, \ldots, x_\ell) \in \hat{\mathcal{SK}}$, $\hat{\mathsf{pk}} = \prod_{i \in [\ell]} g_i^{2x_i}$, and $c = ((-1)^{\gamma_1} \cdot g_1^r, \ldots, (-1)^{\gamma_\ell} \cdot g_\ell^r)$, where $r \in \mathbb{Z}_n$ and $\gamma_i \in \mathbb{Z}_2$ for every $i \in [\ell]$, we define the public evaluation algorithm $\hat{\mathsf{Pub}}$ as

$$\hat{\mathsf{Pub}}\left(\hat{\mathsf{pk}}, c, r\right) = \hat{\mathsf{pk}}^r \ .$$

Similarly to $\mathcal{H}$, we see that $\hat{\mathcal{H}}$ satisfies projective property.

Universal property. We need to prove that the universal property holds not only for all $c \in \mathcal{C} \setminus \mathcal{V}$ but also all $c \in \mathbb{J}_{N^s}^\ell \setminus \mathcal{C}$. This is because we cannot efficiently check the membership of $\mathcal{C}$. Note that we can check the membership of $\mathbb{J}_{N^s}$ by computing Jacobi symbol with respect to N, and Jacobi symbol with respect to N can be computed without factorizations of N, that is P and Q [18, Section 12.3].

For every $c \in \mathbb{J}_{N^s}^\ell \setminus \mathcal{C}$, we define $\hat{\Lambda}_{\hat{\mathsf{sk}}}(c)$ in the same way as above. For every $\hat{\mathsf{pk}} \in \hat{\mathcal{PK}}$, $c = (c_1, \ldots, c_\ell) \in \mathbb{J}_{N^s}^\ell$, and $\pi \in \hat{\mathcal{K}}$, we consider the following probability

$$\begin{aligned} &\Pr_{x_i \xleftarrow{\mathsf{r}} \mathbb{Z}_{Nn}} \left[\prod_{i \in [\ell]} c_i^{2x_i} = \pi \middle| \prod_{i \in [\ell]} g_i^{2x_i} = \hat{\mathsf{pk}}\right] \\ &\quad = \Pr_{x_i \xleftarrow{\mathsf{r}} \mathbb{Z}_{Nn}} \left[\prod_{i \in [\ell]} c_i^{2x_i} = \pi \middle| \prod_{i \in [\ell]} g_i^{2(x_i \bmod n)} = \hat{\mathsf{pk}}\right] \ . \end{aligned} \tag{3}$$

[7] In the actual construction, we sample $\hat{\mathsf{sk}}$ from $\mathbb{Z}_{\frac{\lambda \times \ell}{N(N-1)}}$ to sample $\hat{\mathsf{sk}}$ without knowing n. The uniform distributions over $\mathbb{Z}_{Nn}^{\lambda \times \ell}$ and $\mathbb{Z}_{\frac{\lambda \times \ell}{N(N-1)}}$ are statistically close.

We first consider the case where at least one element of $c = (c_1, \ldots, c_\ell)$ is not in $\langle -1 \rangle \otimes G_n$. Suppose that $c_{i^*} \in \mathbb{J}_{N^s} \setminus \langle -1 \rangle \otimes G_n$ for some $i^* \in [\ell]$.

For two elements $e_0, e_1 \in \mathbb{QR}_{N^s}$, we write $e_0 \equiv e_1 \bmod G_n$ to denote that $e_0 \bmod G_n = e_1 \bmod G_n$. For two elements $e_0, e_1 \in \mathbb{QR}_{N^s}$, if $e_0 = e_1$ holds, then so does $e_0 \equiv e_1 \bmod G_n$. Thus, the probability of Equation 3 is bounded by

$$\Pr_{x_i \xleftarrow{r} \mathbb{Z}_{Nn}} \left[\prod_{i \in [\ell]} c_i^{2x_i} \equiv \pi \bmod G_n \middle| \prod_{i \in [\ell]} g_i^{2(x_i \bmod n)} = \hat{\mathsf{pk}} \right] .$$

For every $i \in [\ell]$, $c_i^{2x_i} \bmod G_n$ is determined by only $x_i \bmod N$ and independent of $x_i \bmod n$ from the Chinese Remainder Theorem since $N = PQ$ and $n = pq$ are relatively prime. Therefore, the above probability is

$$\Pr_{x_i \xleftarrow{r} \mathbb{Z}_{Nn}} \left[\prod_{i \in [\ell]} c_i^{2x_i} \equiv \pi \bmod G_n \right] .$$

Since $c_{i^*} \notin \langle -1 \rangle \otimes G_n$, we can write $c_{i^*} = T^{d_{i^*}} \cdot (-1)^{\gamma_{i^*}} \cdot g^{r_{i^*}}$, where $d_{i^*} \in \mathbb{Z}_N$ such that $d_{i^*} \neq 0$, $\gamma_{i^*} \in \mathbb{Z}_2$, and $r_{i^*} \in \mathbb{Z}_n$. We have

$$c_{i^*}^{2x_{i^*}} = T^{2d_{i^*}(x_{i^*} \bmod N)} \cdot g^{2r_{i^*}(x_{i^*} \bmod n)} .$$

Then, the above probability is the same as

$$\Pr_{x_i \xleftarrow{r} \mathbb{Z}_{Nn}} \left[T^{2d_{i^*}(x_{i^*} \bmod N)} \equiv \pi \cdot \left(\prod_{i \in [\ell], i \neq i^*} c_i^{2x_i} \right)^{-1} \bmod G_n \right] .$$

This probability is smaller than $\frac{1}{P}$ or $\frac{1}{Q}$. Thus, in this case, the probability of Equation 3 is negligible in λ.

We next consider the case where all elements of $c = (c_1, \ldots, c_\ell) \notin \mathcal{V}$ are in $\langle -1 \rangle \otimes G_n$. In this case, we can write $c_i = (-1)^{\gamma_i} \cdot g_i^{r_i}$, where $\gamma_i \in \mathbb{Z}_2$ and $r_i \in \mathbb{Z}_n$ for every $i \in [\ell]$. Since $c \notin \mathcal{V}$, there exist $i_1, i_2 \in [\ell]$ such that $r_{i_1} \neq r_{i_2}$. Let g be a generator of G_n. Since g_i is a generator of G_n, there exists $\alpha_i \in \mathbb{Z}_n^*$ such that $g_i = g^{\alpha_i}$ for every $i \in [\ell]$. The probability of Equation 3 is 0 if $\pi \notin G_n$, and thus we consider cases of $\pi \in G_n$. Then, the probability of Equation 3 is the same as

$$\Pr_{x_i \xleftarrow{r} \mathbb{Z}_n} \left[2 \sum_{i \in [\ell]} \alpha_i r_i x_i \equiv \log_g \pi \bmod n \middle| 2 \sum_{i \in [\ell]} \alpha_i x_i \equiv \log_g \hat{\mathsf{pk}} \bmod n \right] .$$

