Chosen Ciphertext Secure Keyed-Homomorphic Public-Key Encryption [*]

Keita Emura[†] Goichiro Hanaoka[‡] Koji Nuida[§] Go Ohtake[¶]
Takahiro Matsuda[||] Shota Yamada[**]

May 13, 2014

Abstract

In homomorphic encryption schemes, anyone can perform homomorphic operations, and therefore, it is difficult to manage when, where and by whom they are performed. In addition, the property that anyone can "freely" perform the operation inevitably means that ciphertexts are malleable, and it is well-known that adaptive chosen ciphertext (CCA) security and the homomorphic property can never be achieved simultaneously. In this paper, we show that CCA security and the homomorphic property can be simultaneously handled in situations that the user(s) who can perform homomorphic operations on encrypted data should be controlled/limited, and propose a new concept of homomorphic public-key encryption, which we call *keyed-homomorphic public-key encryption* (KH-PKE). By introducing a secret key for homomorphic operations, we can control who is allowed to perform the homomorphic operation. To construct KH-PKE schemes, we introduce a new concept, *transitional universal property*, and present a practical KH-PKE scheme from the DDH assumption. For ℓ-bit security, our DDH-based KH-PKE scheme yields only ℓ-bit longer ciphertext size than that of the Cramer–Shoup PKE scheme.

Keywords: homomorphic public key encryption, CCA2 security, hash proof system

1 Introduction

1.1 Background and Motivation

In homomorphic encryption schemes, homomorphic operations can be performed on encrypted plaintexts without decrypting the corresponding ciphertexts. Owing to this attractive property, several homomorphic public key encryption (PKE) schemes have been proposed [15, 20, 33]. Furthermore, fully homomorphic encryption (FHE) that allows a homomorphic operation with respect to any circuit, has recently been proposed by Gentry [19]. This has had a resounding impact not only in the cryptographic research community, but also in the business community. One of the reasons for such a big impact is that FHE is suitable for ensuring security in cloud environments (e.g., encrypted data stored in a database can be updated without any decryption procedure).

[*] An extended abstract appears in the 16th International Conference on Practice and Theory in Public Key Cryptography (PKC 2013) [16]. We have found some bug in the scheme appeared in the proceeding version, and have corrected it in this version. See Section 1.4 for more details on the differences.

[†] National Institute of Information and Communications Technology (NICT), Japan. `k-emura@nict.go.jp`

[‡] National Institute of Advanced Industrial Science and Technology (AIST), Japan. `hanaoka-goichiro@aist.go.jp`

[§] National Institute of Advanced Industrial Science and Technology (AIST), Japan. `k.nuida@aist.go.jp`

[¶] Japan Broadcasting Corporation, Japan. `ohtake.g-fw@nhk.or.jp`

[||] National Institute of Advanced Industrial Science and Technology (AIST), Japan. The 5th author was supported by a JSPS Fellowship for Young Scientists. `t-matsuda@aist.go.jp`

[**] National Institute of Advanced Industrial Science and Technology (AIST), Japan. The 6th author is supported by a JSPS Fellowship for Young Scientists. `yamada-shota@aist.go.jp`

Improvement in the security of homomorphic encryption will lead to wider deployment of cloud-type applications, whereas the property that anyone can "freely" perform homomorphic operations inevitably means that ciphertexts are malleable. Therefore, it is well-known that adaptive chosen ciphertext (CCA2) security and the homomorphic property can never be achieved simultaneously. In other words, security is sacrificed in exchange for the homomorphic property. Although several previous works (e.g., [2, 8, 21, 35, 36]) have attempted to construct homomorphic PKE schemes that offer security close to CCA2 security while retaining the homomorphic property, these schemes only guarantee security at limited levels. Note that not all functionalities of conventional homomorphic encryption are indispensable for real-world applications, and therefore there is the possibility of realizing a desirable security level by appropriately selecting the functionalities of conventional homomorphic encryption.

Here, we point out that the underlying cause of the incompatibility of CCA2 security and the homomorphic property, lies in the setting that any user can use the homomorphic property, and it is worth discussing whether the free availability of homomorphic operations is an indispensable functionality in real-world applications. For example, consider the situation where some data encrypted by a homomorphic PKE scheme is stored in a public database (e.g., public cloud computing environment) and it is modified by homomorphic operations. If anyone can perform a homomorphic operation, then it is hard to reduce the risk of unexpected changes to the encrypted data in the database in which resources are dynamically allocated. Even in a closed environment (e.g., private cloud computing environment), we cannot rule out the possibility of unexpected changes to a user's data by any user who is authorized to access the database. Of course, it is possible to protect such unexpected modification of encrypted data by setting access permissions of each user appropriately. However, in cloud environments, security of outsourced data storages may not be assured. Therefore, such access control functionality should be included in encrypted data itself.

From the above consideration, we see that the property that anyone can perform homomorphic operations not only inhibits the realization of CCA2 security, but also introduces the problem of unexpected modification of encrypted data.

1.2 Our Contribution

In this paper, we show that CCA2 security and the homomorphic property can be simultaneously handled in situations that the user(s) who can perform homomorphic operations should be controlled. Specifically, we propose a new concept of homomorphic PKE, which we call *keyed-homomorphic public-key encryption* (KH-PKE), that has the following properties: (1) in addition to a conventional public/decryption key pair (pk, sk_d), another secret key for the homomorphic operation (denoted by sk_h) is introduced, (2) homomorphic operations cannot be performed without using sk_h, and (3) ciphertexts cannot be decrypted using only sk_h. Interestingly, KH-PKE implies conventional homomorphic PKE, since the latter can be implemented by publishing sk_h of KH-PKE.

To construct KH-PKE schemes, we introduce a new concept, *transitional universal property*, which can be obtained from any diverse group system [13], and present a number of KH-PKE schemes through hash proof systems (HPSs) [13].

Our Scenarios: Here we introduce situations that the user(s) who can perform homomorphic operations should be controlled/limited. For example, in the situation where encrypted data is stored in a public database, an owner of the data gives sk_h to the database manager, who updates the encrypted data after authentication of users. No outsider can modify the encrypted data in the public database without having sk_h. As another example, by considering sk_h, a counter can take over the role of aggregating an audience survey, voting, and so on. An advantage of separating ballot-counting and ballot-aggregation is that it is possible to reduce the aggregation costs of the counter and to collect the ballot results for individual areas, groups, and communities. We can also consider an application of KH-PKE to prevent illegal distribution of data. A content creator gives sk_h to a digital content provider and the provider embeds some data (e.g., a water mark) for protecting the content against illegal copying, a certification for ownership of the content, and/or a distribution route.

Naive Construction and its Limitations: One might think that the functionality and the security of KH-PKE can be achieved by using the following double encryption methodology: A ciphertext of an "inner" CCA1 secure homomorphic PKE scheme is encrypted by an "outer" CCA2 secure PKE scheme, and the decryption key of the CCA2 secure PKE scheme is used as sk_h. However, this naive construction is not secure in the sense of our security definition. Taking into account the exposure of the homomorphic operation key sk_h, an adversary can request sk_h to be exposed in our security definition. The adversary is allowed to use the decryption oracle "even after the challenge phase", just before the adversary requests sk_h. However, no such decryption query is allowed in the CCA1 security of the underlying "inner" scheme, and therefore it seems hard to avoid this problem.

Even if we turn a blind eye to the above problem, it is obvious that efficiency of the naive construction is roughly equal to the total costs of the building block PKE schemes. On the other hand, the efficiency of our KH-PKE instantiations is very close to the corresponding (non-keyed-homomorphic) PKE schemes based on HPSs. In particular, the efficiency of our decisional Diffie-Hellman (DDH)-based KH-PKE scheme is comparably efficient as the Cramer–Shoup PKE (CS) scheme [11], where for ℓ-bit security, our scheme yields only ℓ-bit longer ciphertext size than that of the CS PKE scheme. Whereas the naive construction yields at least 5ℓ-bit longer ciphertext size even if we choose the Kurosawa–Desmedt (KD) PKE scheme [29] and the Cramer–Shoup lite PKE scheme [11] that seems the most efficient combination under the DDH assumption. We give the comparison in Table 1 in Section 5.3.

To sum up, our construction is superior to the naive construction from both security and efficiency perspectives.

Our Methodology: As a well-known result, CCA2-secure PKE can be constructed via a HPS [13] which has two projective hash families as its internal structure: A *universal*$_2$ projective hash and a *smooth* projective hash. Also it is known that a weaker property of universal$_2$, that is called *universal*$_1$ property, was shown to be useful for achieving CCA1-secure PKE [28], and universal$_1$ property (and smooth property also) does not contradict the homomorphic property. That is, our aim seems to be achieved if we can establish a switching mechanism from universal$_2$ to universal$_1$. Moreover, we can simulate the decryption oracle even after the challenge phase and after revealing sk_h since the simulator knows all secret keys in the security proof.

In this paper, we show such a mechanism, which we call transitional universal property, can be obtained from any diverse group system [13], then we propose a generic construction of KH-PKE through HPSs. Moreover, as an implication result, KH-PKE is implied by CPA-secure homomorphic PKE (with cyclic-group ciphertext space) which implies diverse group systems [23].

Instantiations: In this paper, we present practical KH-PKE schemes from the DDH assumption and the decisional composite residuosity (DCR) assumption, respectively. Other KH-PKE schemes based on the decisional linear (DLIN) assumption from the Shacham HPS [37], and based on the decisional bilinear Diffie-Hellman (DBDH) assumption from the Galindo-Villar HPS [17], and an identity-based analogue of KH-PKE, called *keyed-homomorphic identity-based encryption* (KH-IBE) and its concrete construction from the Gentry IBE scheme [18] will be given in the full version of this paper.

1.3 Related Work

Several previous works have attempted to construct homomorphic PKE schemes that provide security close to CCA2 security, while retaining the homomorphic property. Canetti et al. [8] considered the notion of replayable CCA (RCCA), which leaves a room for an adversary who is given two ciphertexts (C, C'), to gain information on whether C' was derived from C. (Modified RCCA notions have also been proposed [21, 35].) In the RCCA security game, the decryption oracle given to an adversary is restricted in such a way that the challenge ciphertext and ciphertexts derived from the challenge ciphertext cannot be queried to the oracle. Similarly, in benignly-malleable (gCCA) security [2, 38], ciphertexts related to the challenge one cannot be input to the decryption oracle. Therefore, RCCA and gCCA are strictly weaker notions than CCA2, and may not be sufficient if the encryption scheme is used as a building block for higher level protocols/systems.

In [36], Prabhakaran and Rosulek proposed homomorphic CCA (HCCA) security, where only the expected operation, and no other operations, can be performed for any ciphertext. (Targeted malleability, which is a similar concept to HCCA, was considered in [6].) In addition, they also showed that CCA2, gCCA, and RCCA are special cases of HCCA. Note that HCCA does not handle the homomorphic property and CCA2 security simultaneously, since anyone can perform the homomorphic operation. Chase et al. [10] showed that controlled-malleable non-interactive zero-knowledge can be used as a general tool for achieving RCCA and HCCA security.

Embedding a ciphertext of homomorphic PKE into that of CCA2-secure PKE, was considered in [32, 5]. Note that their embedding encryption methods are nothing more than protecting a ciphertext of homomorphic PKE by that of CCA2 PKE, and therefore no homomorphic operation can be performed on embedded ciphertexts. Meanwhile, in our KH-PKE, even after performing the homomorphic operation, a ciphertext is still valid.

Barbosa and Farshim [3] proposed delegatable homomorphic encryption (DHE). The difference between KH-PKE and DHE is that in DHE a trusted authority (TA) issues a token to control the capability to evaluate circuits f over encrypted data M to untrusted evaluators. Furthermore, their security definitions of DHE (input/output privacy (TA-IND-CPA) and evaluation security (IND-EVAL2)) do not allow an adversary to access the decryption oracle and the evaluation oracle (the oracle for homomorphic operation) simultaneously. We note that although Barbosa and Farshim defined verifiability (VRF-CCA2), where no homomorphic operation can be performed without issuing a corresponding token, KH-CCA security for KH-PKE defined in this paper guarantees a similar level of security, since if there exists an adversary that can perform the homomorphic operation without using sk_h, then the adversary can break the KH-CCA security.

Following our work, Libert et al. [31] proposed a KH-PKE scheme for supporting threshold decryption and publicly verifiable ciphertexts. They apply linearly homomorphic structure-preserving signatures [30] to quasi-adaptive non-interactive zero-knowledge (QA-NIZK) proofs [25], propose QA-NIZK proofs with unbounded simulation-soundness (USS), and construct a KH-PKE scheme by applying USS. Their KH-PKE scheme (with multiplicative homomorphic operations) is secure under the DLIN assumption (and strong unforgeability of the underlying one-time signature).

In the signature context, recently, Abe et al. [1] considered selective randomizability, where a strongly unforgeable signature to be randomized with the help of a randomization token, and a randomizable signature is still existentially unforgeable.

1.4 Differences from the Proceedings Version [16]

In the proceedings version [16], there were several bugs. Specifically, in the second last component $\widehat{\pi}$ of a ciphertext of the generic construction (in [16, Fig. 1]) and that of the DDH-based construction (in [16, Fig. 2]) could be malleable, which could lead to CCA attacks on the schemes. Furthermore, the evaluation algorithms for these constructions were (although "correct" in terms of the functionality of KH-PKE) not properly designed in the sense that in the CCA security game, the result of the "evaluation oracle" for challenge ciphertext-dependent inputs could leak some information. Moreover, we did not properly state the requirement of the second hash function (denoted by TCR_2 in [16, Fig. 2]) used to "compress" the proof value $\widetilde{\pi}$ to reduce the ciphertext size.

We fix these bugs in the current version: First, we reconsider the proposed generic construction (in Section 4): (1) the first proof value $\widetilde{\pi}$ in the generic construction (in Fig. 1) is now made dependent on the second proof value $\widehat{\pi}$, and (2) the evaluation algorithm Eval computes and "adds" a new ciphertext of 0. These modifications enable us to prove our modified proposed constructions to be CCA secure.[1] According to these modifications, we do not have to newly define *homomorphic transitional universal hash family.* Instead, we introduce *transitional universal property* of the pair of two HPSs. We also apply the similar modifications

[1] In the previous eprint version (20130618:085049 (posted 18-Jun-2013 08:50:49 UTC)), we considered the first modification only, and therefore we achieved a weaker security which we call weak KH-CCA security, where no challenge-ciphertext-related ciphertext is allowed to input the evaluation oracle. In this version, we can achieve KH-CCA security due to the second modification.

to our DDH-based scheme (in Section 5.3). We would like to emphasize that the modifications here do not incur additional computational cost or increase of the ciphertext size for both of our constructions.

Second, we reconsider the condition of the function that is used to "compress" the proof value $\widetilde{\pi}$ in our DDH-based construction, and newly introduce the notion of *smoothness* for a function. This is a statistical property that roughly ensures that the "min-entropy" of the output of a function (for uniformly random input) is sufficiently high, and thus it is information-theoretically hard to guess the output of a function for random inputs. We also show that natural cryptographic functions, a one-way function (OWF), an always second-preimage resistant (aSec secure) hash function [34], and a key derivation function (KDF) [14], have the property, and thus in practice we can use (an appropriate modification of) cryptographic hash functions such as SHA-series).

2 Preliminaries

In this section, we review the basic notations and definitions related to HPSs (mostly following [13] but slightly customized for our convenience).

Throughout this paper, PPT denotes *probabilistic polynomial time.* If n is a natural number, then $[n] = \{1, \dots, n\}$. If D is a probability distribution (over some set), then $[D]$ denotes its support, i.e. $[D] = \{x | \Pr_{x' \leftarrow D}[x' = x] > 0\}$. Let $\mathbf{X} = \{X_\ell\}_{\ell \geq 0}$ and $\mathbf{Y} = \{Y_\ell\}_{\ell \geq 0}$ be sequences of random variables X_ℓ and Y_ℓ, respectively, defined over a same finite set. As usual, we say that $\mathbf{X}$ and $\mathbf{Y}$ are *statistically (resp. computationally) indistinguishable* if $|\Pr[\mathcal{A}(X_\ell) = 1] - \Pr[\mathcal{A}(Y_\ell) = 1]|$ is negligible in ℓ for any computationally unbounded (resp. PPT) algorithm $\mathcal{A}$. Furthermore, we say that $\mathbf{X}$ and $\mathbf{Y}$ are ϵ-close if the statistical distance of X_ℓ and Y_ℓ is at most $\epsilon = \epsilon(\ell)$. For a finite set B_ℓ and its subset B'_ℓ indexed (often implicitly) by $\ell \geq 0$, we say that B'_ℓ is *approximately samplable relative to* B_ℓ, if there is a sequence of random variables on B_ℓ which is polynomial-time samplable and is statistically indistinguishable from the uniform random variable on B'_ℓ.