Since $r_1 \not\equiv r_2 \bmod n$, either $r_{i_1} \not\equiv r_{i_2} \bmod p$ or $r_{i_1} \not\equiv r_{i_2} \bmod q$ holds. Without loss of generality, we assume that $r_{i_1} \not\equiv r_{i_2} \bmod p$. Since p and q are primes, the above probability is bounded by

$$\Pr_{x_i \xleftarrow{r} \mathbb{Z}_n} \left[\sum_{i \in [\ell]} \alpha_i r_i x_i \equiv 2^{-1} \cdot \log_g \pi \bmod p \middle| \sum_{i \in [\ell]} \alpha_i x_i \equiv 2^{-1} \cdot \log_g \hat{\mathsf{pk}} \bmod p \right] .$$

Since $r_{i_1} \not\equiv r_{i_2} \bmod p$, two equations

$$\sum_{i \in [\ell]} \alpha_i r_i x_i \equiv 2^{-1} \cdot \log_g \pi \bmod p \text{ , and } \sum_{i \in [\ell]} \alpha_i x_i \equiv 2^{-1} \cdot \log_g \hat{\mathsf{pk}} \bmod p$$

are linearly independent. Therefore, the above probability is $\frac{1}{p}$.

Thus, for every $c \in \mathbb{J}_{N^s}^\ell \setminus \mathcal{V}$, the probability of Equation 3 is negligible in λ.

Remark 4 (Difference with previous works [5, 19]) The difference between our construction and previous works is that when we compute a hash value of c, we first square each element of c. By this operation, the ranges of $\mathcal{H}$ and $\hat{\mathcal{H}}$ are $\mathbb{QR}_{N^s} = G_N \cdot G_n$.

If we do not perform squaring, the ranges will be $\mathbb{J}_{N^s} = G_N \cdot \langle -1 \rangle \cdot G_n$ and hash values of some elements can be predicted with high probability since the order of $\langle -1 \rangle$ is 2. In fact, we can correctly guess the hash value of $(-1, \ldots, -1) \in \langle -1 \rangle^\ell$ with probability at least $\frac{1}{2}$. In this case, to achieve universal property of $\hat{\mathcal{H}}$, we need parallelization similarly to the QR based construction.

One might think we have another option where $\mathcal{C}$ and $\mathcal{V}$ are defined as subgroups of $\mathbb{QR}_{N^s}$ and G_n, respectively. This option is not working. The reason is that we cannot efficiently check the membership of $\mathbb{QR}_{N^s}$. Therefore, if we use such $\mathcal{C}$ and $\mathcal{V}$, we still need to take elements of $\mathbb{J}_{N^s}$ into account, and thus we need squaring.

4.3.3 Associated Function Class

The message space of the DCR based construction is $\mathbb{Z}_N$. We define $\phi(m \in \mathbb{Z}_N) = T^m$. Let $\mathcal{F}_{\mathsf{dcr}}$ be a family of functions described as

$$\begin{aligned} f_e\left(\mathsf{sk}, \hat{\mathsf{sk}}, c^*, d^*\right) &= \phi^{-1}\left(\Lambda_{\mathsf{sk}}\left(T^{e_1/2}, \ldots, T^{e_\ell/2}\right) + \phi\left(f\left(\hat{\mathsf{sk}}, c^*, d^*\right)\right)\right) \\ &= \phi^{-1}\left(T^{\sum_{i\in[\ell]} e_i s_i + f\left(\hat{\mathsf{sk}}, c^*, d^*\right)}\right) \\ &= \left(\sum_{i\in[\ell]} e_i s_i + f\left(\hat{\mathsf{sk}}, c^*, d^*\right)\right) \bmod N \ , \end{aligned}$$

where $\frac{1}{2}$ denotes the inverse of 2 modulo N, $e_i \in \mathbb{Z}_N$ for every $i \in [\ell]$, and f is a function whose range is $\mathbb{Z}_N$. The DCR based construction is $\mathcal{F}_{\mathsf{dcr}}$-KDM$^{(1)}$-CCA secure. In the construction, we can maintain $\hat{\mathsf{sk}}$, c^*, and d^* as bit strings. In this case, the above function class includes affine functions and projection functions.

5 KDM-CCA Security of the DDH Based Scheme

Although our framework shown in Section 3 captures only KDM$^{(1)}$-CCA security, we can prove the KDM-CCA security of concrete instantiations. In this section, we prove that our concrete instantiation based on the DDH assumption is KDM-CCA secure. We also briefly explain how to prove the multi user security of our QR and DCR based schemes in Remark 5 at the end of this section.

Let $\mathbb{G}$ be a cyclic group of prime order p and g a random generator of $\mathbb{G}$. Let $\Pi_{\mathsf{cca}} = (\mathsf{KG}_{\mathsf{cca}}, \mathsf{Enc}_{\mathsf{cca}}, \mathsf{Dec}_{\mathsf{cca}})$ be a PKE scheme. We assume that the secret-key space of Π_{cca} is $\mathbb{G}$ for simplicity. We construct the following PKE scheme $\Pi_{\mathsf{ddh}} = (\mathsf{KG}_{\mathsf{ddh}}, \mathsf{Enc}_{\mathsf{ddh}}, \mathsf{Dec}_{\mathsf{ddh}})$. The message space of Π_{ddh} is $\{0,1\}$.

$\mathsf{KG}_{\mathsf{ddh}}(1^\lambda)$:

- Generate $g_1, \ldots, g_\ell \xleftarrow{\mathsf{r}} \mathbb{G}$.
- Generate $s = s_1 \cdots s_\ell \xleftarrow{\mathsf{r}} \{0,1\}^\ell$ and $x_1, \ldots, x_\ell \xleftarrow{\mathsf{r}} \mathbb{Z}_p$.
- Compute $g_0 \leftarrow \prod_{i\in[\ell]} g_i^{s_i}$ and $\hat{g}_0 \leftarrow \prod_{i\in[\ell]} g_i^{x_i}$.
- Generate $(\mathsf{cpk}, \mathsf{csk}) \leftarrow \mathsf{KG}_{\mathsf{cca}}(1^\lambda)$.

- Generate $w_i \xleftarrow{r} \mathbb{Z}_p$ and set $e_i \leftarrow g_i^{w_i}$ for every $i \in [\ell]$.
- Compute $e_0 \leftarrow \prod_{i\in[\ell]} e_i^{s_i}$ and $u \leftarrow e_0 \cdot \mathsf{csk}$.
- Set $v := \{x_i\}_{i\in[\ell]} \| \{e_i\}_{i\in[\ell]} \| u$.
- Return $\mathsf{PK} := \left(\{g_i\}_{i\in[\ell]}, g_0, \hat{g}_0, \mathsf{cpk}\right)$ and $\mathsf{SK} := (s, v)$.

$\mathsf{Enc}_{\mathsf{ddh}}(\mathsf{PK}, m)$:

- Parse $\left(\{g_i\}_{i\in[\ell]}, g_0, \hat{g}_0, \mathsf{cpk}\right) \leftarrow \mathsf{PK}$.
- Generate $r \xleftarrow{r} \mathbb{Z}_p$ and compute $c_i \leftarrow g_i^r$ for every $i \in [\ell]$.
- Compute $d \leftarrow g^m \cdot g_0^r$ and $\pi \leftarrow \hat{g}_0^r$.
- Return $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cca}}\left(\mathsf{cpk}, \left(\{c_i\}_{i\in[\ell]}, d, \pi\right)\right)$.

$\mathsf{Dec}_{\mathsf{ddh}}(\mathsf{SK}, \mathsf{CT})$:

- Parse $\left(s, \{x_i\}_{i\in[\ell]} \| \{e_i\}_{i\in[\ell]} \| u\right) \leftarrow \mathsf{SK}$.
- Compute $\mathsf{csk} \leftarrow u \cdot \left(\prod_{i\in[\ell]} e_i^{s_i}\right)^{-1}$.
- Compute $\left(\{c_i\}_{i\in[\ell]}, d, \pi\right) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{csk}, \mathsf{CT})$. If the decryption result is not in $\mathbb{G}^{\ell+2}$, returns $\perp$. Otherwise, compute as follows.
- Return $\perp$ if $\pi \neq \prod_{i\in[\ell]} c_i^{x_i}$ and $m \leftarrow \log_g \left(d \cdot \left(\prod_{i\in[\ell]} c_i^{s_i}\right)^{-1}\right)$ otherwise.

Correctness. In the decryption algorithm, we need to compute discrete logarithm on $\mathbb{G}$. We can efficiently perform this operation since we restrict the message space to $\{0,1\}$. The decryption algorithm returns $\perp$ if $d \cdot \left(\prod_{i\in[\ell]} c_i^{s_i}\right)^{-1} \notin \{1, g\}$. Then, the correctness of Π_{ddh} follows from that of Π_{cca}.

Let n be the number of key pairs in the security game. We define $\mathcal{F}_{\mathsf{ddh}}$ as a function family consisting of functions described as

$$f'\left(\{s_k, v_k\}_{k\in[n]}\right) = \sum_{k\in[n]} \langle a_k, s_k\rangle + f\left(\{v_k\}_{k\in[n]}\right) \quad ,$$

where $\langle\cdot,\cdot\rangle$ denotes inner product over $\mathbb{Z}$, $a_k \in \{0,1\}^\ell$, and f is a function such that $\sum_{k\in[n]} \langle a_k, s_k\rangle + f\left(\{v_k\}_{k\in[n]}\right) \in \{0,1\}$ for every $\{s_k\}_{k\in[n]}$ and $\{v_k\}_{k\in[n]}$. By maintaining $\{v_k\}_{k\in[n]}$ as bit strings, $\mathcal{F}_{\mathsf{ddh}}$ includes projection functions of single-bit output. Π_{kdm} is KDM-CCA secure with respect to $\mathcal{F}_{\mathsf{ddh}}$. Formally, we prove the following theorem.

Theorem 4 *Let Π_{cca} be IND-CCA secure. Assuming the DDH problem is hard on $\mathbb{G}$, Π_{ddh} is $\mathcal{F}_{\mathsf{ddh}}$-KDM-CCA secure.*

As noted in Section 4.1.3, we can construct a PKE scheme that is KDM-CCA secure with respect to affine functions defined by Boneh et al. [4] by some simple modifications.

Proof of Theorem 4. Let n be the number of keys. Let $\mathcal{A}$ be an adversary that attacks the $\mathcal{F}_{\mathsf{ddh}}$-KDM-CCA security of Π_{ddh}. We proceed the proof via a sequence of games. For every $t \in \{0, \ldots, 11\}$, let $\mathtt{SUC}_t$ be the event that $\mathcal{A}$ succeeds in guessing the challenge bit b in Game t.

Game 0: This is the original $\mathcal{F}_{\mathsf{ddh}}$-KDM$^{(n)}$-CCA game regarding Π_{ddh}. We have $\mathsf{Adv}^{\mathsf{kdmcca}}_{\Pi_{\mathsf{ddh}},\mathcal{F}_{\mathsf{ddh}},\mathcal{A},n}(\lambda) = \left|\Pr[\mathtt{SUC}_0] - \frac{1}{2}\right|$. The detailed description is as follows.

1. The challenger chooses $b \xleftarrow{\mathsf{r}} \{0,1\}$ and generates $(\mathsf{PK}_k, \mathsf{SK}_k)$ for every $k \in [n]$ as follows.
   (a) Generate $g_{k1}, \ldots, g_{k\ell} \xleftarrow{\mathsf{r}} \mathbb{G}$.
   (b) Generate $s_k = s_{k1} \cdots s_{k\ell} \xleftarrow{\mathsf{r}} \{0,1\}^\ell$ and $x_{k1}, \ldots, x_{k\ell} \xleftarrow{\mathsf{r}} \mathbb{Z}_p$.
   (c) Compute $g_{k0} \leftarrow \prod_{i\in[\ell]} (g_{ki})^{s_{ki}}$ and $\hat{g}_{k0} \leftarrow \prod_{i\in[\ell]} (g_{ki})^{x_{ki}}$.
   (d) Generate $(\mathsf{cpk}_k, \mathsf{csk}_k) \leftarrow \mathsf{KG}_{\mathsf{cca}}(1^\lambda)$.
   (e) Generate $w_{ki} \xleftarrow{\mathsf{r}} \mathbb{Z}_p$ and set $e_{ki} \leftarrow (g_{ki})^{w_{ki}}$ for every $i \in [\ell]$.
   (f) Compute $e_{k0} \leftarrow \prod_{i\in[\ell]} (e_{ki})^{s_{ki}}$ and $u_k \leftarrow e_{k0} \cdot \mathsf{csk}_k$.
   (g) Set $v_k := \{x_{ki}\}_{i\in[\ell]} \| \{e_{ki}\}_{i\in[\ell]} \| u_k$.
   (h) Set $\mathsf{PK}_k := \left(\mathsf{cpk}_k, \{g_{ki}\}_{i\in[\ell]}, g_{k0}, \hat{g}_{k0}\right)$ and $\mathsf{SK}_k := (s_k, v_k)$.