Projective Hash Families: Let X, Π, K, and S be finite non-empty sets, X' be a non-empty subset of X, and L be a proper subset of X (i.e., $L \subset X$ and $L \neq X$). Furthermore, let $H = \{H_k : X \to \Pi\}_{k \in K}$ be a collection of hash functions indexed by $k \in K$, and $\alpha : K \to S$ be a function. We say that $\mathbf{H} = (H, K, X, X', L, \Pi, S, \alpha)$ is a *projective hash family* for (X, X', L), if for all $k \in K$, the action of H_k on the subset L is uniquely determined by $\alpha(k) \in S$. When $X' = X$, we may omit the symbol X' in the notations above.

Let $\mathbf{H} = (H, K, X, X', L, \Pi, S, \alpha)$ be a projective hash family, and let $0 \leq \epsilon \leq 1$. We recall the following properties of a projective hash family: We say that $\mathbf{H}$ is ϵ*-universal*$_1$ if for all $s \in S$, $x \in X \setminus L$, and $\pi \in \Pi$, it holds that

$$\Pr_{k \xleftarrow{\$} K}[H_k(x) = \pi \wedge \alpha(k) = s] \leq \epsilon \cdot \Pr_{k \xleftarrow{\$} K}[\alpha(k) = s] \ .$$

We say that $\mathbf{H}$ is ϵ*-universal*$_2$ if for all $s \in S$, $x, x^* \in X \setminus L$ with $x^* \neq x$, and $\pi, \pi^* \in \Pi$, it holds that

$$\Pr_{k \xleftarrow{\$} K}[H_k(x) = \pi \wedge H_k(x^*) = \pi^* \wedge \alpha(k) = s] \leq \epsilon \cdot \Pr_{k \xleftarrow{\$} K}[H_k(x^*) = \pi^* \wedge \alpha(k) = s] \ .$$

We say that $\mathbf{H}$ is ϵ*-smooth* if the following two distributions are ϵ-close:

$$\{k \xleftarrow{\$} K;\ x \xleftarrow{\$} X \setminus L\ :\ (\alpha(k), x, H_k(x))\ \} \text{ and } \{k \xleftarrow{\$} K;\ x \xleftarrow{\$} X \setminus L;\ \pi \xleftarrow{\$} \Pi\ :\ (\alpha(k), x, \pi)\ \} \ .$$

We also introduce a variant of the smoothness property: Suppose that Π is an abelian group (written in additive form), and let Π' be a subgroup of Π. In this case, we say that $\mathbf{H}$ is ϵ*-smooth relative to* (X', Π'), if the following two distributions are ϵ-close:

$$\{k \xleftarrow{\$} K;\ x \xleftarrow{\$} X' \setminus L\ :\ (\alpha(k), x, H_k(x))\ \} \text{ and } \{k \xleftarrow{\$} K;\ x \xleftarrow{\$} X' \setminus L;\ \pi \xleftarrow{\$} \Pi'\ :\ (\alpha(k), x, H_k(x) + \pi)\ \} \ .$$

We note that, when $\Pi' = \Pi$, the term $H_k(x) + \pi$ in the latter probability distribution above can be replaced with π, since now $H_k(x) + \pi$ is also uniformly random over Π. Hence, the notion of smoothness relative to (X', Π') above is in fact a generalization of the smoothness.

If a projective hash family is ϵ-universal$_1$ (resp. -universal$_2$, -smooth) for a negligible ϵ, then we simply call the projective hash family universal$_1$ (resp. universal$_2$, smooth). We note that the ϵ-universal$_2$ property implies the ϵ-universal$_1$ property, by summing up the inequalities in the definition of the universal$_2$ property over all $\pi^* \in \Pi$. We also show some relations between the smoothness property and the universal$_1$ property.

Lemma 2.1. *If a projective hash family* $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ *is* 0*-smooth, then it is* $(1/|\Pi|)$*-universal*$_1$.

Proof. Since $\mathbf{H}$ is 0-smooth, the two distributions appearing in the definition of the smoothness for $\mathbf{H}$ are identical. Therefore, for any $x \in X \setminus L$, $s \in S$ and $\pi \in \Pi$, we have

$$\Pr_{k \xleftarrow{\$} K}[(\alpha(k), H_k(x)) = (s, \pi)] = \Pr_{k \xleftarrow{\$} K\,,\, \pi^\dagger \xleftarrow{\$} \Pi}[(\alpha(k), \pi^\dagger) = (s, \pi)] = \frac{1}{|\Pi|} \cdot \Pr_{k \xleftarrow{\$} K}[\alpha(k) = s] \enspace .$$

This implies that $\mathbf{H}$ is $(1/|\Pi|)$-universal$_1$, as desired. □

Lemma 2.2. *If a projective hash family* $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ *is* ϵ*-universal*$_1$*, then it is* ϵ'*-smooth where* $\epsilon' = (\epsilon|\Pi| - 1)(|\Pi| - 1)/2$. *In particular, if a projective hash family* $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ *is* $(1/|\Pi|)$*-universal*$_1$*, then it is* 0*-smooth.*

Proof. First, we note that $\epsilon \geq 1/|\Pi|$ by the definition of the ϵ-universal$_1$ property. Since $\mathbf{H}$ is ϵ-universal$_1$, for any $x \in X \setminus L$, $s \in S$ and $\pi \in \Pi$, we have

$$\begin{aligned}\Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L}[(\alpha(k), x^\dagger, H_k(x^\dagger)) = (s, x, \pi)] &= \frac{1}{|X \setminus L|} \cdot \Pr_{k \xleftarrow{\$} K}[(\alpha(k), H_k(x)) = (s, \pi)] \\ &\leq \frac{\epsilon}{|X \setminus L|} \cdot \Pr_{k \xleftarrow{\$} K}[\alpha(k) = s] = \epsilon \cdot \Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L}[(\alpha(k), x^\dagger) = (s, x)] \enspace .\end{aligned}$$

Since the right-hand side is independent of π, by summing up the inequality over all $\pi \in \Pi$ except a fixed $\pi' \in \Pi$, we have

$$\Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L}[(\alpha(k), x^\dagger) = (s, x) \wedge H_k(x^\dagger) \neq \pi'] \leq (|\Pi| - 1)\epsilon \cdot \Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L}[(\alpha(k), x^\dagger) = (s, x)] \enspace ,$$

therefore

$$\begin{aligned}&\Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L}[(\alpha(k), x^\dagger) = (s, x) \wedge H_k(x^\dagger) = \pi'] \\ &\geq \Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L}[(\alpha(k), x^\dagger) = (s, x)] - (|\Pi| - 1)\epsilon \cdot \Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L}[(\alpha(k), x^\dagger) = (s, x)] \\ &= (1 - (|\Pi| - 1)\epsilon) \cdot \Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L}[(\alpha(k), x^\dagger) = (s, x)] \enspace .\end{aligned}$$

By combining this inequality (where π' is replaced with π) with the first inequality above, and by using the relation $\Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L}[(\alpha(k), x^\dagger) = (s, x)] = |\Pi| \cdot \Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L\,,\, \pi^\dagger \xleftarrow{\$} \Pi}[(\alpha(k), x^\dagger, \pi^\dagger) = (s, x, \pi)]$, we have

$$\begin{aligned}&(|\Pi| - (|\Pi| - 1)|\Pi|\epsilon) \cdot \Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L\,,\, \pi^\dagger \xleftarrow{\$} \Pi}[(\alpha(k), x^\dagger, \pi^\dagger) = (s, x, \pi)] \\ &\leq \Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L}[(\alpha(k), x^\dagger, H_k(x^\dagger)) = (s, x, \pi)] \leq \epsilon|\Pi| \cdot \Pr_{k \xleftarrow{\$} K\,,\, x^\dagger \xleftarrow{\$} X \setminus L\,,\, \pi^\dagger \xleftarrow{\$} \Pi}[(\alpha(k), x^\dagger, \pi^\dagger) = (s, x, \pi)] \enspace ,\end{aligned}$$

therefore (since $\epsilon \geq 1/|\Pi|$)

$$\begin{aligned}\Bigg| \Pr_{k \overset{\$}{\leftarrow} K \,,\, x^\dagger \overset{\$}{\leftarrow} X \setminus L} [(\alpha(k), x^\dagger, H_k(x^\dagger)) = (s, x, \pi)] - \Pr_{k \overset{\$}{\leftarrow} K \,,\, x^\dagger \overset{\$}{\leftarrow} X \setminus L \,,\, \pi^\dagger \overset{\$}{\leftarrow} \Pi} [(\alpha(k), x^\dagger, \pi^\dagger) = (s, x, \pi)] \Bigg| \\ \leq \max\{1 - |\Pi| + \epsilon|\Pi|(|\Pi| - 1), \epsilon|\Pi| - 1\} \cdot \Pr_{k \overset{\$}{\leftarrow} K \,,\, x^\dagger \overset{\$}{\leftarrow} X \setminus L \,,\, \pi^\dagger \overset{\$}{\leftarrow} \Pi} [(\alpha(k), x^\dagger, \pi^\dagger) = (s, x, \pi)] \ .\end{aligned}$$

By summing up the inequality over all $s \in S$, $x \in X \setminus L$ and $\pi \in \Pi$, and by dividing it by two, the statistical distance between the two distributions appearing in the definition of the smoothness for $\mathbf{H}$ is bounded by $\max\{1 - |\Pi| + \epsilon|\Pi|(|\Pi| - 1), \epsilon|\Pi| - 1\}/2$. Note that $1 - |\Pi| + \epsilon|\Pi|(|\Pi| - 1) = (\epsilon|\Pi| - 1)(|\Pi| - 1)$. Now if $|\Pi| = 1$, then we have $\epsilon = 1$ since $1/|\Pi| \leq \epsilon \leq 1$, therefore $(\epsilon|\Pi| - 1)(|\Pi| - 1) = \epsilon|\Pi| - 1 = 0$. On the other hand, if $|\Pi| \geq 2$, then we have $(\epsilon|\Pi| - 1)(|\Pi| - 1) \geq \epsilon|\Pi| - 1$, therefore $\max\{1 - |\Pi| + \epsilon|\Pi|(|\Pi| - 1), \epsilon|\Pi| - 1\}/2 = (\epsilon|\Pi| - 1)(|\Pi| - 1)/2$. Hence, the claim holds. □

Subset Membership Problems: A subset membership problem $\mathbf{M}$ specifies a collection of probabilistic distribution $\{I_\ell\}_{\ell \geq 0}$ (indexed by a security parameter ℓ) over instance descriptions. An instance description $\Lambda[X, X', L, W, R] \in [I_\ell]$ specifies a non-empty set X and its non-empty subsets $X', L \subset X$, a non-empty set W, and a binary relation R defined over $X \times W$ with the property that an $x \in X$ is in the subset L if and only if there exists a "witness" $\omega \in W$ such that $(x, w) \in R$. When $X' = X$, we may simply write $\Lambda[X, L, W, R]$ instead of $\Lambda[X, X', L, W, R]$. Moreover, if these objects are clear from the context, we will just write Λ to indicate an instance description.

We require that a subset membership problem $\mathbf{M}$ provides the following algorithms: (1) the instance sampling algorithm takes as input 1^ℓ, and returns $\Lambda[X, X', L, W, R] \in [I_\ell]$ chosen according to I_ℓ, and (2) the subset sampling algorithm takes as input 1^ℓ and an instance $\Lambda[X, X', L, W, R] \in [I_\ell]$, and returns $x \overset{\$}{\leftarrow} L$ and a witness $\omega \in W$ for x. We say that a subset membership problem $\mathbf{M} = \{I_\ell\}_{\ell \geq 0}$ is *hard relative to* X', if the following two distributions are computationally indistinguishable:

$$\{\Lambda \leftarrow I_\ell; x \overset{\$}{\leftarrow} L : (\Lambda, x)\} \text{ and } \{\Lambda \leftarrow I_\ell; x \overset{\$}{\leftarrow} X' \setminus L : (\Lambda, x)\} \ .$$

When $X' = X$, we simply say that $\mathbf{M}$ is hard.

Hash Proof System (HPS): Informally, a HPS is a special kind of (designated-verifier) non-interactive zero-knowledge proof system for a subset membership problem $\mathbf{M} = \{I_\ell\}_{\ell > 0}$. A HPS has, as its internal structure, a family of hash functions with the special projective property, and this projective hash family is associated with each instance of the subset membership problems. Although HPS does not treat for all NP languages, HPS leads to an efficient CCA2-secure PKE construction.

As in [13], we will occasionally introduce an arbitrary finite set E to extend the sets X, X' and L in an instance $\Lambda[X, X', L, W, R] \in [I_\ell]$ of $\mathbf{M}$ into $X \times E$, $X' \times E$ and $L \times E$. If E is not required (e.g., for a smooth HPS in our construction of KH-PKE), then we omit E from the following algorithms. A HPS $\mathbf{P} = (\mathsf{HPS.param}, \mathsf{HPS.priv}, \mathsf{HPS.pub})$, for $\mathbf{M}$ associates each instance $\Lambda = \Lambda[X, X', L, W, R]$ of $\mathbf{M}$ with a projective hash family $\mathbf{H} = (H, K, X \times E, X' \times E, L \times E, \Pi, S, \alpha)$, provides the following three efficient algorithms:

1. The index sampling algorithm $\mathsf{HPS.param}$ takes an instance Λ as input, and returns $k \in K$ and $s \in S$ such that $\alpha(k) = s$.

2. The private evaluation algorithm $\mathsf{HPS.priv}$ takes $\Lambda \in [I_\ell]$, $k \in K$ and $(x, e) \in X \times E$ as input, and returns $\pi = H_k(x, e) \in \Pi$.

3. The public evaluation algorithm $\mathsf{HPS.pub}$ takes $\Lambda \in [I_\ell]$, $s \in S$, $x \in L$, $e \in E$, and a witness ω for x as input, and returns $\pi = H_k(x, e) \in \Pi$.

We say that $\mathbf{P}$ is ϵ-universal$_1$ (resp. ϵ-universal$_2$, ϵ-smooth) if for all $\ell > 0$ and for all $\Lambda \in [I_\ell]$, $\mathbf{H}$ is an ϵ-universal$_1$ (resp. ϵ-universal$_2$, ϵ-smooth) projective hash family.

The following homomorphic property of hash proof systems is required in our proposed construction.

Definition 2.1 (Homomorphic Projective Hash Family)**.** *We say that a projective hash family* $\mathbf{H} = (H, K, X \times E, X' \times E, L \times E, \Pi, S, \alpha)$ *is* homomorphic*, if X, E and Π are abelian groups (written in additive form), L is a subgroup of X, and we have $H_k(x_1) + H_k(x_2) = H_k(x_1 + x_2)$ for any $k \in K$ and $x_1, x_2 \in X$. We also say that a hash proof system* $\mathbf{P}$ *is* homomorphic*, if the underlying projective hash family is homomorphic. (We note that X' is not required to be a subgroup of X.)*

Diverse Group System and Derived Projective Hash Family: Here, we recall the definition of diverse group systems introduced in [13], which were used to construct projective hash families. Let X, L, and Π be abelian groups, where L is a proper subgroup of X, and $\mathsf{Hom}(X, \Pi)$ be the group of all homomorphisms $\phi : X \to \Pi$. Let $\mathcal{H}$ be a subgroup of $\mathsf{Hom}(X, \Pi)$. Then $\mathbf{G} = (\mathcal{H}, X, L, \Pi)$ is called a *group system.* In addition, we say that $\mathbf{G}$ is *diverse* if for all $x \in X \setminus L$, there exists $\phi \in \mathcal{H}$ such that $\phi(L) = \langle 0 \rangle$, but $\phi(x) \neq 0$.

We recall the projective hash family $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ derived from a diverse group system $\mathbf{G}$ ([13, Definition 2]): The instance $\Lambda = \Lambda[X, L, W, R]$ of the underlying subset membership problem satisfies that $W = (\mathbb{Z}_{|L|})^d$ and $(x, (\omega_1, \dots, \omega_d)) \in R$ if and only if $x = \sum_{i=1}^{d} \omega_i g_i$, where $\{g_1, \dots, g_d\}$ is a fixed generating set of L. Let the elements of the subgroup $\mathcal{H}$ of $\mathsf{Hom}(X, \Pi)$ be indexed as $\mathcal{H} = \{H_k \mid k \in K\}$ for a set K. Set $S = \Pi^d$, and define $\alpha : K \to S$ by $\alpha(k) = (\phi(g_1), \dots, \phi(g_d))$, where $\phi = H_k$. Note that $\mathbf{H}$ is a homomorphic projective hash family because $H_k(x)$ for $x \in L$ is determined by $\alpha(k)$ such that $H_k(x) = \phi(\sum_{i=1}^{d} \omega_i g_i) = \sum_{i=1}^{d} \omega_i \phi(g_i)$. The following was shown by Cramer and Shoup [13, Theorem 2].