   The challenger sends $\{\mathsf{PK}_k\}_{k\in[n]}$ to $\mathcal{A}$ and prepares a list L_{kdm}.
2. The challenger responds to queries made by $\mathcal{A}$.

   For a KDM query $\left(j, \left(\{a^0_k\}_{k\in[n]}, f^0\right), \left(\{a^1_k\}_{k\in[n]}, f^1\right)\right)$ made by $\mathcal{A}$, the challenger responds as follows.
   (a) Set $m := \sum_{k\in[n]} \langle a^b_k, s_k \rangle + f^b\left(\{v_k\}_{k\in[n]}\right)$.
   (b) Generate $r \xleftarrow{\mathsf{r}} \mathbb{Z}_p$ and compute $c_i \leftarrow (g_{ji})^r$ for every $i \in [\ell]$.
   (c) Compute $d \leftarrow g^m \cdot (g_{j0})^r$ and $\pi \leftarrow (\hat{g}_{j0})^r$.
   (d) Return $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cca}}\left(\mathsf{cpk}_j, \left(\{c_i\}_{i\in[\ell]}, d, \pi\right)\right)$ and add (j, CT) to L_{kdm}.

   For a decryption query (j, CT) made by $\mathcal{A}$, the challenger returns $\perp$ to $\mathcal{A}$ if $(j, \mathsf{CT}) \in L_{\mathsf{kdm}}$, and otherwise responds as follows.
   (a) Compute $\left(\{c_i\}_{i\in[\ell]}, d, \pi\right) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{csk}_j, \mathsf{CT})$. If the decryption result is not in $\mathbb{G}^{\ell+2}$, return $\perp$ and otherwise respond as follows.
   (b) Return $\perp$ if $\pi \neq \prod_{i\in[\ell]} c_i^{x_{ji}}$ and $m \leftarrow \log_g \left(d \cdot \left(\prod_{i\in[\ell]} c_i^{s_{ji}}\right)^{-1}\right)$ otherwise.
3. $\mathcal{A}$ outputs $b' \in \{0,1\}$.

Game 1: Same as Game 0 except how the challenger computes $\{e_{ki}\}_{i\in[\ell],k\in[n]}$. The challenger generates $w_i \xleftarrow{\mathsf{r}} \mathbb{Z}_p$ for every $i \in [\ell]$ and computes $e_{ki} \leftarrow (g_{ki})^{w_i}$ for every $i \in [\ell]$ and $k \in [n]$.

We have $|\Pr[\mathtt{SUC}_0] - \Pr[\mathtt{SUC}_1]| = \mathsf{negl}(\lambda)$ since $((g_{1i})^{w_{1i}}, \ldots, (g_{ni})^{w_{ni}})$ and $((g_{1i})^{w_i}, \ldots, (g_{ni})^{w_i})$ are computationally indistinguishable by the DDH assumption for every $i \in [\ell]$.

Game 2: Same as Game 1 except how the challenger generates $\{s_k\}_{k\in[n]}$ and $\{g_{ki}\}_{i\in[\ell],k\in[n]}$. The challenger first generates $s = s_1 \cdots s_\ell \xleftarrow{\mathsf{r}} \{0,1\}^\ell$ and $g_1, \ldots g_\ell \xleftarrow{\mathsf{r}} \mathbb{G}$. Then, for every $k \in [n]$, the challenger generates $\Delta_k \xleftarrow{\mathsf{r}} \{0,1\}^\ell$ and computes $s_k \leftarrow s \oplus \Delta_k$. In addition, for every $i \in [\ell]$ and $k \in [n]$, the challenger generates $\gamma_{ki} \xleftarrow{\mathsf{r}} \mathbb{Z}_p$ and computes $g_{ki} \leftarrow g_i^{\gamma_{ki}}$.

$|\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]| = 0$ holds since the difference between Game 1 and 2 is only conceptual.

From Game 3 to 7, we change the game so that we do not need s to respond to KDM queries made by $\mathcal{A}$.

In Game 2, we have

$$g_{k0} = \prod_{i\in[\ell]} (g_{ki})^{s_{ki}} = \prod_{i\in[\ell]} (g_i^{\gamma_{ki}})^{s_i \oplus \Delta_{ki}} \quad ,$$

where Δ_{ki} is the i-th bit of Δ_k for every $i \in [\ell]$. For every $i \in [\ell]$ and $k \in [n]$, we have

$$s_i \oplus \Delta_{ki} = \begin{cases} s_i & (\Delta_{ki} = 0) \\ 1 - s_i & (\Delta_{ki} = 1) \end{cases} .$$

Thus, by defining

$$\delta_{ki} = \begin{cases} 1 & (\Delta_{ki} = 0) \\ -1 & (\Delta_{ki} = 1) \end{cases} , \tag{4}$$

for every $i \in [\ell]$ and $k \in [n]$, we have

$$g_{k0} = \prod_{i\in\Delta_k} g_i^{\gamma_{ki}} \cdot \prod_{i\in[\ell]} g_i^{\delta_{ki}\gamma_{ki}s_i} = \prod_{i\in\Delta_k} g_{ki} \cdot \prod_{i\in[\ell]} g_i^{\delta_{ki}\gamma_{ki}s_i}$$

for every $k \in [n]$, where $\prod_{i\in\Delta_k} X_i$ denotes $\prod_{i\in[\ell]} X_i^{\Delta_{ki}}$.[8]

Game 3: Same as Game 2 except that the challenger uses $\delta_{ki}\gamma_{ki}$ instead of γ_{ki} for every $i \in [\ell]$ and $k \in [n]$. More precisely, the challenger computes $g_{ki} \leftarrow g_i^{\delta_{ki}\gamma_{ki}}$ for every $i \in [\ell]$ and $k \in [n]$, and $g_{k0} \leftarrow \prod_{i\in\Delta_k} g_{ki} \cdot \prod_{i\in[\ell]} g_i^{\gamma_{ki}s_i}$ for every $k \in [n]$. Note that $\delta_{ki} \cdot \delta_{ki} = 1$ for every $i \in [\ell]$ and $k \in [n]$.

If γ_{ki} distributes uniformly at random, then so does $\delta_{ki}\gamma_{ki}$ for every $i \in [\ell]$ and $k \in [n]$. Therefore, we have $|\Pr[\mathtt{SUC}_2] - \Pr[\mathtt{SUC}_3]| = 0$.

Game 4: Same as Game 3 except how the challenger computes $g_{k0}, g_{k1}, \ldots, g_{k\ell}$ for every $k \in [n]$. The challenger generates γ_k for every $k \in [n]$ and computes $g_{ki} \leftarrow g_i^{\gamma_k\delta_{ki}}$ for every $i \in [\ell]$ and $k \in [n]$. Moreover, the challenger computes $g_{k0} \leftarrow \prod_{i\in\Delta_k} g_{ki} \cdot \prod_{i\in[\ell]} g_i^{\gamma_k s_i}$ for every $k \in [n]$.