Lemma 2.3. *The projective hash family* $\mathbf{H}$ *derived from a diverse group system* $\mathbf{G}$ *as above is $1/\widetilde{p}$-universal$_1$, where $\widetilde{p}$ is the smallest prime dividing $|X/L|$.*

3 Definition of KH-PKE

In this section, we give the formal definitions of the syntax and the security requirements of KH-PKE.

3.1 Syntax of KH-PKE

Definition 3.1 (Syntax of KH-PKE for homomorphic operation $\odot$)**.** *Let $\mathcal{M}$ be a message space. We require that for all $M_1, M_2 \in \mathcal{M}$, it holds that $M_1 \odot M_2 \in \mathcal{M}$. A KH-PKE scheme $\mathcal{KH}\text{-}\mathcal{PKE}$ =* ($\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}$) *for homomorphic operation $\odot$ consists of the following four algorithms:*

KeyGen**:** *This algorithm takes a security parameter 1^ℓ ($\ell \in \mathbb{N}$) as input, and returns a public key pk, a decryption key sk_d, and a homomorphic operation key sk_h.*

Enc**:** *This algorithm takes pk, and a message $M \in \mathcal{M}$ as input, and returns a ciphertext C.*

Dec**:** *This algorithm takes sk_d and C as input, and returns M or $\perp$.*

Eval**:** *This algorithm takes sk_h, two ciphertexts C_1 and C_2 as input, and outputs a ciphertext C or $\perp$.*

Note that the above definition for the evaluation algorithm Eval does not say anything about the homomorphic property, and its functionality is defined as a correctness requirement below. Let pk be a public key generated by the KeyGen algorithm, and $\mathcal{C}_{pk,M}$ be the set of all ciphertexts of $M \in \mathcal{M}$ under the public key pk, i.e., $\mathcal{C}_{pk,M} = \{C | \exists r \in \{0,1\}^* \text{ s.t. } C = \mathsf{Enc}(pk, M; r)\}$.

Definition 3.2 (Correctness)**.** *A KH-PKE scheme for homomorphic operation $\odot$ is said to be* correct *if for all $(pk, sk_d, sk_h) \leftarrow \mathsf{KeyGen}(1^\ell)$, the following two conditions are satisfied: (1) For all $M \in \mathcal{M}$, and all $C \in \mathcal{C}_{pk,M}$, it holds that $\mathsf{Dec}(sk_d, C) = M$. (2) For all $M_1, M_2 \in \mathcal{M}$, all $C_1 \in \mathcal{C}_{pk,M_1}$, and all $C_2 \in \mathcal{C}_{pk,M_2}$, it holds that $\mathsf{Eval}(sk_h, C_1, C_2) \in \mathcal{C}_{pk,M_1 \odot M_2}$.*

We call the Eval algorithm *commutative* if an operation $\odot$ is commutative, the distribution of $\mathsf{Eval}(sk_h, C_1, C_2)$ and that of $\mathsf{Eval}(sk_h, C_2, C_1)$ are identical. In fact, our KH-PKE schemes proposed in the paper are all commutative; for example, the homomorphic property of the DDH-based instantiation given in later section corresponds to the group operation in a multiplicative cyclic group.

Next, we define the security notion for KH-PKE, which we call *indistinguishability of message under adaptive chosen ciphertext attacks* (KH-CCA).

Definition 3.3 (KH-CCA)**.** *A KH-PKE scheme is said to be KH-CCA secure if for any PPT adversary $\mathcal{A}$, the advantage*

$$
\begin{aligned}
Adv^{KH\text{-}CCA}_{\mathsf{KH\text{-}PKE},\mathcal{A}}(\ell) = \big| \Pr[&(pk, sk_d, sk_h) \leftarrow \mathsf{KeyGen}(1^\ell); \\
&(M_0^*, M_1^*, State) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{find}, pk);\ \beta \xleftarrow{\$} \{0,1\}; \\
&C^* \leftarrow \mathsf{Enc}(pk, M_\beta^*);\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{guess}, State, C^*);\ \beta = \beta'] - \frac{1}{2}\big|
\end{aligned}
$$

is negligible in ℓ, where $\mathcal{O}$ consists of the three oracles $\mathsf{Eval}(sk_h, \cdot, \cdot)$, RevHK, and $\mathsf{Dec}(sk_d, \cdot)$ defined as follows. Let $\mathcal{D}$ be a list which is set as $\mathcal{D} = \{C^*\}$ right after the challenge stage ($\mathcal{D}$ is set as $\emptyset$ in the find *stage).*

- *The evaluation oracle $\mathsf{Eval}(sk_h, \cdot, \cdot)$: If* RevHK *has already been queried before, then this oracle is not available. Otherwise, this oracle responds to a query (C_1, C_2) with the result of $C \leftarrow \mathsf{Eval}(sk_h, C_1, C_2)$. In addition, if $C \neq \bot$ and either $C_1 \in \mathcal{D}$ or $C_2 \in \mathcal{D}$, then the oracle updates the list by $\mathcal{D} \leftarrow \mathcal{D} \cup \{C\}$.*
- *The homomorphic key reveal oracle* RevHK*: Upon a request, this oracle responds with sk_h. (This oracle is available only once.)*
- *The decryption oracle $\mathsf{Dec}(sk_d, \cdot)$: This oracle is not available if $\mathcal{A}$ has queried to* RevHK and *$\mathcal{A}$ has obtained the challenge ciphertext C^*. Otherwise, this oracle responds to a query C with the result of* $\mathsf{Dec}(sk_d, \cdot)$ *if $C \notin \mathcal{D}$ or returns $\bot$ otherwise.*

Here, let us remark on the definition of KH-CCA security. Throughout this paper, an adversary who has sk_h is called an *insider*, whereas an adversary who does not have sk_h is called an *outsider*.

In case $\mathcal{A}$ does not query the RevHK oracle (i.e., $\mathcal{A}$ is an outsider), $\mathcal{A}$ is allowed to adaptively issue decryption queries and evaluation queries of any ciphertexts. In particular, in order to capture the malleability in the presence of the homomorphic operation, the Eval oracle allows the challenge ciphertext C^* as input. To avoid an unachievable security definition, the Dec oracle immediately answers $\bot$ for "unallowable ciphertexts" that are the results of a homomorphic operation for C^* and any ciphertext of an adversary's choice. Such unallowable ciphertexts are maintained by the list $\mathcal{D}$.

The situation that the Dec oracle does not answer for ciphertexts that are derived from the challenge ciphertext C^* might seem somewhat analogous to the definition of RCCA security [8]. However, there is a critical difference between KH-CCA and RCCA: In the RCCA security game, the Dec oracle does not answer if a ciphertext C satisfies $\mathsf{Dec}(sk_d, C) \in \{M_0^*, M_1^*\}$. That is, the functionality of the Dec oracle is restricted regardless of the adversary's strategy. On the other hand, in the KH-CCA security game, in case an adversary selects the strategy that it does not submit C^* to the Eval oracle, the restriction on the Dec oracle is exactly the same as the CCA2 security for ordinary PKE scheme, and it is one of the adversary's possible strategies whether it submits C^* to the Eval oracle, and thus the adversary has more flexibility than in the RCCA game.

If an outsider $\mathcal{A}$ becomes an insider *after* $\mathcal{A}$ obtains the challenge ciphertext C^*, then $\mathcal{A}$ is not allowed to issue a decryption query *after* obtaining sk_h via the RevHK oracle. In other words, $\mathcal{A}$ is allowed to issue a decryption query until right before obtaining sk_h, even if C^* is given to $\mathcal{A}$. This restriction is again to avoid a triviality. (If $\mathcal{A}$ obtains sk_h, $\mathcal{A}$ can freely perform homomorphic operations over the challenge ciphertexts, and we cannot meaningfully define the "unallowable set" of ciphertexts.)

Note that we can show that any KH-CCA secure PKE scheme satisfies CCA1 (thus CPA also) security against an adversary who is given (pk, sk_h) in the setup phase. Showing this implication is possible mainly due to the RevHK oracle that returns sk_h to an adversary, and the Dec oracle in the KH-CCA game. Here, we explain how the implication of KH-CCA security to CCA1 security is proved. Let $\mathcal{A}$ be a CCA1 adversary. Using $\mathcal{A}$ as a building block, we can construct a reduction algorithm $\mathcal{B}$ that attacks KH-CCA security, as follows: First, $\mathcal{B}$ is firstly given pk. Then $\mathcal{B}$ asks the RevHK oracle to obtain sk_h, and runs $\mathcal{A}$ with input (pk, sk_h). Wnen $\mathcal{A}$ sends a ciphertext C as a decryption query, $\mathcal{B}$ forwards C as $\mathcal{B}$'s decryption query. After $\mathcal{A}$ submits (M_0^*, M_1^*) as $\mathcal{A}$'s challenge, $\mathcal{B}$ submits (M_0^*, M_1^*) as $\mathcal{B}$'s challenge. Given the challenge ciphertext C^*, $\mathcal{B}$ runs $\mathcal{A}$ with input C^*. When $\mathcal{A}$ terminates with output a guess bit, $\mathcal{B}$ uses what $\mathcal{A}$ outputs as its guess for the challenge bit, and terminates. It is easy to see that $\mathcal{B}$ perfectly simulates the CCA1 game for $\mathcal{A}$. Therefore, $\mathcal{B}$'s KH-CCA advantage equals $\mathcal{A}$'s CCA1 advantage. This implies that if the scheme is KH-CCA secure, then the scheme is CCA1 secure as well.

4 Generic Construction of KH-PKE

In this section, we describe the proposed generic construction of KH-PKE scheme from projective hash families, and give the security proof. For the purpose, in Section 4.1, we introduce two "computationally secure" variants of the notion of universal$_2$ projective hash families. Then in Section 4.2, we give the description of the generic construction. Then in Section 4.3, we prove the security of the proposed construction. We note that all of the projective hash families used in our construction can be constructed from a diverse group system [13]. Therefore, our proposed construction is fairly generic.

4.1 Computationally Universal$_2$ Projective Hash Families

In our generic construction of KH-PKE, a "computationally secure" variant of the notion of universal$_2$ projective hash families is utilized. Here we describe a formalization of the notion, which we call (first-uniform or first-adaptive) computationally universal$_2$ property. We note that the computationally universal$_2$ property for projective hash families introduced by Hofheinz and Kiltz [24] implies the first-uniform computationally universal$_2$ property in our sense, therefore our definition of the notion here covers wider situations than that in the previous work.

First, we define the first-uniform version of the computationally universal$_2$ property as follows:

Definition 4.1 (First-Uniform Computationally Universal$_2$ Property). *Let $\mathbf{H} = (H, K, X, X', L, \Pi, S, \alpha)$ be a projective hash family. We say that $\mathbf{H}$ is* first-uniform computationally universal$_2$ relative to X', *if for any oracle PPT adversary $\mathcal{A}$ with oracle* Hash *defined below, the probability that $\mathcal{A}^{\mathsf{Hash}}$ wins the following game (called the advantage of $\mathcal{A}$ and denoted by $Adv_{\mathcal{A}}^{\mathsf{UComp.Univ}_2}(\ell)$) is negligible in the security parameter ℓ, where the game is as follows:*

- *First, the challenger generates $k \xleftarrow{\$} K$ and $x^* \xleftarrow{\$} X' \setminus L$, and computes $s = \alpha(k)$ and $\pi^* = H_k(x^*)$. Then the challenger sends x^*, s and π^* to the adversary $\mathcal{A}$.*
- *During the game, the adversary can make queries $\mathsf{Hash}(x)$ to the oracle Hash adaptively, where $x \in X$. The oracle returns $\perp$ if the input x satisfies $x \in X \setminus L$, and returns $H_k(x)$ if $x \in L$.*
- *Finally, the adversary outputs elements $x \in X$ and $\pi \in \Pi$. We define that adversary wins if and only if $x \notin L$, $x \neq x^*$ and $H_k(x) = \pi$.*

We may omit the term "relative to X'" above when $X' = X$. We also say that a hash proof system is first-uniform computationally universal$_2$, *if the underlying projective hash family is first-uniform computationally universal$_2$.*

The word "first-uniform" in the definition above means that, for the two inputs x^* and x for the projective hash H_k in the game, the first one x^* is generated uniformly at random and the adversary cannot choose the first input. Secondly, we define the first-adaptive version of the computationally universal$_2$ property as follows, where the adversary can choose the first input for the projective hash:

Definition 4.2 (First-Adaptive Computationally Universal$_2$ Property)**.** *Let* $\mathbf{H} = (H, K, X, X', L, \Pi, S, \alpha)$ *be a projective hash family. We say that* $\mathbf{H}$ *is* first-adaptive computationally universal$_2$*, if for any oracle PPT adversary* $\mathcal{A}$ *with oracle* Hash *defined below, the probability that* $\mathcal{A}^{\mathsf{Hash}}$ *wins the following game (called the advantage of* $\mathcal{A}$ *and denoted by* $Adv_{\mathcal{A}}^{\mathsf{AComp.Univ}_2}(\ell)$*) is negligible in the security parameter* ℓ*, where the game is as follows:*

- *First, the challenger generates* $k \xleftarrow{\$} K$ *and computes* $s = \alpha(k)$*. Then the challenger sends* s *to the adversary* $\mathcal{A}$*.*
- *During the game, the adversary can make queries* $\mathsf{Hash}(x)$ *to the oracle* Hash *adaptively, where* $x \in X$*. The oracle returns* $\perp$ *if the input* x *satisfies* $x \in X \setminus L$*, and returns* $H_k(x)$ *if* $x \in L$*.*
- *At any time in the game decided by the adversary, the adversary has to* submit *an element* $x^* \in X$ *to the challenger. Then the challenger returns* $\pi^* = H_k(x^*)$ *to the adversary, regardless of whether* $x^* \in L$ *or not.*
- *Finally, the adversary outputs elements* $x \in X$ *and* $\pi \in \Pi$*. We define that adversary wins if and only if* $x, x^* \notin L$*,* $x \neq x^*$ *and* $H_k(x) = \pi$*.*

We also say that a hash proof system is first-adaptive computationally universal$_2$*, if the underlying projective hash family is first-adaptive computationally universal*$_2$*.*

Here we show the implication relations among the two computationally universal$_2$ properties and the original universal$_2$ property. Namely, we have the followings.