$|\Pr[\mathtt{SUC}_3] - \Pr[\mathtt{SUC}_4]| = \mathsf{negl}(\lambda)$ holds since $\left(g_1^{\gamma_{k1}}, \ldots, g_\ell^{\gamma_{k\ell}}\right)$ and $\left(g_1^{\gamma_k}, \ldots, g_\ell^{\gamma_k}\right)$ are computationally indistinguishable by the DDH assumption for every $k \in [n]$.

Below, we let $g_0 = \prod_{i\in[\ell]} g_i^{s_i}$. In Game 4, for every $k \in [n]$, we have

$$g_{ki} = g_i^{\gamma_k\delta_{ki}} \ (i \in [\ell]) \ , \text{ and}$$

$$g_{k0} = \prod_{i\in\Delta_k} g_{ki} \cdot \prod_{i\in[\ell]} g_i^{\gamma_k s_i} = \left(\prod_{i\in\Delta_k} g_{ki}\right) \cdot g_0^{\gamma_k} \quad .$$

[8] That is, $\prod_{i\in\Delta_k} X_i$ denotes the summation of X_i over positions i such that $\Delta_{ki} = 1$.

Then, the answer to a KDM query $\left(j, \left(\{a_k^0\}_{k\in[n]}, f^0\right), \left(\{a_k^1\}_{k\in[n]}, f^1\right)\right)$ in Game 4 is $\mathsf{Enc}_{\mathsf{cca}}\left(\mathsf{cpk}_j, \left(\{c_i\}_{i\in[\ell]}, d, \pi\right)\right)$, where

$$c_i = (g_{ji})^r = (g_i^r)^{\gamma_j \delta_{ji}} \ \ (i \in [\ell]) \ , \ d = g^{\sum_{k\in[k]} \langle a_k^b, s_k\rangle + f^b(\{v_k\}_{k\in[n]})} \cdot (g_{j0})^r \ ,$$
$$\pi = (\hat{g}_{j0})^r = \prod_{i\in[\ell]} c_i^{x_{ji}} \ , \text{ and } r \xleftarrow{\mathsf{r}} \mathbb{Z}_p \ .$$

We also have

$$\sum_{k\in[n]} \left\langle a_k^b, s_k \right\rangle = \sum_{k\in[n]} \left\langle a_k^b, s \oplus \Delta_k \right\rangle = \sum_{k\in[n]} \sum_{i\in\Delta_k} a_{ki}^b + \sum_{k\in[n]} \sum_{i\in[\ell]} a_{ki}^b \delta_{ki} s_i \ ,$$

where summation is done over $\mathbb{Z}$ and a_{ki}^b is the i-th bit of a_k^b for every $i \in [\ell]$. Thus, by defining

$$Y^b = \sum_{k\in[n]} \sum_{i\in\Delta_k} a_{ki}^b + f^b\left(\{v_k\}_{k\in[n]}\right) \text{ and } \mu_i^b = \sum_{k\in[n]} a_{ki}^b \delta_{ki} \ (i \in [\ell]) \ , \tag{5}$$

we have

$$\begin{aligned} d &= g^{Y^b + \sum_{i\in[\ell]} \mu_i^b s_i} \cdot \left(\prod_{i\in\Delta_j} g_{ji}\right)^r \cdot \left(\prod_{i\in[\ell]} g_i^{s_i}\right)^{\gamma_j r} \\ &= g^{Y^b} \cdot \prod_{i\in\Delta_j} c_i \cdot \prod_{i\in[\ell]} \left(g^{\mu_i^b} \cdot (g_i^r)^{\gamma_j}\right)^{s_i} \ . \end{aligned}$$

Note that Y^b and $\{\mu_i^b\}_{i\in[\ell]}$ are computed from $\left(\{a_k^b\}_{k\in[n]}, f^b\right)$ and $\{\delta_{ki}\}_{i\in[\ell],k\in[n]}$.

Hereafter, we show the difference from the previous game by colored parts.

Game 5: Same as Game 4 except how the challenger responds to KDM queries.

For a KDM query $\left(j, \left(\{a_k^0\}_{k\in[n]}, f^0\right), \left(\{a_k^1\}_{k\in[n]}, f^1\right)\right)$ made by $\mathcal{A}$, the challenger responds as follows.

1. Compute Y^b and $\{\mu_i^b\}_{i\in[\ell]}$ as Equation 5.
2. Generate $r_1, \ldots, r_\ell \xleftarrow{\mathsf{r}} \mathbb{Z}_p$.
3. Compute $c_i \leftarrow (g_i^{r_i})^{\gamma_j \delta_{ji}}$ for every $i \in [\ell]$.
4. Compute $d \leftarrow g^{Y^b} \cdot \prod_{i\in\Delta_j} c_i \cdot \prod_{i\in[\ell]} \left(g^{\mu_i^b} \cdot (g_i^{r_i})^{\gamma_j}\right)^{s_i}$.
5. Compute $\pi \leftarrow \prod_{i\in[\ell]} c_i^{x_{ji}}$.
6. Return $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cca}}\left(\mathsf{cpk}_j, \left(\{c_i\}_{i\in[\ell]}, d, \pi\right)\right)$ and add (j, CT) to L_{kdm}.

By the DDH assumption, $(g_1^r, \ldots, g_\ell^r)$ and $\left(g_1^{r_1}, \ldots, g_\ell^{r_\ell}\right)$ are computationally indistinguishable. Thus, we have $|\Pr[\mathsf{SUC}_4] - \Pr[\mathsf{SUC}_5]| = \mathsf{negl}(\lambda)$.

Game 6: Same as Game 5 except how the challenger responds to KDM queries.

For a KDM query $\left(j, \left(\{a_k^0\}_{k\in[n]}, f^0\right), \left(\{a_k^1\}_{k\in[n]}, f^1\right)\right)$ made by $\mathcal{A}$, the challenger responds as follows.

1. Compute Y^b and $\{\mu_i^b\}_{i\in[\ell]}$ as Equation 5.
2. Generate $r_1, \ldots, r_\ell \xleftarrow{\mathsf{r}} \mathbb{Z}_p$.
3. Compute $c_i \leftarrow \left(g^{-\mu_i^b} \cdot g_i^{r_i}\right)^{\delta_{ji}}$ for every $i \in [\ell]$.
4. Compute $d \leftarrow g^{Y^b} \cdot \prod_{i\in\Delta_j} c_i \cdot \prod_{i\in[\ell]} g_i^{r_i s_i}$.
5. Compute $\pi \leftarrow \prod_{i\in[\ell]} c_i^{x_{ji}}$.
6. Return $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cca}}\left(\mathsf{cpk}_j, \left(\{c_i\}_{i\in[\ell]}, d, \pi\right)\right)$ and add (j, CT) to L_{kdm}.

We can make this change in two steps. We first replace $g_i^{r_i\gamma_j}$ with $g_i^{r_i}$. We then replace $g_i^{r_i}$ with $g^{-\mu_i^b} \cdot g_i^{r_i}$. Since r_i is uniformly at random for every $i \in [\ell]$, the answer to a KDM query made by $\mathcal{A}$ identically distributes between Game 5 and 6. Thus, we have $|\Pr[\mathtt{SUC}_5] - \Pr[\mathtt{SUC}_6]| = 0$.

Game 7: Same as Game 6 except how the challenger responds to KDM queries.

For a KDM query $\left(j, \left(\{a_k^0\}_{k\in[n]}, f^0\right), \left(\{a_k^1\}_{k\in[n]}, f^1\right)\right)$ made by $\mathcal{A}$, the challenger responds as follows.

1. Compute Y^b and $\{\mu_i^b\}_{i\in[\ell]}$ as Equation 5.
2. Generate $r \xleftarrow{\mathsf{r}} \mathbb{Z}_p$.
3. Compute $c_i \leftarrow \left(g^{-\mu_i^b} \cdot g_i^{r}\right)^{\delta_{ji}}$ for every $i \in [\ell]$.
4. Compute $d \leftarrow g^{Y^b} \cdot \prod_{i\in\Delta_j} c_i \cdot \prod_{i\in[\ell]} g_i^{r s_i}$.
5. Compute $\pi \leftarrow \prod_{i\in[\ell]} c_i^{x_{ji}}$.
6. Return $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cca}}\left(\mathsf{cpk}_j, \left(\{c_i\}_{i\in[\ell]}, d, \pi\right)\right)$ and add (j, CT) to L_{kdm}.

By the DDH assumption, $(g_1^r, \ldots, g_\ell^r)$ and $\left(g_1^{r_1}, \ldots, g_\ell^{r_\ell}\right)$ are computationally indistinguishable. Thus, $|\Pr[\mathtt{SUC}_6] - \Pr[\mathtt{SUC}_7]| = \mathsf{negl}(\lambda)$ holds.

In Game 7, d generated to respond to a KDM query is of the form

$$d = g^{Y^b} \cdot \prod_{i\in\Delta_j} c_i \cdot \prod_{i\in[\ell]} g_i^{r s_i} = g^{Y^b} \cdot \left(\prod_{i\in\Delta_j} c_i\right) \cdot g_0^r \ .$$

Thus, we can reply to a KDM query made by $\mathcal{A}$ using g_0 instead of s in Game 7.

Next, we eliminate secret keys of Π_{cca} from the view of $\mathcal{A}$. For this aim, we make e_{k0} that is used to mask csk_k uniformly at random for every $k \in [n]$. We first change how the challenger responds to decryption queries made by $\mathcal{A}$.

Game 8: Same as Game 7 except how the challenger responds to decryption queries.

For a decryption query (j, CT) made by $\mathcal{A}$, the challenger returns $\perp$ to $\mathcal{A}$ if $(j, \mathsf{CT}) \in L_{\mathsf{kdm}}$, and otherwise responds as follows.

1. Compute $\left(\{c_i\}_{i\in[\ell]}, d, \pi\right) \leftarrow \mathsf{Dec}_{\mathsf{cca}}\left(\mathsf{csk}_j, \mathsf{CT}\right)$. If the decryption result is not in $\mathbb{G}^{\ell+2}$, return $\perp$. Otherwise, compute as follows.

2. Let $c_i = (g_{ji})^{r_i}$ for every $i \in [\ell]$.[9] If there exists $i' \in \{2, \ldots, \ell\}$ such that $r_1 \neq r_{i'}$, return $\bot$. Otherwise, respond as follows.
3. Return $\bot$ if $\pi \neq \prod_{i\in[\ell]} c_i^{x_{ji}}$ and otherwise $m \leftarrow \log_g \left(d \cdot \left(\prod_{i\in[\ell]} c_i^{s_{ji}} \right)^{-1} \right)$.

We define the following event in Game i ($i = 7, \ldots, 11$).

$\mathtt{BDQ}_i$: $\mathcal{A}$ makes a decryption query $(j, \mathsf{CT}) \notin L_{\mathsf{kdm}}$ which satisfies the following conditions, where $\left(\{c_i\}_{i\in[\ell]}, d, \pi \right) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{csk}_j, \mathsf{CT})$.

- $\left(\{c_i\}_{i\in[\ell]}, d, \pi \right) \in \mathbb{G}^{\ell+2}$. Then, let $c_i = g_{ji}^{r_i}$, where $r_i \in \mathbb{Z}_p$ for every $i \in [\ell]$.
- There exists $i' \in \{2, \ldots, \ell\}$ such that $r_1 \neq r_{i'}$.
- $\pi = \prod_{i\in[\ell]} c_i^{x_{ji}}$.

We call such a decryption query a "**bad decryption query**".

Games 7 and 8 are identical games unless $\mathcal{A}$ make a bad decryption query in each game. Therefore, we have $|\Pr[\mathtt{SUC}_7] - \Pr[\mathtt{SUC}_8]| \leq \Pr[\mathtt{BDQ}_8]$.

For every $k \in [n]$, we have

$$e_{ki} = (g_{ki})^{w_i} = (g_i^{w_i})^{\gamma_k \delta_{ki}} \quad (i \in [\ell])$$
$$e_{k0} = \prod_{i\in[\ell]} (e_{ki})^{s_{ki}} = \left(\prod_{i\in\Delta_k} e_{ki} \right) \prod_{i\in[\ell]} (e_{ki})^{\delta_{ki} s_i} = \left(\prod_{i\in\Delta_k} e_{ki} \right) \left(\prod_{i\in[\ell]} g_i^{w_i s_i} \right)^{\gamma_k} .$$

Note that $\delta_{ki} \cdot \delta_{ki} = 1$ for every $i \in [\ell]$ and $k \in [n]$.

Game 9: Same as Game 8 except that $e_0 \xleftarrow{\mathsf{r}} \mathbb{G}$ is used instead of $\prod_{i\in[\ell]} g_i^{w_i s_i}$.