Lemma 4.1. *If a projective hash family* $\mathbf{H}$ *is universal*$_2$*, then* $\mathbf{H}$ *is first-adaptive computationally universal*$_2$*.*

Proof. Suppose that $\mathbf{H}$ is ϵ-universal$_2$ for negligible ϵ. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a PPT adversary for the first-adaptive computationally universal$_2$ game for $\mathbf{H}$, where $\mathcal{A}_1$ denotes the first part of $\mathcal{A}$ that takes 1^ℓ and $s = \alpha(k)$ as input and outputs the submitted element $x^* \in X$ as well as the internal state st, and $\mathcal{A}_2$ denotes the second part of $\mathcal{A}$ that takes st and $\pi^* = H_k(x^*)$ as input and outputs the elements $x \in X$ and $\pi \in \Pi$. Namely, we have

$$\begin{aligned}&Adv_{\mathcal{A}}^{\mathsf{AComp.Univ}_2}(\ell)\\ &= \Pr_{k \xleftarrow{\$} K}\left[(x^*, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Hash}}(1^\ell, \alpha(k)); (x, \pi) \leftarrow \mathcal{A}_2^{\mathsf{Hash}}(\mathsf{st}, H_k(x^*))\colon x, x^* \notin L \wedge x \neq x^* \wedge H_k(x) = \pi\right] .\end{aligned}$$

This expression can be rewritten as

$$\begin{aligned}&Adv_{\mathcal{A}}^{\mathsf{AComp.Univ}_2}(\ell)\\ &= \sum_{s \in S} \sum_{\substack{x, x^* \in X \setminus L \\ x \neq x^*}} \sum_{\pi, \pi^* \in \Pi} \Pr_{k \xleftarrow{\$} K}[(x^{*\dagger}, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Hash}}(1^\ell, s); (x^\dagger, \pi^\dagger) \leftarrow \mathcal{A}_2^{\mathsf{Hash}}(\mathsf{st}, \pi^*) \\ &\qquad\qquad : x^{*\dagger} = x^* \wedge x^\dagger = x \wedge \pi^\dagger = \pi \wedge H_k(x^*) = \pi^* \wedge H_k(x) = \pi \wedge \alpha(k) = s] .\end{aligned}$$

Now note that each of the algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ does not use any information on the key k except the information on $s = \alpha(k)$, while the oracle Hash can be simulated (not efficiently, in general) without using k (by exhaustively searching elements of L and witnesses for elements of L). This implies that, the expression of the advantage of $\mathcal{A}$ is equal to

$$\begin{aligned}&Adv_{\mathcal{A}}^{\mathsf{AComp.Univ}_2}(\ell)\\ &= \sum_{s \in S} \sum_{\substack{x, x^* \in X \setminus L \\ x \neq x^*}} \sum_{\pi, \pi^* \in \Pi} \Big(\Pr[(x^{*\dagger}, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Hash}}(1^\ell, s); (x^\dagger, \pi^\dagger) \leftarrow \mathcal{A}_2^{\mathsf{Hash}}(\mathsf{st}, \pi^*)\colon x^{*\dagger} = x^* \wedge x^\dagger = x \wedge \pi^\dagger = \pi] \\ &\qquad\qquad \cdot \Pr_{k \xleftarrow{\$} K}[H_k(x^*) = \pi^* \wedge H_k(x) = \pi \wedge \alpha(k) = s]\Big) .\end{aligned}$$

Since $\mathbf{H}$ is ϵ-universal$_2$, it follows that

$$
\begin{aligned}
&Adv_{\mathcal{A}}^{\mathsf{AComp.Univ}_2}(\ell)\\
&\le \sum_{s\in S}\sum_{\substack{x,x^*\in X\setminus L\\ x\neq x^*}}\sum_{\pi,\pi^*\in\Pi}\Big(\Pr[(x^{*\dagger},\mathsf{st})\leftarrow \mathcal{A}_1^{\mathsf{Hash}}(1^\ell,s);(x^\dagger,\pi^\dagger)\leftarrow\mathcal{A}_2^{\mathsf{Hash}}(\mathsf{st},\pi^*)\colon x^{*\dagger}=x^*\wedge x^\dagger=x\wedge\pi^\dagger=\pi] \quad .\\
&\qquad\qquad\qquad\qquad \cdot\epsilon\cdot \Pr\nolimits_{k\xleftarrow{\$}K}[H_k(x^*)=\pi^*\wedge\alpha(k)=s]\Big)
\end{aligned}
$$

The right-hand side is equal to

$$
\begin{aligned}
&\epsilon\cdot\sum_{s\in S}\sum_{\substack{x,x^*\in X\setminus L\\ x\neq x^*}}\sum_{\pi,\pi^*\in\Pi}\Pr_{k\xleftarrow{\$}K}\Big[(x^{*\dagger},\mathsf{st})\leftarrow \mathcal{A}_1^{\mathsf{Hash}}(1^\ell,s);(x^\dagger,\pi^\dagger)\leftarrow\mathcal{A}_2^{\mathsf{Hash}}(\mathsf{st},\pi^*)\\
&\qquad\qquad\qquad\qquad\qquad\colon x^{*\dagger}=x^*\wedge x^\dagger=x\wedge\pi^\dagger=\pi\wedge H_k(x^*)=\pi^*\wedge\alpha(k)=s\Big]\\
&=\epsilon\cdot\Pr_{k\xleftarrow{\$}K}[(x^*,\mathsf{st})\leftarrow\mathcal{A}_1^{\mathsf{Hash}}(1^\ell,\alpha(k));(x,\pi)\leftarrow\mathcal{A}_2^{\mathsf{Hash}}(\mathsf{st},H_k(x^*))\colon x,x^*\notin L\wedge x\neq x^*]\\
&\le\epsilon\quad .
\end{aligned}
$$

Hence we have $Adv_{\mathcal{A}}^{\mathsf{AComp.Univ}_2}(\ell)\le\epsilon$ which is negligible, as desired. □

Lemma 4.2. *Suppose that $X'\setminus L$ is approximately samplable relative to X. If a projective hash family $\mathbf{H}$ is first-adaptive computationally universal$_2$, then $\mathbf{H}$ is first-uniform computationally universal$_2$ relative to X'.*

Proof. Let $\mathcal{A}$ be any PPT adversary for the first-uniform computationally universal$_2$ game for $\mathbf{H}$ relative to X'. We construct an adversary $\mathcal{A}^\dagger$ for the first-adaptive computationally universal$_2$ game for $\mathbf{H}$ as follows. Given input 1^ℓ and s for $\mathcal{A}^\dagger$, the algorithm $\mathcal{A}^\dagger$ first samples an element $x^*\in X$ which is negligibly close to the uniform distribution on $X'\setminus L$ (this can be efficiently done since $X'\setminus L$ is approximately samplable relative to X), submits x^* to the challenger in the first-adaptive computationally universal$_2$ game, and receives $\pi^*=H_k(x^*)$ by the challenger. Then $\mathcal{A}^\dagger$ executes $\mathcal{A}$ with input $(1^\ell,x^*,s,\pi^*)$, where $\mathcal{A}^\dagger$ simulates the oracle Hash_U in the first-uniform computationally universal$_2$ game in the following manner: For each query x' to Hash_U made dy $\mathcal{A}$, $\mathcal{A}^\dagger$ makes a query x' to Hash_A, receives its reply π' and then returns π' to $\mathcal{A}$ as the reply to the query. Finally, $\mathcal{A}^\dagger$ receives the output (x,π) by $\mathcal{A}$, and outputs (x,π). We note that the algorithm $\mathcal{A}^\dagger$ is PPT as well as $\mathcal{A}$.

To evaluate the advantage of $\mathcal{A}^\dagger$, we may assume without loss of generality that x^* is a uniformly random element of $X'\setminus L$, since the modification causes at most negligible change of the advantage of $\mathcal{A}^\dagger$. In the present case, the simulation by $\mathcal{A}^\dagger$ of the first-uniform computationally universal$_2$ game for $\mathcal{A}$ is perfect, and $\mathcal{A}^\dagger$ wins the game if and only if $\mathcal{A}$ wins. This implies that $Adv_{\mathcal{A}^\dagger}^{\mathsf{AComp.Univ}_2}(\ell)=Adv_{\mathcal{A}}^{\mathsf{UComp.Univ}_2}(\ell)$, therefore $\mathcal{A}^\dagger$ has non-negligible advantage whenever $\mathcal{A}$ has. Hence, the claim holds. □

4.2 The Generic Construction

First, we summarize the primitives used in the generic construction. Let $\mathbf{M}=\{I_\ell\}_{\ell\ge0}$ be a subset membership problem which specifies an instance description $\Lambda=\Lambda[X,X',L,W,R]\in[I_\ell]$. In our construction, we use the following three projective hash families $\mathbf{H}$, $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$, and the corresponding hash proof systems $\mathbf{P}$, $\widehat{\mathbf{P}}$ and $\widetilde{\mathbf{P}}$ associated to $\mathbf{M}$.

- $\mathbf{H}=(H,K,X,X',L,\Pi,S,\alpha)$ is a homomorphic projective hash family which is smooth relative to (X',Π'), and $\mathbf{P}=(\mathsf{HPS.param},\mathsf{HPS.priv},\mathsf{HPS.pub})$ is the corresponding hash proof system, associated to the subset membership problem $\mathbf{M}$. In particular, Π is an abelian group (written in additive form) and Π' is a subgroup of Π. Moreover, Π' is approximately samplable relative to Π.
- $\widehat{\mathbf{H}}=(\widehat{H},\widehat{K},X,X',L,\widehat{\Pi},\widehat{S},\widehat{\alpha})$ is a homomorphic universal$_1$ projective hash family, and $\widehat{\mathbf{P}}=(\widehat{\mathsf{HPS}}.\mathsf{param},\widehat{\mathsf{HPS}}.\mathsf{priv},\widehat{\mathsf{HPS}}.\mathsf{pub})$ is the corresponding hash proof system associated to $\mathbf{M}$.

- $\widetilde{\mathbf{H}} = (\widetilde{H}, \widetilde{K}, X \times \Pi \times \widehat{\Pi}, X' \times \Pi \times \widehat{\Pi}, L \times \Pi \times \widehat{\Pi}, \widetilde{\Pi}, \widetilde{S}, \widetilde{\alpha})$ is a computationally or information-theoretically universal$_2$ projective hash family (see below for the detail), and $\widetilde{\mathbf{P}} = (\widetilde{\mathsf{HPS}}.\mathsf{param}, \widetilde{\mathsf{HPS}}.\mathsf{priv}, \widetilde{\mathsf{HPS}}.\mathsf{pub})$ is the corresponding hash proof system.

We also introduce some additional assumptions on the objects above. For the purpose, we introduce an auxiliary terminology:

Definition 4.3. *Let $\Lambda = \Lambda[X, X', L, W, R]$ be an instance description for $\mathbf{M}$. We say that a positive integer is a* critical integer*, if it is not coprime to $|X|$ and is not a multiple of $o(\Lambda)$, where $o(\Lambda)$ denotes the least common multiplier of the orders of elements of X' in the quotient group X/L.*

Now we describe the additional assumptions mentioned above. Here we introduce three kinds of assumptions, which have the following trade-off relations: The requirement for the HPS $\widetilde{\mathbf{P}}$ is weakened in the direction Assumption I → Assumption A → Assumption U, while the other conditions is relaxed in the other direction Assumption U → Assumption A → Assumption I. The reason of considering the three incomparable assumptions is to cover several instantiations of the proposed generic construction under various settings discussed in later sections. Now the three assumptions are as follows:

Assumption I: $\widetilde{\mathbf{P}}$ is (information-theoretically) universal$_2$.

Assumption A: $\widetilde{\mathbf{P}}$ is first-adaptive computationally universal$_2$, and $X' \setminus L$ is approximately samplable relative to X.

Assumption U: $\widetilde{\mathbf{P}}$ is first-uniform computationally universal$_2$ relative to $X' \times \Pi \times \widehat{\Pi}$, $\widehat{\mathbf{P}}$ is smooth relative to $(X', \widehat{\Pi})$, $X' \setminus L$ is approximately samplable relative to X, and $\Pi' = \Pi$. Moreover, it is computationally hard to find a critical integer from a given instance Λ of $\mathbf{M}$ (see Definition 4.3 for the terminology); it can be efficiently checked whether a given integer is a critical integer or not; we have $x + y \in X'$ for any $x \in X' \setminus L$ and $y \in L$; and we have $a \cdot x \in X'$ for any $x \in X' \setminus L$ and any integer a coprime to $|X|$.

Using these building blocks, we construct a KH-PKE scheme as in Figure 1. Roughly, the homomorphic smooth projective hash family $\mathbf{H}$ is used to hide a plaintext in a ciphertext. Moreover the universal property of $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$ are used to detect the invalidity of ciphertexts, which leads to resistance against ciphertext modification. However, the latter property looks contradictory to the homomorphic property that inherently involves such modification. In order to manage to deal with these two properties consistently, we utilize the following "transitional universal" property of the pair of $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$:

- If an adversary does not have the secret key of $\widetilde{\mathbf{H}}$ (which is the homomorphic key), then the (computationally or information-theoretically) universal$_2$ property of $\widetilde{\mathbf{H}}$ can be used to reject invalid input ciphertexts for the decryption and the evaluation algorithms.

- On the other hand, if an adversary has obtained the secret key of $\widetilde{\mathbf{H}}$, then the evaluation algorithm can update the values of $\widehat{\mathbf{H}}$ and $\mathbf{H}$ by using the key for $\widetilde{\mathbf{H}}$ and the homomorphic property of $\widehat{\mathbf{H}}$, while the universal$_1$ property of $\widehat{\mathbf{H}}$ (instead of the universal$_2$ property of $\widetilde{\mathbf{H}}$ which is no longer available) can be still used to reject invalid input ciphertexts for the decryption algorithm.

One might think that in the construction, $\widetilde{\mathbf{H}}$ is redundant, and thus is not necessary. However, this is not true. Namely, if $\widetilde{\mathbf{H}}$ is removed, then the adversary can extract meaningful information from the Eval oracle by submitting invalid ciphertexts, and therefore, the resulting scheme becomes insecure. In other words, with the help of $\widetilde{\mathbf{H}}$, the Eval oracle can distinguish invalid ciphertexts from valid ones, and consequently, the above attack is prevented.

Now we state the main theorem of the paper.

Theorem 4.1. *Our construction above is KH-CCA-secure, if the subset membership problem $\mathbf{M}$ is hard relative to $X' \subset X$, the hash proof systems $\mathbf{P}$, $\widehat{\mathbf{P}}$ and $\widetilde{\mathbf{P}}$ are as above, and one of Assumption I, Assumption A and Assumption U above is satisfied.*

```
$\mathsf{KeyGen}(1^\ell)$ :
  Pick $\Lambda = \Lambda[X, X', L, W, R] \leftarrow [I_\ell]$
  $(k, s) \leftarrow \mathsf{HPS.param}(1^\ell, \Lambda)$
  $(\widehat{k}, \widehat{s}) \leftarrow \widehat{\mathsf{HPS}}.\mathsf{param}(1^\ell, \Lambda)$
  $(\widetilde{k}, \widetilde{s}) \leftarrow \widetilde{\mathsf{HPS}}.\mathsf{param}(1^\ell, \Lambda)$
  $pk \leftarrow (s, \widehat{s}, \widetilde{s})$
  $sk_d \leftarrow (k, \widehat{k}, \widetilde{k})$; $sk_h \leftarrow (\widetilde{k})$
  Return $(pk, sk_d, sk_h)$
```

```
$\mathsf{Dec}(sk_d, C)$ :
  Parse $sk_d$ as $(k, \widehat{k}, \widetilde{k})$
  Parse $C$ as $(x, e, \widehat{\pi}, \widetilde{\pi})$
  $\widehat{\pi}' \leftarrow \widehat{\mathsf{HPS}}.\mathsf{priv}(1^\ell, \Lambda, \widehat{k}, x)$
  $\widetilde{\pi}' \leftarrow \widetilde{\mathsf{HPS}}.\mathsf{priv}(1^\ell, \Lambda, \widetilde{k}, (x, e, \widehat{\pi}'))$
  If $\widehat{\pi} \neq \widehat{\pi}'$ or $\widetilde{\pi} \neq \widetilde{\pi}'$ then return $\perp$
  $\pi \leftarrow \mathsf{HPS.priv}(1^\ell, \Lambda, k, x)$
  Return $M \leftarrow e - \pi$
```

```
$\mathsf{Enc}(pk, M)$ (for $M \in \mathcal{M} := \Pi'$) :
  Choose $x \xleftarrow{\$} L$ and its witness $\omega \in W$
  $\pi \leftarrow \mathsf{HPS.pub}(1^\ell, \Lambda, s, x, \omega)$;   $e \leftarrow M + \pi$
  $\widehat{\pi} \leftarrow \widehat{\mathsf{HPS}}.\mathsf{pub}(1^\ell, \Lambda, \widehat{s}, x, \omega)$
  $\widetilde{\pi} \leftarrow \widetilde{\mathsf{HPS}}.\mathsf{pub}(1^\ell, \Lambda, \widetilde{s}, (x, e, \widehat{\pi}), \omega)$
  Return $C \leftarrow (x, e, \widehat{\pi}, \widetilde{\pi})$
```

```
$\mathsf{Eval}(sk_h, C_1, C_2)$ :
  Parse $C_b$ as $(x_b, e_b, \widehat{\pi}_b, \widetilde{\pi}_b)$ for $b = 1, 2$
  $\widetilde{\pi}'_b \leftarrow \widetilde{\mathsf{HPS}}.\mathsf{priv}(1^\ell, \Lambda, \widetilde{k}, (x_b, e_b, \widehat{\pi}_b))$ for $b = 1, 2$
  If $\widetilde{\pi}_1 \neq \widetilde{\pi}'_1$ or $\widetilde{\pi}_2 \neq \widetilde{\pi}'_2$ then return $\perp$
  Choose $x_0 \xleftarrow{\$} L$ and its witness $\omega_0 \in W$
  $e_0 \leftarrow \mathsf{HPS.pub}(1^\ell, \Lambda, s, x_0, \omega_0)$
  $\widehat{\pi}_0 \leftarrow \widehat{\mathsf{HPS}}.\mathsf{pub}(1^\ell, \Lambda, \widehat{s}, x_0, \omega_0)$
  $x \leftarrow x_0 + x_1 + x_2$;   $e \leftarrow e_0 + e_1 + e_2$
  $\widehat{\pi} \leftarrow \widehat{\pi}_0 + \widehat{\pi}_1 + \widehat{\pi}_2$
  $\widetilde{\pi} \leftarrow \widetilde{\mathsf{HPS}}.\mathsf{priv}(1^\ell, \Lambda, \widetilde{k}, (x, e, \widehat{\pi}))$
  Return $C \leftarrow (x, e, \widehat{\pi}, \widetilde{\pi})$
```

Figure 1: The proposed KH-PKE construction from HPS.

Since all of the projective hash families used in our construction can be constructed from a diverse group system, from the result of [23] (where CPA-secure homomorphic PKE (with cyclic-group ciphertext space) implies diverse group systems), the following corollary is given.

Corollary 4.1. *KH-CCA secure KH-PKE is implied by CPA-secure homomorphic PKE with cyclic-group ciphertext space.*

4.3 Security Proof

From now, we give a proof of Theorem 4.1. First, we show the correctness of the Eval algorithm. Suppose that Eval receives validly generated ciphertexts $C_1 = (x_1, e_1, \widehat{\pi}_1, \widetilde{\pi}_1)$ and $C_2 = (x_2, e_2, \widehat{\pi}_2, \widetilde{\pi}_2)$ of plaintexts M_1 and M_2, respectively. Then the algorithm first generates a triple $(x_0, e_0, \widehat{\pi}_0)$, which is identical to the first three components of a ciphertext of plaintext 0 generated by the encryption algorithm. By the homomorphic properties of $\mathbf{H}$ and $\widehat{\mathbf{H}}$, by putting $x = x_0 + x_1 + x_2$, $e = e_0 + e_1 + e_2$ and $\widehat{\pi} = \widehat{\pi}_0 + \widehat{\pi}_1 + \widehat{\pi}_2$, we have

$$e = (0 + H_k(x_0)) + (M_1 + H_k(x_1)) + (M_2 + H_k(x_2)) = (M_1 + M_2) + H_k(x) \ ,$$

$$\widehat{\pi} = \widehat{H}_{\widehat{k}}(x_0) + \widehat{H}_{\widehat{k}}(x_1) + \widehat{H}_{\widehat{k}}(x_2) = \widehat{H}_{\widehat{k}}(x) \ .$$

Therefore, $(x, e, \widehat{\pi})$ is identical to the first three components of a ciphertext of $M_1 + M_2$. This implies that the output $C = (x, e, \widehat{\pi}, \widetilde{\pi})$ of the evaluation algorithm is a valid ciphertext of $M_1 + M_2$, as desired.

Intuitively, the evaluation algorithm performs the homomorphic operation of C_1, C_2 and a random ciphertext of 0. The reason of introducing the last random factor is to realize the following property, which plays a key role in the security proof:

Lemma 4.3 (Source Ciphertext Hiding Property)**.** *Let* $(pk, sk_d, sk_h) \leftarrow \mathsf{KeyGen}(1^\ell)$*,* $M \in \mathcal{M}$*,* $C_1, C'_1 \leftarrow \mathsf{Enc}(pk, M)$*, and* C_2 *be an arbitrary ciphertext. Then the output distributions of* $\mathsf{Eval}(sk_h, C_1, C_2)$ *and of* $\mathsf{Eval}(sk_h, C'_1, C_2)$ *are identical.*

Proof. In the argument above, since x_0 is uniformly random on L and $x_1 \in L$, $x' := x_0 + x_1$ is uniformly random on L as well and is independent of x_1. Now we have $e_0 + e_1 = M_1 + H_k(x')$ and $\widehat{\pi}_0 + \widehat{\pi}_1 = \widehat{H}_{\widehat{k}}(x')$, therefore the distribution of $(x, e, \widehat{\pi})$ depends solely on M_1 and is independent of x_1. This implies that the output distribution of the evaluation algorithm depends solely on M_1 and is independent of the choice of the ciphertext C_1. Hence, the claim holds. □

Here we give an intuitive explanation of how the source ciphertext hiding property is used to prove the security. A very brief outline of the security proof is the following: First we replace the valid challenge ciphertext $C^* = (x^*, e^*, \widehat{\pi}^*, \widetilde{\pi}^*)$, $x^* \in L$, with an invalid one with $x^* \in X' \setminus L$ (owing to the hardness of the subset membership problem $\mathbf{M}$ relative to $X' \subset X$). Secondly, we replace the second component $e^* = M_\beta^* + H_k(x^*)$ (where M_β^* denotes the challenge plaintext with challenge bit β) with $\pi^\dagger + H_k(x^*)$ where $\pi^\dagger \in \Pi$ is statistically close to the uniformly random element of Π' (owing to the smoothness of $\mathbf{H}$ relative to (X', Π')). Then the resulting challenge ciphertext is not dependent on M_β^* any longer, therefore any adversary has negligible advantage, as desired. However, in the proof strategy, for the step where $x^* \in X' \setminus L$ and $e^* = M_\beta^* + H_k(x^*)$, *an adversary is allowed to make several evaluation queries with input ciphertexts which are pairs of the (invalid) challenge ciphertext and any distinct valid ciphertext generated by the adversary.* Then the adversary can obtain many invalid ciphertexts legally, which involve values of $\widetilde{\mathbf{H}}$ for a large number of inputs chosen from $X \setminus L$. In such a case, the universal$_2$ property of $\widetilde{\mathbf{H}}$ is no longer enough to prevent the adversary to make a decryption or an evaluation query with invalid input ciphertext(s) which has fourth component being consistent to the value of $\widetilde{\mathbf{H}}$. Now the reply to the query sent to the adversary depends not only on the public key $\alpha(k)$ for $\mathbf{H}$ but also on the secret key k, which prevents us to safely replace the choice $e^* = M_\beta^* + H_k(x^*)$ of e^* with $e^* = \pi^\dagger + H_k(x^*)$ by utilizing the smoothness of $\mathbf{H}$. In short, the problem here is that the replies to the evaluation queries may in general depend on the challenge ciphertext (which is switched from being valid to being invalid in the proof); this is the reason why, despite the similarity of the construction of our proposed KH-PKE scheme to the Cramer–Shoup PKE scheme, a straightforward extension of the original proof strategy for CCA security of the Cramer–Shoup scheme is not sufficient for the proof for KH-CCA security of our scheme.

Our new idea to resolve the above-mentioned problem specific to KH-CCA security is the following: We modify the KH-CCA security game in such a way that, when an evaluation query involves the challenge ciphertext as input, the challenger first generates a fresh ciphertext (which we call "source ciphertext" in the security proof) of the same plaintext as the challenge ciphertext, and then proceeds the remaining calculation of the query by using the source ciphertext instead of the challenge ciphertext. In the starting case of the proof where $x^* \in L$, the source ciphertext hiding property implies that the output of the evaluation query calculated from the source ciphertext is identical to that calculated from the challenge ciphertext, therefore the modification of the game does not affect the advantage of the adversary. On the other hand, after the modification of the game, when $x^* \in L$ is replaced with $x^* \in X' \setminus L$ as above, the challenge ciphertext becomes invalid but each source ciphertext is kept valid. This prevents the adversary to obtain additional invalid ciphertexts, involving values of $\widetilde{\mathbf{H}}$, by using the evaluation queries as above; this implies that the universal$_2$ property of $\widetilde{\mathbf{H}}$ is still sufficient to achieve the KH-CCA security. (In fact, we should also introduce the replacement of the challenge ciphertext with a source ciphertext, not only for the cases of evaluation queries involving the challenge ciphertext, but also for the cases that an evaluation query involves a ciphertext related to the challenge ciphertext; see the proof of Theorem 4.1 below for the detail.)

Based on the discussion above, we start the proof of our main theorem.

Proof of Theorem 4.1. Let $\mathcal{A}$ be a PPT adversary against the KH-CCA security of our construction. Our goal in the proof is to show that the advantage $Adv^{KH\text{-}CCA}_{\mathsf{KH\text{-}PKE},\mathcal{A}}(\ell)$ of $\mathcal{A}$ is negligible. First note that, since $\mathcal{A}$ is of polynomial time, there exists a polynomial $Q(\ell)$ with the property that the total number of decryption queries and evaluation queries made by $\mathcal{A}$ is not larger than $Q(\ell)$ for any security parameter ℓ.

We proceed the proof by using game-hopping from the original KH-CCA game to the ideal situation that the challenge bit β is not used during the game; the advantage of the adversary becomes zero in the latter case. The game-hopping below consists of three large parts; the preliminary part, the main part, and the concluding part. In the proof, we say that a ciphertext is **regular**, if its first component belongs to L; otherwise, we say that the ciphertext is **irregular**. Similar terminology is used for inputs for H_k, $\widehat{H}_{\widehat{k}}$ and $\widetilde{H}_{\widetilde{k}}$. Moreover, we say that a decryption query is **regular** (respectively, **irregular**), if its input ciphertext is regular (respectively, irregular). We also say that an evaluation query is **regular**, if at least one of the two input ciphertexts is either in the dictionary $\mathcal{D}$ or regular; otherwise, we say that the query is **irregular**. On the other hand, let the term **private information** on the secret key for a projective hash family mean any information on the key except the corresponding public key.

Preliminary part of the game-hopping: At the beginning of the game-hopping for the proof, we start with the KH-CCA game (Game pre-0). First we introduce the idea of replacing the challenge ciphertext involved in each evaluation query with a fresh ciphertext, as mentioned before the proof (Game pre-1). Then, in order to deal also with an extended situation where an output of the encryption algorithm Enc becomes irregular, we use the private evaluation algorithms for hash proof systems $\mathbf{P}$, $\widehat{\mathbf{P}}$ and $\widetilde{\mathbf{P}}$ instead of the public evaluation algorithms (Game pre-3). Moreover, for the case of Assumption U, we also introduce an additional technical step of the game-hopping (Game pre-2) between Game pre-1 and Game pre-3, which avoids a problem in later parts of the proof caused by certain evaluation queries related to critical integers by automatically rejecting such queries. We formalize the preliminary part of the game-hopping as follows. In the proof, let $T_\ell^{(i)}$ denote the event that Game i outputs 1.

Game pre-0: This game simulates the KH-CCA game with adversary $\mathcal{A}$. We give a notational remark: Let $C^* = (x^*, e^*, \widehat{\pi}^*, \widetilde{\pi}^*)$ denote the challenge ciphertext, which is generated by $C^* \leftarrow \mathsf{Enc}(pk, M_\beta^*)$ where (M_0^*, M_1^*) is the pair of challenge plaintexts and $\beta \in \{0,1\}$ is the challenge bit. We say that the challenger rejects a query, if the reply to the query is $\bot$. Then, the game outputs 1 if the guessing bit β' output by the adversary in the simulated KH-CCA game is equal to β, and outputs 0 otherwise.

We note that the complexity of the game is polynomial, and $|\Pr[T_\ell^{(\text{pre-0})}] - 1/2| = Adv^{KH\text{-}CCA}_{\mathsf{KH\text{-}PKE},\mathcal{A}}(\ell)$.

Game pre-1: In comparison to Game pre-0, in guess stage, we introduce another auxiliary dictionary $\mathcal{D}'$ and modify the rule for the challenger to reply to evaluation queries (C', C'') satisfying that at least one of C' and C'' is listed in the original dictionary $\mathcal{D}$ and the query is not rejected (i.e., RevHK has not been queried, and for any input ciphertext of the query that is not in $\mathcal{D}$, its fourth component is consistent with the value of $\widetilde{\mathbf{H}}$ calculated from the first three components; note that any ciphertext in $\mathcal{D}$ passes the test by the definition of the algorithm Eval). When $\mathcal{D} = (C_0, C_1, \dots, C_\kappa)$ where $C_0 = C^*$ and $C_1, \dots, C_\kappa$ were added to $\mathcal{D}$ in this order, $\mathcal{D}'$ is of the form $((D_1', D_1''), (D_2', D_2''), \dots, (D_\kappa', D_\kappa''))$ where each of D_i' and D_i'' is either a ciphertext with fourth component being consistent or an index in $\{0, 1, \dots, i-1\}$. (We note that $\mathcal{D}'$ is empty at the beginning of the guess stage with $\mathcal{D} = (C^*)$.) Intuitively, the content of $\mathcal{D}'$ means that C_i was the reply to the evaluation query (D_i', D_i'') where, if D_i' or D_i'' is an index j, then it is interpreted as C_j.

Now we describe the modified rule for the challenger to reply to $(\kappa+1)$-th evaluation queries (C', C'') as above, where $\mathcal{D} = (C_0, C_1, \dots, C_\kappa)$ and $\mathcal{D}' = ((D_1', D_1''), (D_2', D_2''), \dots, (D_\kappa', D_\kappa''))$. We call it the $(\kappa+1)$-th **refreshing process** in the sequel, and we also call the query (C', C'') the $(\kappa+1)$-th **refreshing query**. In the process, the challenger first generates auxiliary ciphertexts $\overline{C}_0{}^{(\kappa+1)} = \overline{C}^{*(\kappa+1)}, \overline{C}_1{}^{(\kappa+1)}, \dots, \overline{C}_\kappa{}^{(\kappa+1)}$ as follows:

- The challenger generates $\overline{C}^{*(\kappa+1)}$ by $\overline{C}^{*(\kappa+1)} \leftarrow \mathsf{Enc}(pk, M_\beta^*)$ instead of using C^* itself, which we call the **source ciphertext** for the refreshing process.
- For each $i = 1, 2, \dots, \kappa$, the challenger generates $\overline{C}_i{}^{(\kappa+1)}$ by using the algorithm Eval, where its first (respectively, second) input is D_i' (respectively, D_i'') if D_i' (respectively, D_i'') is a ciphertext (i.e., not an index), and it is $\overline{C}_j{}^{(\kappa+1)}$ if D_i' (respectively, D_i'') is an index $j \in \{0, 1, \dots, i-1\}$.

Secondly, the challenger sets $D_{\kappa+1}'$ to be C' if C' is not in the dictionary $\mathcal{D}$, and to be an index i if C' is in $\mathcal{D}$ and i is the smallest index satisfying $C' = C_i$. The challenger also determines $D_{\kappa+1}''$ similarly by using C'' instead of C'. Thirdly, the challenger generates $C_{\kappa+1}$ by using the algorithm Eval, where its first (respectively, second) input is $D_{\kappa+1}'$ (respectively, $D_{\kappa+1}''$) if $D_{\kappa+1}'$ (respectively, $D_{\kappa+1}''$) is a ciphertext, and it is $\overline{C}_i{}^{(\kappa+1)}$ if $D_{\kappa+1}'$ (respectively, $D_{\kappa+1}''$) is an index $i \in \{0, 1, \dots, \kappa\}$. Finally, the challenger adds $C_{\kappa+1}$ to $\mathcal{D}$, adds $(D_{\kappa+1}', D_{\kappa+1}'')$ to $\mathcal{D}'$ and gives $C_{\kappa+1}$ to the adversary as the reply to the evaluation query.

By the source ciphertext hiding property, the distributions of $\overline{C}_1{}^{(\kappa+1)}, \dots, \overline{C}_\kappa{}^{(\kappa+1)}$ are identical to those of $C_1, \dots, C_\kappa$. Therefore, by the source ciphertext hiding property again, the distribution of $C_{\kappa+1}$ in the modified rule is identical to that of $C_{\kappa+1}$ in the original rule. This implies that the distribution of the adversary's view is identical in the two cases, therefore we have $\Pr[T_\ell^{(\text{pre-1})}] = \Pr[T_\ell^{(\text{pre-0})}]$. We note that the (time and memory) complexity of this game is still polynomial, since the number of evaluation queries made by $\mathcal{A}$ is bounded by the polynomial $Q(\ell)$ and the complexity of each refreshing process is linear in κ.

Game pre-2: Before describing the game, we introduce some auxiliary definitions. First, for the dictionary $\mathcal{D} = (C_0 = C^*, C_1, \dots, C_\kappa)$ and a ciphertext C, we define $\iota_{\mathcal{D}}(C) = h$ if $C \in \mathcal{D}$ and h is the smallest index with $C = C_h$, and $\iota_{\mathcal{D}}(C) = \perp$ if $C \not\in \mathcal{D}$. Secondly, for an index $h \in \{0, 1, \dots, \kappa\}$, we define a positive integer $\lambda_{\mathcal{D}}(h)$ in the following manner: We set $\lambda_{\mathcal{D}}(0) = 1$, and for $h > 0$, if C_h was the reply to an evaluation query (C', C'') where either C' or C'' was in $\mathcal{D}$, then we set $\lambda_{\mathcal{D}}(h) = \lambda' + \lambda''$ where $\lambda' = \lambda_{\mathcal{D}}(\iota_{\mathcal{D}}(C'))$ if $C' \in \mathcal{D}$ and $\lambda' = 0$ if $C' \not\in \mathcal{D}$, and λ'' is similarly defined by using C'' instead of C'. Intuitively, the integer $\lambda_{\mathcal{D}}(h)$ indicates how many copies of the challenge ciphertext C^* were added in the calculation of the ciphertext C_h in $\mathcal{D}$.