The view of $\mathcal{A}$ in Games 8 and 9 can be perfectly simulated by

$$\left(g_1, \ldots, g_\ell, g_1^{w_1}, \ldots, g_\ell^{w_\ell}, \prod_{i\in[\ell]} g_i^{w_i s_i}, g_0 \right) \text{ and } \left(g_1, \ldots, g_\ell, g_1^{w_1}, \ldots, g_\ell^{w_\ell}, e_0, g_0 \right) ,$$

respectively, where $g_i \xleftarrow{\mathsf{r}} \mathbb{G}$ and $w_i \xleftarrow{\mathsf{r}} \mathbb{Z}_p$ for every $i \in [\ell]$, $s = s_1 \cdots s_\ell \xleftarrow{\mathsf{r}} \{0,1\}^\ell$, $e_0 \xleftarrow{\mathsf{r}} \mathbb{G}$, and $g_0 = \prod_{i\in[\ell]} g_i^{s_i}$. By the leftover hash lemma, the view of $\mathcal{A}$ in Game 8 is $2^{-\frac{\ell-2\lambda}{2}}$-close to that in Game 9. Thus, by setting $\ell = 3\lambda$, $|\Pr[\mathtt{SUC}_8] - \Pr[\mathtt{SUC}_9]| = \mathsf{negl}(\lambda)$ and $|\Pr[\mathtt{BDQ}_8] - \Pr[\mathtt{BDQ}_9]| = \mathsf{negl}(\lambda)$ hold.

There exists $\alpha_i \in \mathbb{Z}_p$ such that $g^{\alpha_i} = g_i$ for every $i \in [\ell]$. Then, for every $k \in [n]$, we have

$$g_{ki} = g^{\alpha_i \gamma_k \delta_{ki}} = (g^{\gamma_k})^{\alpha_i \delta_{ki}} \quad (i \in [\ell]) , \quad g_{k0} = \prod_{i\in[\ell]} (g_{ki})^{s_{ki}}$$
$$e_{ki} = (g_{ki})^{w_i} \ (i \in [\ell]) , \text{ and } e_{k0} = \left(\prod_{i\in\Delta_k} e_{ki} \right) e_0^{\gamma_k} .$$

[9] Note that such r_i exists unless g_{ji} is the identity element since $c_i \in \mathbb{G}$ for every $i \in [\ell]$. The probability that g_{ji} is the identity element is negligible. Thus, we ignore this issue for simplicity.

Game 10**:** Same as Game 9 except that for every $k \in [n]$, the challenger generates $e_{k0} \leftarrow \left(\prod_{i \in \Delta_k} e_{ki}\right) e_0^{z_k}$, where $z_k \xleftarrow{\mathsf{r}} \mathbb{Z}_p$.

$|\Pr[\mathtt{SUC}_9] - \Pr[\mathtt{SUC}_{10}]| = \mathsf{negl}(\lambda)$ holds since $(g, e_0, g^{\gamma_k}, e_0^{\gamma_k})$ and $(g, e_0, g^{\gamma_k}, e_0^{z_k})$ are computationally indistinguishable by the DDH assumption for every $k \in [n]$.

Moreover, we can efficiently check whether $\mathcal{A}$ makes a bad decryption query or not by using $\{\mathsf{csk}_k\}_{k \in [n]}$, $\{\alpha_i\}_{i \in [\ell]}$, and $\{x_{ki}\}_{i \in [\ell], k \in [n]}$. Therefore, we also have $|\Pr[\mathtt{BDQ}_9] - \Pr[\mathtt{BDQ}_{10}]| = \mathsf{negl}(\lambda)$ by the DDH assumption.

In Game 10, e_{k0} distributes uniformly at random for every $k \in [n]$. Therefore, $\mathcal{A}$ cannot obtain any information of csk_k from $u_k = e_{k0} \cdot \mathsf{csk}_k$ for every $k \in [n]$, and thus we can use IND-CCA security of Π_{cca}.

Game 11**:** Same as Game 10 except that the challenger responds to KDM queries made by $\mathcal{A}$ with $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cca}}\left(\mathsf{cpk}_j, 0^{(\ell+2)\cdot|g|}\right)$.

By the IND-CCA security of Π_{cca}, we obtain $|\Pr[\mathtt{SUC}_{10}] - \Pr[\mathtt{SUC}_{11}]| = \mathsf{negl}(\lambda)$.

Moreover, we can efficiently check whether $\mathcal{A}$ makes a bad decryption query or not by using decryption queries for Π_{cca}, $\{\alpha_i\}_{i \in [\ell]}$, and $\{x_{ki}\}_{i \in [\ell], k \in [n]}$. Thus, $|\Pr[\mathtt{BDQ}_{10}] - \Pr[\mathtt{BDQ}_{11}]| = \mathsf{negl}(\lambda)$ also holds by the IND-CCA security of Π_{cca}.

The value of b is information theoretically hidden from the view of $\mathcal{A}$ in Game 11. Thus, we have $\left|\Pr[\mathtt{SUC}_{11}] - \frac{1}{2}\right| = 0$.

In Game 11, $x_{k1}, \ldots, x_{k\ell}$ are hidden from the view of $\mathcal{A}$ except $\hat{g}_{k0} = \prod_{i \in [\ell]} (g_{ki})^{x_{ki}}$ for every $k \in [n]$. Note that $\mathcal{A}$ cannot obtain information of $x_{k1}, \ldots, x_{k\ell}$ other than $\hat{g}_{k0}$ through decryption queries for every $k \in [n]$. The reason is as follows. If a decryption query (j, CT) made by $\mathcal{A}$ is not a bad decryption query, there exists $r_1 \in \mathbb{Z}_p$ such that $c_i = (g_{ji})^{r_1}$ for every $i \in [\ell]$, where $\left(\{c_i\}_{i \in [\ell]}, d, \pi\right) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{csk}_j, \mathsf{CT})$. Then, we have

$$\prod_{i \in [\ell]} c_i^{x_{ji}} = \prod_{i \in [\ell]} (g_{ji})^{r_1 x_{ji}} = \left(\prod_{i \in [\ell]} (g_{ji})^{x_{ji}}\right)^{r_1} = (\hat{g}_{j0})^{r_1} \quad .$$

In addition, bad decryption queries made by $\mathcal{A}$ are replied with $\perp$ in Game 11. This means that for every $k \in [n]$, $\mathcal{A}$ cannot obtain information of $x_{k1}, \ldots, x_{k\ell}$ other than $\hat{g}_{k0}$ through decryption queries.