Based on the definition, in the case of Assumption U, we modify Game pre-1 in such a way that any evaluation query (C', C'') satisfying the condition for a refreshing query and that $C' \in \mathcal{D}$, $C'' \in \mathcal{D}$ and $\lambda_{\mathcal{D}}(\iota_{\mathcal{D}}(C')) + \lambda_{\mathcal{D}}(\iota_{\mathcal{D}}(C''))$ is a critical integer is always rejected (we call such a query a **critical query**); in the sequel, we regard each critical query as being not a refreshing query, in other words, the term "refreshing query" does not involve a critical query. On the other hand, in the cases of Assumption I and Assumption A, we define the game to be the same as Game pre-1.

By the definition, we have $\Pr[T_\ell^{(\text{pre-2})}] = \Pr[T_\ell^{(\text{pre-1})}]$ in the cases of Assumption I and Assumption A, while in the case of Assumption U, the difference $|\Pr[T_\ell^{(\text{pre-2})}] - \Pr[T_\ell^{(\text{pre-1})}]|$ will be evaluated owing to the hardness of finding a critical integer included in the Assumption U; details will be discussed later. We note that the complexity of the game is polynomial (in the case of Assumption U, this is due to the efficiency of deciding whether a given integer is a critical integer or not).

Game pre-3: Recall that, in the algorithm Enc, values of $\mathbf{H}$, $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$ are computed by using the public evaluation algorithms of $\mathbf{P}$, $\widehat{\mathbf{P}}$ and $\widetilde{\mathbf{P}}$, respectively, and a witness of a chosen element of L. In the game, we modify Game pre-2 in such a way that the challenger executes all of the algorithms Enc by using the private evaluation algorithms of $\mathbf{P}$, $\widehat{\mathbf{P}}$ and $\widetilde{\mathbf{P}}$ and the secret keys, where witnesses of elements of L are not required any longer. To avoid confusion, we write the modified version of Enc by Enc' from now. Since the modification does not change the output distributions of the encryption algorithms, we have $\Pr[T_\ell^{(\text{pre-3})}] = \Pr[T_\ell^{(\text{pre-2})}]$. We note that the complexity of the game is still polynomial.

Main part of the game-hopping: In Game pre-3, the source ciphertexts in the refreshing queries have been made fresh and their first components have been made independent of the challenge ciphertext. Owing to this, now the replacement of the first component $x^* \in L$ of the challenge ciphertext with $x^* \in X' \setminus L$ performed in the following game-hopping does not affect the behaviors of the refreshing queries, as desired. However, as a trade-off, now not only the challenge ciphertext but also the source ciphertexts involve information on M_β^*, hence information on the challenge bit β. From now, we proceed the game-hopping to remove the information on β from the source ciphertexts one by one. The process is performed by the following sequence of Games 0, 1, ..., $Q(\ell)$, where Game 0 is identical to Game pre-3:

Game κ ($0 \leq \kappa \leq Q(\ell)$): In comparison to Game pre-3, the constructions of the source ciphertexts $\overline{C}^{*(\kappa')}$ in the first κ refreshing processes, $1 \leq \kappa' \leq \kappa$ (or, when there are only less than κ refreshing processes, the source ciphertexts in all those refreshing processes) are modified as follows: The second component $e^{*(\kappa')}$ of $\overline{C}^{*(\kappa')}$ is chosen as $e^{*(\kappa')} \leftarrow \pi^\dagger + \mathsf{HPS.priv}(1^\ell, \Lambda, k, x^{*(\kappa')})$, instead of $e^{*(\kappa')} \leftarrow M_\beta^* + \mathsf{HPS.priv}(1^\ell, \Lambda, k, x^{*(\kappa')})$ as in the algorithm Enc', where $x^{*(\kappa')}$ is the first component of $\overline{C}^{*(\kappa')}$ and $\pi^\dagger \in \Pi$ is chosen independently of β according to the probability distribution (specified by the assumption that Π' is approximately samplable relative to Π) which is negligibly close to the uniform distribution on Π'. We note that the complexity of the game is polynomial.

In order to evaluate the differences of probabilities $\Pr[T_\ell^{(\kappa)}]$ between these games later, we introduce the following subdivision of the game sequence that connects each Game $(\kappa - 1)$ to Game κ. The main strategy is as follows: We replace the first component $x^{*(\kappa)} \in L$ of the source ciphertext $\overline{C}^{*(\kappa)}$ with $x^{*(\kappa)} \in X' \setminus L$ owing to the hardness of the subset membership problem $\mathbf{M}$ (SubGame κ.1), and then the second component $e^{*(\kappa)}$

of $\overline{C}^{*(\kappa)}$ is chosen as $e^{*(\kappa)} \leftarrow \pi^{\dagger} + \mathsf{HPS.priv}(1^{\ell}, \Lambda, k, x^{*(\kappa)})$ instead of $e^{*(\kappa)} \leftarrow M_{\beta}^{*} + \mathsf{HPS.priv}(1^{\ell}, \Lambda, k, x^{*(\kappa)})$ owing to the smoothness of $\mathbf{H}$ (SubGame κ.4). In order to utilize the smoothness of $\mathbf{H}$, we should guarantee that the private information on the key k for $\mathbf{H}$ is not used at any other step. For the purpose, before utilizing the smoothness of $\mathbf{H}$, we modify the game in such a way that all queries involving irregular ciphertexts are automatically rejected, owing to the universal properties of $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$ (SubGame κ.2 and SubGame κ.3). Moreover, after the replacement of the choice of $e^{*(\kappa)}$ as above, we restore the modifications introduced in SubGame κ.1 to SubGame κ.3 to the original situation (SubGame κ.5 to SubGame κ.7) The precise description is as follows:

SubGame κ.1: In the game, we modify the construction in Game $(\kappa - 1)$ of the source ciphertext $\overline{C}^{*(\kappa)}$ in the following manner: For the source ciphertext $\overline{C}^{*(\kappa)} = (x^{*(\kappa)}, e^{*(\kappa)}, \widehat{\pi}^{*(\kappa)}, \widetilde{\pi}^{*(\kappa)})$ originally generated by $\overline{C}^{*(\kappa)} \leftarrow \mathsf{Enc}'(M_{\beta}^{*})$, the first component $x^{*(\kappa)}$ is chosen uniformly at random from $X' \setminus L$ instead of L. Moreover, we modify the game further in such a way that the uniformly random element $x^{*(\kappa)} \in X' \setminus L$ is given as a part of the input for the game, instead of being chosen by the challenger. (When there are only less than κ refreshing processes, we interpret the situation in such a way that an element $x^{*(\kappa)} \in X' \setminus L$ is given as a part of the input for the game but it is actually not used during the game.)

We note that now $x^{*(\kappa)}$ is a part of the input for the game, therefore the uniformly random choice of $x^{*(\kappa)} \in X' \setminus L$ (which may be inefficient in general) is performed at outside of the game. Hence, the complexity of the game is polynomial as well as Game $(\kappa - 1)$. Now we can bound the difference $|\Pr[T_{\ell}^{(\kappa.1)}] - \Pr[T_{\ell}^{(\kappa - 1)}]|$ owing to the hardness of the subset membership problem $\mathbf{M}$; we will give a detailed argument later.

SubGame κ.2: In the game, we modify SubGame κ.1 in the following manner: In algorithms Enc', Dec and Eval, the challenger computes the values of $\mathbf{H}$, $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$ for regular inputs by first finding a witness ω for the first component of the input (which is an element of L) by an exhaustive search and then using the public evaluation algorithms and the witness ω instead of the private evaluation algorithms. To avoid confusion, we write the resulting algorithms after the modification by Enc'', Dec'' and Eval'', respectively.

Now the projective property of the projective hash family implies that the computed value is not changed by the modification, therefore we have $\Pr[T_{\ell}^{(\kappa.2)}] = \Pr[T_{\ell}^{(\kappa.1)}]$. Here we emphasize that, despite the complexity of the challenger in the game is in general not polynomial, the complexity of the adversary $\mathcal{A}$ alone is still polynomial.

SubGame κ.3: In comparison to SubGame κ.2, in the game, we modify the rule to decide in which case the challenger rejects each decryption or evaluation query made by $\mathcal{A}$ in such a way that any irregular query is automatically rejected (while the rule for regular queries is the same as SubGame κ.2). From now, we refer to the new rule as the **enhanced rejection rule**, while we refer to the original rule as the **original rejection rule**.

In the game, the complexity of the adversary alone is still polynomial. The difference $\Pr[T_{\ell}^{(\kappa.3)}] - \Pr[T_{\ell}^{(\kappa.2)}]$ can be evaluated owing to the computationally or information-theoretically universal properties of $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$. In fact, in order to carefully analyze the behaviors of the games, we will introduce further subdivision of the game-hopping, SubSubGames κ.3.0 to κ.3.$Q(\ell)$ that connect SubGame κ.2 to SubGame κ.3, where the rejection rule for one query is replaced at each step of the subdivided game-hopping. Details will be described later.

SubGame κ.4: In the game, we modify the construction in SubGame κ.3 of the source ciphertext $\overline{C}^{*(\kappa)} = (x^{*(\kappa)}, e^{*(\kappa)}, \widehat{\pi}^{*(\kappa)}, \widetilde{\pi}^{*(\kappa)})$ in the κ-th refreshing process as follows: The second component $e^{*(\kappa)}$ is chosen as $e^{*(\kappa)} \leftarrow \pi^{\dagger} + \mathsf{HPS.priv}(1^{\ell}, \Lambda, k, x^{*(\kappa)})$, instead of $e^{*(\kappa)} \leftarrow M_{\beta}^{*} + \mathsf{HPS.priv}(1^{\ell}, \Lambda, k, x^{*(\kappa)})$ as in SubGame κ.3, where $\pi^{\dagger} \in \Pi$ is chosen independently of β according to the probability distribution (specified by the assumption that Π' is approximately samplable relative to Π) which is negligibly close to the uniform distribution on Π'. (When there are only less than κ refreshing processes, we define SubGame κ.4 to be identical to SubGame κ.3.)

We can show that, the private information on the key k for $\mathbf{H}$ is not used in SubGame κ.3 except the computation of $e^{*(\kappa)} = M_{\beta}^{*} + \mathsf{HPS.priv}(1^{\ell}, \Lambda, k, x^{*(\kappa)})$, and the smoothness of $\mathbf{H}$ relative to (X', Π') implies that the difference $|\Pr[T_\ell^{(\kappa.4)}] - \Pr[T_\ell^{(\kappa.3)}]|$ is negligible. A detailed argument will be given later. We note that, in the game, the complexity of the adversary alone is polynomial, but the complexity of the whole game is not polynomial in general.

SubGame κ.5: In the game, to cancel the modification performed by SubGame κ.3, we restore the enhanced rejection rules for decryption and evaluation queries in SubGame κ.4 to the original rejection rules.

In the same way as SubGame κ.3, the difference $\Pr[T_\ell^{(\kappa.5)}] - \Pr[T_\ell^{(\kappa.4)}]$ can be evaluated owing to the computationally or information-theoretically universal properties of $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$. We will introduce further subdivision of the game-hopping, SubSubGames κ.5.0 to $\kappa.5.Q(\ell)$ that connect SubGame κ.4 to SubGame κ.5, where the rejection rule for one query is restored at each step. Details will be described later. We note that, in the game, the complexity of the adversary alone is polynomial, but the complexity of the whole game is still not polynomial in general.

SubGame κ.6: In the game, to cancel the modification performed by SubGame κ.2, we restore the algorithms Enc'', Dec'' and Eval'' used in SubGame κ.5 by the challenger to the algorithms Enc', Dec and Eval, respectively.

In the same way as SubGame κ.2, the projective property of the projective hash family implies again that $\Pr[T_\ell^{(\kappa.6)}] = \Pr[T_\ell^{(\kappa.5)}]$. Now we note that the complexity of the whole game (not only of the adversary alone) becomes polynomial again, as well as SubGame κ.1.

SubGame κ.7: In the game, to cancel the modification performed by SubGame κ.1, we restore the choice of $x^{*(\kappa)}$ in SubGame κ.6 in such a way that it is chosen uniformly at random from L instead of $X' \setminus L$. Moreover, now $x^{*(\kappa)}$ is chosen by the challenger when the source ciphertext $\overline{C}^{*(\kappa)}$ is generated, instead of the way in SubGame κ.6 where $x^{*(\kappa)}$ is given as a part of the input for the game.

We note that the resulting game is the same as Game κ, as desired. Moreover, since the complexity of SubGame κ.6 is polynomial as mentioned above, in the same way as SubGame κ.1, we can bound the difference $|\Pr[T_\ell^{(\kappa.7)}] - \Pr[T_\ell^{(\kappa.6)}]|$ owing to the hardness of the subset membership problem $\mathbf{M}$. We will give a detailed argument later.

Concluding part of the game-hopping: In Game $Q(\ell)$ (the last game in the main part of the game-hopping above), the challenge ciphertext C^* involves information on the challenge bit β, while the source ciphertexts do not involve information on β any longer. Finally, we proceed the game-hopping to remove the remaining information on β from the challenge ciphertext, which makes the behavior of the game completely independent of β, hence makes the advantage of the adversary zero. The process is similar to SubGames κ.1 to κ.4 above to remove the information on β from the source ciphertext in the κ-th refreshing process. The precise description is as follows:

Game con-1: In the game, we modify the construction in Game $Q(\ell)$ of the challenge ciphertext $C^* = (x^*, e^*, \widehat{\pi}^*, \widetilde{\pi}^*)$, which is originally generated by $C^* \leftarrow \mathsf{Enc}'(M_\beta^*)$, in such a way that x^* is chosen uniformly at random from $X' \setminus L$ instead of L. Moreover, we modify the game further in such a way that the uniformly random element $x^* \in X' \setminus L$ is given as a part of the input for the game, instead of being chosen by the challenger.

In the same way as SubGame κ.1 above, the complexity of the game is polynomial, and we can bound the difference $|\Pr[T_\ell^{(\text{con-1})}] - \Pr[T_\ell^{(Q(\ell))}]|$ owing to the hardness of the subset membership problem $\mathbf{M}$; we will give a detailed argument later.

Game con-2: In the same way as SubGame κ.2 above, in the game, we modify Game con-1 by replacing the algorithms Enc', Dec and Eval used by the challenger with algorithms Enc'', Dec'' and Eval'', respectively.

Now the projective property of the projective hash family implies that $\Pr[T_\ell^{(\text{con-2})}] = \Pr[T_\ell^{(\text{con-1})}]$. We note that the complexity of the adversary alone is still polynomial in the game.

Game con-3: In the same way as SubGame κ.3 above, in the game, we modify Game con-2 by replacing the original rejection rules for decryption and evaluation queries with the enhanced rejection rules.

Now the difference $\Pr[T_\ell^{(\text{con-3})}] - \Pr[T_\ell^{(\text{con-2})}]$ can be evaluated owing to the computationally or information-theoretically universal properties of $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$. In fact, in the same way as SubGame κ.3, we will introduce

SubGames con-3.0 to con-3.$Q(\ell)$ that connect Game con-2 to Game con-3, where the rejection rule for one query is replaced at each step. Details will be described later.

Game con-4: Finally, in the game, we modify the construction in Game con-3 of the challenge ciphertext $C^* = (x^*, e^*, \widehat{\pi}^*, \widetilde{\pi}^*)$ in the following manner: The second component e^* is chosen as $e^* \leftarrow \pi^\dagger + \mathsf{HPS.priv}(1^\ell, \Lambda, k, x^*)$, instead of $e^* \leftarrow M_\beta^* + \mathsf{HPS.priv}(1^\ell, \Lambda, k, x^*)$ as in Game con-3, where $\pi^\dagger \in \Pi$ is chosen independently of β according to the probability distribution (specified by the assumption that Π' is approximately samplable relative to Π) which is negligibly close to the uniform distribution on Π'.

We can show that, in the same way as SubGame κ.4 above, the private information on the key k for $\mathbf{H}$ is not used in Game con-3 except the computation of $e^* = M_\beta^* + \mathsf{HPS.priv}(1^\ell, \Lambda, k, x^*)$, and the smoothness of $\mathbf{H}$ relative to (X', Π') implies that the difference $|\Pr[T_\ell^{(\text{con-4})}] - \Pr[T_\ell^{(\text{con-3})}]|$ is negligible. A detailed argument will be given later.