We estimate $\Pr[\mathtt{BDQ}_{11}]$. Let (j, CT) be a decryption query made by $\mathcal{A}$ and let $\left(\{c_i\}_{i \in [\ell]}, d, \pi\right) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{csk}_j, \mathsf{CT})$. Suppose that $\left(\{c_i\}_{i \in [\ell]}, d, \pi\right) \in \mathbb{G}^{\ell+2}$, $c_i = (g_{ji})^{r_i}$ for every $i \in [\ell]$, and there exists $i' \in \{2, \ldots, \ell\}$ such that $r_1 \neq r_{i'}$. The probability that this query is a bad decryption query is

$$\Pr_{x_{ji} \xleftarrow{\mathsf{r}} \mathbb{Z}_p}\left[\prod_{i \in [\ell]} c_i^{x_{ji}} = \pi \middle| \prod_{i \in [\ell]} (g_{ji})^{x_{ji}} = \hat{g}_{j0}\right] \quad . \tag{6}$$

This probability is the same as

$$\Pr_{x_{ji} \xleftarrow{\mathsf{r}} \mathbb{Z}_p}\left[\sum_{i \in [\ell]} \alpha_i \gamma_j \delta_{ji} r_i x_{ji} = \log_g \pi \bmod p \middle| \sum_{i \in [\ell]} \alpha_i \gamma_j \delta_{ji} x_{ji} = \log_g \hat{g}_{j0} \bmod p\right] \quad .$$

$\alpha_i \neq 0$ and $\gamma_j \neq 0$ holds for every $i \in [\ell]$ and $j \in [n]$ with high probability and thus we assume so. Then, two equations

$$\sum_{i\in[\ell]} \alpha_i\gamma_j\delta_{ji}r_i x_{ji} = \log_g \pi \bmod p \text{ and } \sum_{i\in[\ell]} \alpha_i\gamma_j\delta_{ji}x_{ji} = \log_g \hat{g}_{j0} \bmod p$$

are linearly independent, and thus the probability shown in Equation 6 is $\frac{1}{p}$. Therefore, we obtain $\Pr[\mathtt{BDQ}_{11}] = \mathsf{negl}(\lambda)$.

From the above arguments, we have

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{kdmcca}}_{\Pi_{\mathsf{ddh}},\mathcal{F}_{\mathsf{ddh}},\mathcal{A},n}(\lambda) &= \left|\Pr[\mathtt{SUC}_0] - \frac{1}{2}\right| \\
&= \sum_{t=0}^{10} |\Pr[\mathtt{SUC}_t] - \Pr[\mathtt{SUC}_{t+1}]| + \left|\Pr[\mathtt{SUC}_{11}] - \frac{1}{2}\right| \\
&= \sum_{t\in\{0,\ldots,10\},t\neq 7} |\Pr[\mathtt{SUC}_t] - \Pr[\mathtt{SUC}_{t+1}]| + |\Pr[\mathtt{SUC}_7] - \Pr[\mathtt{SUC}_8]| \\
&= \sum_{t\in\{0,\ldots,10\},t\neq 7} |\Pr[\mathtt{SUC}_t] - \Pr[\mathtt{SUC}_{t+1}]| + \sum_{t=8}^{10} |\Pr[\mathtt{BDQ}_t] - \Pr[\mathtt{BDQ}_{t+1}]| + \Pr[\mathtt{BDQ}_{11}] \\
&= \mathsf{negl}(\lambda) \ .
\end{aligned}$$

Since the choice of $\mathcal{A}$ and n is arbitrary, Π_{ddh} is $\mathcal{F}_{\mathsf{ddh}}$-KDM-CCA secure. □ (**Theorem 4**)

Remark 5 (The multi user security of the QR and DCR based schemes) Our QR and DCR based constructions are based on those proposed by Brakerski and Goldwasser [5]. If we allow the length of secret keys to depend on the number of users n, we can prove that our QR and DCR based constructions are KDM$^{(n)}$-CCA secure using a technique similar to Brakerski and Goldwasser.

To prove KDM$^{(n)}$-CCA security, we need to eliminate encrypted n secret keys of the outer IND-CCA secure PKE scheme contained in secret keys of the KDM-CCA secure scheme. In the above proof of DDH based scheme, by using the self reducibility of the DDH problem, we complete such a task by making a single group element $\prod_{i\in[\ell]} g_i^{w_i s_i}$ random using the leftover hash lemma.

However, when proving the KDM$^{(n)}$-CCA security of the QR and DCR based constructions, to complete such a task, we need to make n group elements random using the leftover hash lemma. Therefore, in that case, we need to set the length of secret keys depending on n similarly to the proof of KDM$^{(n)}$-CPA security by Brakerski and Goldwasser.

References

[1] J. Alperin-Sheriff and C. Peikert. Circular and KDM security for identity-based encryption. *PKC 2012*, *LNCS* 7293, pp. 334–352. 2012.

[2] B. Applebaum. Key-dependent message security: Generic amplification and completeness. *EUROCRYPT 2011*, *LNCS* 6632, pp. 527–546. 2011.

[3] J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. *SAC 2002*, *LNCS* 2595, pp. 62–75. 2003.

[4] D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. *CRYPTO 2008*, *LNCS* 5157, pp. 108–125. 2008.

[5] Z. Brakerski and S. Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). *CRYPTO 2010*, *LNCS* 6223, pp. 1–20. 2010.

[6] J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. *EUROCRYPT 2009*, *LNCS* 5479, pp. 351–368. 2009.

[7] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. *EUROCRYPT 2001*, *LNCS* 2045, pp. 93–118. 2001.

[8] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. *EUROCRYPT 2002*, *LNCS* 2332, pp. 45–64. 2002.

[9] S. Han, S. Liu, and L. Lyu. Efficient KDM-CCA secure public-key encryption for polynomial functions. *ASIACRYPT 2016, Part II*, *LNCS* 10032, pp. 307–338. 2016.

[10] D. Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. *EUROCRYPT 2013*, *LNCS* 7881, pp. 520–536. 2013.

[11] S. Hohenberger, A. B. Lewko, and B. Waters. Detecting dangerous queries: A new approach for chosen ciphertext security. *EUROCRYPT 2012*, *LNCS* 7237, pp. 663–681. 2012.

[12] F. Kitagawa, T. Matsuda, G. Hanaoka, and K. Tanaka. Completeness of single-bit projection-KDM security for public key encryption. *CT-RSA 2015*, *LNCS* 9048, pp. 201–219. 2015.

[13] K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. *CRYPTO 2004*, *LNCS* 3152, pp. 426–442. 2004.

[14] X. Lu, B. Li, and D. Jia. KDM-CCA security from RKA secure authenticated encryption. *EUROCRYPT 2015, Part I*, *LNCS* 9056, pp. 559–583. 2015.

[15] S. Myers and a. shelat. Bit encryption is complete. In *50th FOCS*, pp. 607–616. 2009.

[16] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pp. 427–437. 1990.

[17] V. Shoup. Using hash functions as a hedge against chosen ciphertext attack. *EUROCRYPT 2000*, *LNCS* 1807, pp. 275–288. 2000.

[18] V. Shoup. *A computational introduction to number theory and algebra*. Cambridge University Press, 2006.

[19] H. Wee. KDM-security via homomorphic smooth projective hashing. *PKC 2016, Part II*, *LNCS* 9615, pp. 159–179. 2016.