In Game con-4, the information on the challenge bit β is not used during the game, which implies that $\Pr[T_\ell^{(\text{con-4})}] = 1/2$. This is the goal of the game-hopping. Then we have

$$Adv_{\mathsf{KH\text{-}PKE},\mathcal{A}}^{KH\text{-}CCA}(\ell) = |\Pr[T_\ell^{(\text{pre-0})}] - 1/2| = |\Pr[T_\ell^{(\text{pre-0})}] - \Pr[T_\ell^{(\text{con-4})}]| \ .$$

Now, since we have mentioned that the differences for some steps of the game-hopping are zero, we have

$$\begin{aligned}
&\Pr[T_\ell^{(\text{pre-0})}] - \Pr[T_\ell^{(\text{con-4})}] \\
&= \big(\Pr[T_\ell^{(\text{pre-1})}] - \Pr[T_\ell^{(\text{pre-2})}]\big) + \sum_{\kappa=1}^{Q(\ell)} \Big(\big(\Pr[T_\ell^{(\kappa-1)}] - \Pr[T_\ell^{(\kappa.1)}]\big) + \big(\Pr[T_\ell^{(\kappa.2)}] - \Pr[T_\ell^{(\kappa.3)}]\big) \\
&\qquad\qquad + \big(\Pr[T_\ell^{(\kappa.3)}] - \Pr[T_\ell^{(\kappa.4)}]\big) + \big(\Pr[T_\ell^{(\kappa.4)}] - \Pr[T_\ell^{(\kappa.5)}]\big) + \big(\Pr[T_\ell^{(\kappa.6)}] - \Pr[T_\ell^{(\kappa)}]\big) \Big) \\
&\qquad + \big(\Pr[T_\ell^{(Q(\ell))}] - \Pr[T_\ell^{(\text{con-1})}]\big) + \big(\Pr[T_\ell^{(\text{con-2})}] - \Pr[T_\ell^{(\text{con-3})}]\big) + \big(\Pr[T_\ell^{(\text{con-3})}] - \Pr[T_\ell^{(\text{con-4})}]\big) \\
&= \delta_1 + \delta_2 + \delta_3 + \delta_4 \ ,
\end{aligned}$$

where

$$\delta_1 = \Pr[T_\ell^{(\text{pre-1})}] - \Pr[T_\ell^{(\text{pre-2})}] \ ,$$

$$\delta_2 = \sum_{\kappa=1}^{Q(\ell)} \Big(\big(\Pr[T_\ell^{(\kappa-1)}] - \Pr[T_\ell^{(\kappa.1)}]\big) + \big(\Pr[T_\ell^{(\kappa.6)}] - \Pr[T_\ell^{(\kappa)}]\big) \Big) + \big(\Pr[T_\ell^{(Q(\ell))}] - \Pr[T_\ell^{(\text{con-1})}]\big) \ ,$$

$$\delta_3 = \sum_{\kappa=1}^{Q(\ell)} \Big(\big(\Pr[T_\ell^{(\kappa.2)}] - \Pr[T_\ell^{(\kappa.3)}]\big) + \big(\Pr[T_\ell^{(\kappa.4)}] - \Pr[T_\ell^{(\kappa.5)}]\big) \Big) + \big(\Pr[T_\ell^{(\text{con-2})}] - \Pr[T_\ell^{(\text{con-3})}]\big) \ ,$$

$$\delta_4 = \sum_{\kappa=1}^{Q(\ell)} \big(\Pr[T_\ell^{(\kappa.3)}] - \Pr[T_\ell^{(\kappa.4)}]\big) + \big(\Pr[T_\ell^{(\text{con-3})}] - \Pr[T_\ell^{(\text{con-4})}]\big) \ .$$

From now, we evaluate the quantities δ_1, δ_2, δ_3 and δ_4 above.

Evaluation of δ_1: We note that $\delta_1 = 0$ for the cases of Assumption I and Assumption A. On the other hand, for the case of Assumption U, we divide the game-hopping from Game pre-1 to pre-2 by introducing the following subdivision:

SubGame pre-2.κ ($0 \le \kappa \le Q(\ell)$): In the game, we modify Game pre-1 in such a way that, in the first κ evaluation queries (C', C'') (or, when the number of evaluation queries is less than κ, in all of the evaluation queries), the challenger rejects the query if the query satisfies the condition for a refreshing query and it is a critical query. Note that SubGame pre-2.0 and SubGame pre-2.$Q(\ell)$ are the same as Game pre-1 and Game pre-2, respectively. Note also that the the complexity of the game is polynomial, by the efficiency of deciding whether a given integer is a critical integer or not (see Assumption U).

For each $1 \leq \kappa \leq Q(\ell)$, let $R_\ell^{(\text{pre-2.}\kappa)}$ denote the event that, in SubGame pre-2.κ, the κ-th evaluation query (exists and) is a critical query. Since SubGame pre-2.$(\kappa-1)$ and SubGame pre-2.κ are identical unless the κ-th evaluation query it is a critical query, we have $|\Pr[T_\ell^{(\text{pre-2.}(\kappa\,-\,1))}] - \Pr[T_\ell^{(\text{pre-2.}\kappa)}]| \leq \Pr[R_\ell^{(\text{pre-2.}\kappa)}]$, therefore $|\delta_1| = |\Pr[T_\ell^{(\text{pre-1})}] - \Pr[T_\ell^{(\text{pre-2})}]| \leq \sum_{\kappa=1}^{Q(\ell)} \Pr[R_\ell^{(\text{pre-2.}\kappa)}]$ by the triangle inequality. Now, to evaluate the right-hand side, we introduce the following auxiliary adversary $\mathcal{A}_1$ finding a critical integer:

1. Given input 1^ℓ and Λ for $\mathcal{A}_1$, first $\mathcal{A}_1$ chooses a game uniformly at random from the $Q(\ell)$ Games pre-2.1 to pre-2.$Q(\ell)$.

2. When Game pre-2.κ, $1 \leq \kappa \leq Q(\ell)$, is chosen, $\mathcal{A}_1$ simulates Game pre-2.κ with input $(1^\ell, \Lambda)$ for the game. In the case that the κ-th evaluation query (C', C'') is critical, $\mathcal{A}_1$ outputs $\lambda_{\mathcal{D}}(\iota_{\mathcal{D}}(C')) + \lambda_{\mathcal{D}}(\iota_{\mathcal{D}}(C''))$ which is a critical integer. Otherwise, $\mathcal{A}_1$ outputs 1.

By the definition, $\mathcal{A}_1$ is PPT, and the advantage $Adv_{\mathcal{A}_1}$ of $\mathcal{A}_1$ (that is, the probability that $\mathcal{A}_1$ output a critical integer) satisfies that

$$Adv_{\mathcal{A}_1} = \frac{1}{Q(\ell)} \sum_{\kappa=1}^{Q(\ell)} \Pr[R_\ell^{(\text{pre-2.}\kappa)}]$$

(we note that 1 is never a critical integer by the definition). This implies that $|\delta_1| \leq Q(\ell) Adv_{\mathcal{A}_1}$, while $Adv_{\mathcal{A}_1}$ is negligible since finding a critical integer is hard by Assumption U. Hence, $|\delta_1|$ is negligible as well.

Evaluation of δ_2: In order to evaluate the quantity δ_2, we introduce the following auxiliary distinguisher $\mathcal{A}_2$ for the subset membership problem $\mathbf{M}$ relative to X':

1. Given input 1^ℓ and $(\Lambda, x^\dagger)$ for $\mathcal{A}_2$, where we have either $x^\dagger \in L$ or $x^\dagger \in X' \setminus L$, first $\mathcal{A}_2$ chooses a pair of games uniformly at random from the $2(\ell) + 1$ pairs (Game $(\kappa - 1)$, SubGame κ.1) with $1 \leq \kappa \leq Q(\ell)$, (SubGame κ.6, Game κ) with $1 \leq \kappa \leq Q(\ell)$, and (Game $Q(\ell)$, Game con-1).

2. In the case that the pair (Game $(\kappa - 1)$, SubGame κ.1) was chosen, $\mathcal{A}_2$ executes SubGame κ.1 with input 1^ℓ, Λ and $x^{*(\kappa)} = x^\dagger$. (By the definition of the games, the behavior of the game inside $\mathcal{A}_2$ becomes identical to Game $(\kappa - 1)$ and SubGame κ.1 if $x^\dagger \in L$ and $x^\dagger \in X' \setminus L$, respectively.) Then $\mathcal{A}_2$ outputs the output bit of the game.

3. In the case that the pair (SubGame κ.6, Game κ) was chosen, $\mathcal{A}_2$ executes SubGame κ.6 with input 1^ℓ, Λ and $x^{*(\kappa)} = x^\dagger$. (By the definition of the games, the behavior of the game inside $\mathcal{A}_2$ becomes identical to Game κ and SubGame κ.6 if $x^\dagger \in L$ and $x^\dagger \in X' \setminus L$, respectively.) Then $\mathcal{A}_2$ outputs the opposite bit to the output bit of the game; that is, $\mathcal{A}_2$ outputs $1 - b$ if the game inside $\mathcal{A}_2$ outputs b.

4. In the case that the pair (Game $Q(\ell)$, Game con-1) was chosen, $\mathcal{A}_2$ executes Game con-1 with input 1^ℓ, Λ and $x^* = x^\dagger$. (By the definition of the games, the behavior of the game inside $\mathcal{A}_2$ becomes identical to Game $Q(\ell)$ and Game con-1 if $x^\dagger \in L$ and $x^\dagger \in X' \setminus L$, respectively.) Then $\mathcal{A}_2$ outputs the output bit of the game.

We note that $\mathcal{A}_2$ is PPT. By the definition of $\mathcal{A}_2$, we have

$$\Pr[1 \leftarrow \mathcal{A}_2 \mid x^\dagger \in L] = \frac{1}{2Q(\ell)+1} \left(\sum_{\kappa=1}^{Q(\ell)} \bigl(\Pr[T_\ell^{(\kappa-1)}] + (1 - \Pr[T_\ell^{(\kappa)}]) \bigr) + \Pr[T_\ell^{(Q(\ell))}] \right)$$

and

$$\Pr[1 \leftarrow \mathcal{A}_2 \mid x^\dagger \in X' \setminus L] = \frac{1}{2Q(\ell)+1} \left(\sum_{\kappa=1}^{Q(\ell)} \bigl(\Pr[T_\ell^{(\kappa.1)}] + (1 - \Pr[T_\ell^{(\kappa.6)}]) \bigr) + \Pr[T_\ell^{(\text{con-1})}] \right) ,$$

therefore

$$\begin{aligned}
&\Pr[1 \leftarrow \mathcal{A}_2 \mid x^\dagger \in L] - \Pr[1 \leftarrow \mathcal{A}_2 \mid x^\dagger \in X' \setminus L] \\
&= \frac{1}{2Q(\ell)+1} \left(\sum_{\kappa=1}^{Q(\ell)} \big(\Pr[T_\ell^{(\kappa-1)}] - \Pr[T_\ell^{(\kappa)}] - \Pr[T_\ell^{(\kappa.1)}] + \Pr[T_\ell^{(\kappa.6)}]\big) + \Pr[T_\ell^{(Q(\ell))}] - \Pr[T_\ell^{(\text{con-1})}] \right) \\
&= \frac{\delta_2}{2Q(\ell)+1} \enspace .
\end{aligned}$$

Hence we have

$$|\delta_2| = (2Q(\ell)+1) Adv_{\mathbf{M},\mathcal{A}_2}(\ell) \enspace ,$$

where $Adv_{\mathbf{M},\mathcal{A}_2}(\ell) = |\Pr[1 \leftarrow \mathcal{A}_2 \mid x^\dagger \in L] - \Pr[1 \leftarrow \mathcal{A}_2 \mid x^\dagger \in X' \setminus L]|$ denotes the advantage of $\mathcal{A}_2$ for the subset membership problem $\mathbf{M}$ relative to X'. Moreover, by the assumption that $\mathbf{M}$ is hard relative to X', $Adv_{\mathbf{M},\mathcal{A}_2}(\ell)$ is negligible. Hence, $|\delta_2|$ is negligible as well.

Evaluation of δ_3: From now, we evaluate the quantity δ_3. For the purpose, first we divide the game-hopping from SubGame $\kappa.2$ to $\kappa.3$ for each $1 \leq \kappa \leq Q(\ell)$ by introducing the following subdivision:

SubSubGame κ.3.ρ ($0 \leq \rho \leq Q(\ell)$): In comparison to SubGame $\kappa.2$, in the game, we replace the original rejection rules for the first ρ decryption or evaluation queries (or, when the total number of decryption queries and evaluation queries is less than ρ, the original rejection rules for all of the decryption and evaluation queries) with the enhanced rejection rules. Note that SubSubGame $\kappa.3.0$ and SubSubGame $\kappa.3.Q(\ell)$ are the same as SubGame $\kappa.2$ and SubGame $\kappa.3$, respectively.

Let $R_\ell^{(\kappa.3.\rho)}$ denote the event in SubSubGame $\kappa.3.\rho$ that the ρ-th query (exists and) is rejected by the enhanced rejection rule but is not rejected by the original rejection rule. Then we have $|\Pr[T_\ell^{(\kappa.3.(\rho-1))}] - \Pr[T_\ell^{(\kappa.3.\rho)}]| \leq \Pr[R_\ell^{(\kappa.3.\rho)}]$, therefore $|\Pr[T_\ell^{(\kappa.2)}] - \Pr[T_\ell^{(\kappa.3)}]| \leq \sum_{\rho=1}^{Q(\ell)} \Pr[R_\ell^{(\kappa.3.\rho)}]$ by the triangle inequality. Now, to evaluate the right-hand side, we prove the following properties.

Claim 1. In SubSubGame $\kappa.3.\rho$, the replies to the first ρ queries are independent of the private information on the keys for $\mathbf{H}$, $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$, except the reply to the κ-th refreshing query which depends (if the κ-th refreshing query is ρ-th or earlier query) on one value of the private evaluation algorithm for each of these projective hash families. Moreover, the replies to the evaluation queries (which are not rejected) among the first ρ queries and the ciphertexts in $\mathcal{D}$ at the end of the ρ-th query are regular ciphertexts, except (if the κ-th refreshing query is ρ-th or earlier query) the ciphertext added to $\mathcal{D}$ at the κ-th refreshing process that is the reply to the κ-th refreshing query.

More precisely, when the κ-th refreshing query is the ρ-th or earlier query, let $\mathcal{D} = (C_0 = C^*, C_1, \dots, C_\kappa)$ and $\mathcal{D}' = ((D'_1, D''_1), \dots, (D'_\kappa, D''_\kappa))$ be the two dictionaries after the κ-th refreshing query, let $\overline{C}_0{}^{(\kappa)}, \overline{C}_1{}^{(\kappa)}, \dots, \overline{C}_\kappa{}^{(\kappa)}$ be the ciphertexts calculated in the κ-th refreshing process (hence $\overline{C}_0{}^{(\kappa)}$ is the source ciphertext $\overline{C^*}^{(\kappa)} = (x^{*(\kappa)}, e^{*(\kappa)}, \widehat{\pi}^{*(\kappa)}, \widetilde{\pi}^{*(\kappa)})$ and $\overline{C}_\kappa{}^{(\kappa)} = C_\kappa$), and put $\overline{C}_{\kappa'}{}^{(\kappa)} = (x_{\kappa'}, e_{\kappa'}, \widehat{\pi}_{\kappa'}, \widetilde{\pi}_{\kappa'})$ for each $\kappa' = 0, 1, \dots, \kappa$. Then for each κ', we have:

- $x_{\kappa'}$ is the sum of $\lambda_{\mathcal{D}}(\kappa') \cdot x^{*(\kappa)}$, an integer linear combination of elements of L independent of $x^{*(\kappa)}$, and an integer linear combination of the first components of ciphertexts D'_i and D''_i (i.e., those D'_i and D''_i are not indices) listed in $\mathcal{D}'$ with $1 \leq i \leq \kappa'$. Hence, $x_{\kappa'} - \lambda_{\mathcal{D}}(\kappa') \cdot x^{*(\kappa)}$ is an element of L independent of $x^{*(\kappa)}$.

- $e_{\kappa'}$ is the sum of $\lambda_{\mathcal{D}}(\kappa') \cdot e^{*(\kappa)}$, an integer linear combination of elements of the form $H_k(\overline{x})$ with $\overline{x} \in L$ being independent of $x^{*(\kappa)}$, and an integer linear combination of the second components of ciphertexts D'_i and D''_i listed in $\mathcal{D}'$ with $1 \leq i \leq \kappa'$.

- For any ciphertext $C = (x, e, \widehat{\pi}, \widetilde{\pi})$, we define $\widehat{\Delta}(C) = \widehat{\pi} - \widehat{H}_{\widehat{k}}(x)$. Then, $\widehat{\Delta}(\overline{C}_{\kappa'}{}^{(\kappa)})$ is an integer linear combination of $\widehat{\Delta}(D'_i)$ and $\widehat{\Delta}(D''_i)$ for ciphertexts D'_i and D''_i listed in $\mathcal{D}'$ with $1 \leq i \leq \kappa'$. Moreover, the calculation of these $\widehat{\Delta}(D'_i)$ and $\widehat{\Delta}(D''_i)$ from given the D'_i and D''_i is independent of the private information on the key $\widehat{k}$ for $\widehat{\mathbf{H}}$.

- We have $\widetilde{\pi}_{\kappa'} = \widetilde{H}_{\widetilde{k}}(x_{\kappa'}, e_{\kappa'}, \widehat{\pi}_{\kappa'})$.

Proof of Claim 1. We proceed the proof by induction on ρ, where the starting case $\rho = 0$ is trivial. We suppose that $\rho > 0$, and we focus on the ρ-th query. First, for the case that the query is a decryption query $C = (x, e, \widehat{\pi}, \widetilde{\pi})$, by the enhanced rejection rule, the query is always rejected if $x \in X \setminus L$. On the other hand, if $x \in L$, then for replying to the query, the challenger may compute the values of projective hashes for x and $(x, e, \widehat{\pi})$ in the algorithm Dec'', but it is done by using the public evaluation algorithms (rather than the private ones) since $x \in L$. Hence, the reply to the query is indeed independent of the private information on the keys.

Secondly, we consider the case that the query is an evaluation query (C', C'') with $C' = (x', e', \widehat{\pi}', \widetilde{\pi}')$ and $C'' = (x'', e'', \widehat{\pi}'', \widetilde{\pi}'')$. In the enhanced rejection rule, the query is always rejected if it is irregular. From now, we suppose that the query is regular. Now if $C' \not\in \mathcal{D}$ and $C'' \not\in \mathcal{D}$, then we have $x', x'' \in L$. In this case, for replying to the query, the challenger may compute the values of projective hashes for $(x', e', \widehat{\pi}')$ and $(x'', e'', \widehat{\pi}'')$ in the algorithm Eval'', but it is done by using the public evaluation algorithms (rather than the private ones) since $x', x'' \in L$. Hence, the reply to the query is indeed independent of the private information on the keys in this case. Moreover, if the query is not rejected, then the first component of the ciphertext that is the reply to the query is $x' + x'' \in L$, as desired.

From now, we consider the remaining case that either C' or C'' is in $\mathcal{D}$. If RevHK has been queried before, then the query is rejected. If we are in the case of Assumption U and the ρ-th query is a critical query, then the query is rejected due to the definition of Game pre-2. We suppose from now that RevHK has not been queried and (for the case of Assumption U) the ρ-th query is not a critical query. If $C' \not\in \mathcal{D}$ and $\widetilde{\pi}' \neq \widetilde{H}_{\widetilde{k}}(x', e', \widehat{\pi}')$ (note that the condition can be checked in Eval'' without the private information on the key, since now $x' \in L$), then the query is rejected. The same holds for C'' instead of C'. In these cases, the reply to the query is indeed independent of the private information of the keys. Therefore, it suffices to consider the remaining case; now the ρ-th query is the $\kappa^\dagger$-th refreshing query for some $\kappa^\dagger$.

Let $\mathcal{D} = (C_0 = C^*, C_1, \dots, C_{\kappa^\dagger - 1})$ and $\mathcal{D}' = ((D'_1, D''_1), \dots, (D'_{\kappa^\dagger - 1}, D''_{\kappa^\dagger - 1}))$ be the two dictionaries before the $\kappa^\dagger$-th refreshing query, which satisfy the conditions in the claim by the induction hypothesis. Let $C_{\kappa^\dagger}$ be the ciphertext which is the reply to the query and is added to $\mathcal{D}$ at the end of the query, and let $(D'_{\kappa^\dagger}, D''_{\kappa^\dagger})$ be the object added to $\mathcal{D}'$ at the end of the query. Let $\overline{C}_0{}^{(\kappa^\dagger)}, \overline{C}_1{}^{(\kappa^\dagger)}, \dots, \overline{C}_{\kappa^\dagger}{}^{(\kappa^\dagger)}$ be the ciphertexts calculated in the $\kappa^\dagger$-th refreshing process, hence $\overline{C}_0{}^{(\kappa^\dagger)} = \overline{C}^{*(\kappa^\dagger)}$ and $\overline{C}_{\kappa^\dagger}{}^{(\kappa^\dagger)} = C_{\kappa^\dagger}$. We note that, any object D'_i or D''_i, $1 \leq i \leq \kappa^\dagger - 1$, in $\mathcal{D}'$ which is a ciphertext (i.e., not an index) is an input ciphertext (which was not in $\mathcal{D}$) for some previous evaluation query which was not rejected, therefore it is a regular ciphertext by the enhanced rejection rule. On the other hand, since any of C' and C'' which is not in $\mathcal{D}$ is regular as discussed above, it follows that any of $D'_{\kappa^\dagger}$ and $D''_{\kappa^\dagger}$ which is a ciphertext is also regular.

From now, we first consider the case $\kappa^\dagger \neq \kappa$. We show that all the ciphertexts appearing in the $\kappa^\dagger$-th refreshing process are regular, therefore $C_{\kappa^\dagger}$ is also regular and the calculation of $C_{\kappa^\dagger}$ does not use the private information on the keys for $\mathbf{H}$, $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$, as desired. The claim for $\overline{C}_0{}^{(\kappa^\dagger)} = \overline{C}^{*(\kappa^\dagger)}$ follows from the construction $\overline{C}^{*(\kappa^\dagger)} \leftarrow \mathsf{Enc}''(M^*_\beta)$ (recall that $\kappa^\dagger \neq \kappa$). On the other hand, for $1 \leq i \leq \kappa^\dagger$, $\overline{C}_i{}^{(\kappa^\dagger)}$ is the output of Eval'' with inputs being either a ciphertext listed in $\mathcal{D}'$ or the ciphertext $\overline{C}_j{}^{(\kappa^\dagger)}$ for some $0 \leq j < i$. By the induction on i and the argument in the previous paragraph, both of the two input ciphertexts for the Eval'' are regular, therefore $\overline{C}_i{}^{(\kappa^\dagger)}$ is also regular and the calculation of $\overline{C}_i{}^{(\kappa^\dagger)}$ does not use the private information on the keys, as desired. Hence, the claim holds for the present case $\kappa^\dagger \neq \kappa$.

Finally, we consider the case $\kappa^\dagger = \kappa$. We prove the second paragraph in the statement of the claim by induction on $\kappa' = 0, 1, \dots, \kappa$. For the case $\kappa' = 0$, since $\lambda_{\mathcal{D}}(0) = 1$ and $\overline{C}_0{}^{(\kappa)} = \overline{C}^{*(\kappa)}$, the claim follows from the construction of $\overline{C}^{*(\kappa)}$ in the κ-th refreshing process. We suppose that $\kappa' > 0$. We divide the proof into the following three cases:

- Suppose that $D'_{\kappa'}$ is an index $h' \in \{0, 1, \dots, \kappa' - 1\}$ and $D''_{\kappa'}$ is an index $h'' \in \{0, 1, \dots, \kappa' - 1\}$. In this case, by the definition of the algorithm Eval'', for elements $\overline{x} \overset{\$}{\leftarrow} L$ and $\overline{e} = H_k(\overline{x})$, we have

$x_{\kappa'} = x_{h'} + x_{h''} + \overline{x}$, $e_{\kappa'} = e_{h'} + e_{h''} + \overline{e}$,

$$\begin{aligned}\widehat{\Delta}(\overline{C}_{\kappa'}{}^{(\kappa)}) = \widehat{\pi}_{\kappa'} - \widehat{H}_{\widehat{k}}(x_{\kappa'}) &= \widehat{\pi}_{h'} + \widehat{\pi}_{h''} + \widehat{H}_{\widehat{k}}(\overline{x}) - \widehat{H}_{\widehat{k}}(x_{h'}) - \widehat{H}_{\widehat{k}}(x_{h''}) - \widehat{H}_{\widehat{k}}(\overline{x}) \\ &= \widehat{\Delta}(\overline{C}_{h'}{}^{(\kappa)}) + \widehat{\Delta}(\overline{C}_{h''}{}^{(\kappa)})\end{aligned}$$

since $\widehat{H}$ is homomorphic, and $\widetilde{\pi}_{\kappa'} = \widetilde{H}_{\widetilde{k}}(x_{\kappa'}, e_{\kappa'}, \widehat{\pi}_{\kappa'})$ (we note that the fourth components of $\overline{C}_{h'}{}^{(\kappa)}$ and $\overline{C}_{h''}{}^{(\kappa)}$ are both consistent by the induction hypothesis, therefore the checks in Eval'' using the projective hash $\widetilde{H}$ are now always passed). Then, since $\lambda_{\mathcal{D}}(\kappa') = \lambda_{\mathcal{D}}(h') + \lambda_{\mathcal{D}}(h'')$ by the definition, the induction hypothesis for $\overline{C}_{h'}{}^{(\kappa)}$ and $\overline{C}_{h''}{}^{(\kappa)}$ implies that the claim here also holds for the $\overline{C}_{\kappa'}{}^{(\kappa)}$.

- Suppose that $D'_{\kappa'}$ is an index $h' \in \{0, 1, \dots, \kappa' - 1\}$ and $D''_{\kappa'} = (x^{\dagger}, e^{\dagger}, \widehat{\pi}^{\dagger}, \widetilde{\pi}^{\dagger})$ is a ciphertext. In this case, by the definition of the algorithm Eval'', for elements $\overline{x} \xleftarrow{\$} L$ and $\overline{e} = H_k(\overline{x})$, we have $x_{\kappa'} = x_{h'} + x^{\dagger} + \overline{x}$, $e_{\kappa'} = e_{h'} + e^{\dagger} + \overline{e}$,

$$\begin{aligned}\widehat{\Delta}(\overline{C}_{\kappa'}{}^{(\kappa)}) = \widehat{\pi}_{\kappa'} - \widehat{H}_{\widehat{k}}(x_{\kappa'}) &= \widehat{\pi}_{h'} + \widehat{\pi}^{\dagger} + \widehat{H}_{\widehat{k}}(\overline{x}) - \widehat{H}_{\widehat{k}}(x_{h'}) - \widehat{H}_{\widehat{k}}(x^{\dagger}) - \widehat{H}_{\widehat{k}}(\overline{x}) \\ &= \widehat{\Delta}(\overline{C}_{h'}{}^{(\kappa)}) + \widehat{\Delta}(D''_{\kappa'})\end{aligned}$$

since $\widehat{H}$ is homomorphic, and $\widetilde{\pi}_{\kappa'} = \widetilde{H}_{\widetilde{k}}(x_{\kappa'}, e_{\kappa'}, \widehat{\pi}_{\kappa'})$ (we note that the fourth components of $\overline{C}_{h'}{}^{(\kappa)}$ and $D''_{\kappa'}$ are both consistent by the induction hypothesis, therefore the checks in Eval'' using the projective hash $\widetilde{H}$ are now always passed). Then, since $\lambda_{\mathcal{D}}(\kappa') = \lambda_{\mathcal{D}}(h')$ by the definition, the induction hypothesis for $\overline{C}_{h'}{}^{(\kappa)}$ implies that the claim here also holds for the $\overline{C}_{\kappa'}{}^{(\kappa)}$.

- Suppose that $D'_{\kappa'}$ is a ciphertext and $D''_{\kappa'}$ is an index. In the case, the symmetry implies that the claim holds by the same argument as the previous case.

Hence, the second paragraph in the statement of the claim holds for any of the three cases.

Finally, by the result above, for the second component e_{κ} of C_{κ}, since $e^{*(\kappa)} = M^{*}_{\beta} + H_k(x^{*(\kappa)})$, only the element, among the elements needed to compute e_{κ}, which may require the private information on the keys is the value $H_k(x^{*(\kappa)})$ of the projective hash H with key k. For the third component $\widehat{\pi}_{\kappa}$ of C_{κ}, only the element, among the elements needed to compute $\widehat{\pi}_{\kappa} = \widehat{H}_{\widehat{k}}(x_{\kappa}) + \widehat{\Delta}(C_{\kappa})$, which may require the private information on the keys is the value $\widehat{H}_{\widehat{k}}(x_{\kappa})$ of the projective hash $\widehat{H}$ with key $\widehat{k}$. Moreover, it is obvious that the computation of the fourth component $\widetilde{\pi}_{\kappa} = \widetilde{H}_{\widetilde{k}}(x_{\kappa}, e_{\kappa}, \widehat{\pi}_{\kappa})$ consists of only one computation of a value of the projective hash $\widetilde{H}$ with key $\widetilde{k}$ that may require the private information on the keys. This completes the proof of the claim. □

Claim 2. In SubSubGame κ.3.ρ, suppose that the κ-th refreshing query is ρ-th or earlier query and $C_{\kappa} = (x_{\kappa}, e_{\kappa}, \widehat{\pi}_{\kappa}, \widetilde{\pi}_{\kappa})$ is the reply to the κ-th refreshing query. Then, for the case of Assumption U, if $\lambda_{\mathcal{D}}(\kappa)$ is not a multiple of $o(\Lambda)$ (see Definition 4.3 for the definition of $o(\Lambda)$), then both $\lambda_{\mathcal{D}}(\kappa) \cdot x^{*(\kappa)}$ and x_{κ} are uniformly random over $X' \setminus L$; otherwise, both $\lambda_{\mathcal{D}}(\kappa) \cdot x^{*(\kappa)}$ and x_{κ} are always in L.

Proof of Claim 2. First, we consider the case that $\lambda_{\mathcal{D}}(\kappa)$ is a multiple of $o(\Lambda)$. By the definition of $o(\Lambda)$, $\lambda_{\mathcal{D}}(\kappa)$ is a multiple of the order of $x^{*(\kappa)}$ in the quotient group X/L, therefore we have $\lambda_{\mathcal{D}}(\kappa) \cdot x^{*(\kappa)} \in L$. Since $x_{\kappa} - \lambda_{\mathcal{D}}(\kappa) \cdot x^{*(\kappa)} \in L$ by Claim 1, it follows that $x_{\kappa} \in L$, as desired.

Secondly, we consider the other case that $\lambda_{\mathcal{D}}(\kappa)$ is not a multiple of $o(\Lambda)$. Recall that any critical query is rejected owing to the definition of Game pre-2; therefore, $\lambda_{\mathcal{D}}(h)$ is not a critical integer for any index h by induction on h. By the definition of critical integers, it follows that $\lambda_{\mathcal{D}}(\kappa)$ is coprime to $|X|$. Therefore, there is an integer λ' satisfying that $\lambda_{\mathcal{D}}(\kappa)\lambda' \equiv 1 \bmod |X|$, which is also coprime to $|X|$. This relation implies that the multiplications by $\lambda_{\mathcal{D}}(\kappa)$ and by λ' define two mappings $X \to X$ which are inverses of each other. Moreover, each of the mappings maps L to L since L is a subgroup of X, while by Assumtion U, it maps $X' \setminus L$ to X'. This implies that the multiplication by $\lambda_{\mathcal{D}}(\kappa)$ defines a bijection $X' \setminus L \to X' \setminus L$, therefore $\lambda_{\mathcal{D}}(h) \cdot x^{*(\kappa)}$ is uniformly random over $X' \setminus L$ as well as $x^{*(\kappa)}$. On the other hand, by Assumption U, for any $y \in L$, the addition by y defines a bijection $X' \setminus L \to X' \setminus L$ (since L is a subgroup of X). Since

$x_\kappa - \lambda_{\mathcal{D}}(\kappa) \cdot x^{*(\kappa)}$ is an element of L independent of $x^{*(\kappa)}$ by Claim 1, it follows that x_κ is also uniformly random over $X' \setminus L$, as desired. This completes the proof of Claim 2. □

Before evaluating the quantities $\Pr[R_\ell^{(\kappa.3.\rho)}]$, we also introduce subdivision of the game-hopping from SubGame κ.4 to SubGame κ.5 for $1 \leq \kappa \leq Q(\ell)$ and subdivision of the game-hopping from Game con-2 to Game con-3 in a similar way:

SubSubGame κ.5.ρ ($0 \leq \rho \leq Q(\ell)$): In comparison to SubGame κ.4, in the game, we replace the enhanced rejection rules for the $(Q(\ell)+1-\rho)$-th or later queries with the original rejection rules. Note that SubSubGame κ.5.$Q(\ell)$ and SubSubGame κ.5.0 are the same as SubGame κ.4 and SubGame κ.5, respectively.

SubGame con-3.ρ ($0 \leq \rho \leq Q(\ell)$): In comparison to Game con-2, in the game, we replace the original rejection rules for the first ρ queries with the enhanced rejection rules. Note that SubGame con-3.ρ and SubGame con-3.$Q(\ell)$ are the same as Game con-2 and Game con-3, respectively.

Now we note that each SubSubGame κ.5.ρ satisfies properties similar to Claim 1 and Claim 2 above, where the $(Q(\ell)+1-\rho)$-th query plays the role of the ρ-th query in the original Claim 1 and Claim 2. On the other hand, for each SubGame con-3.ρ, an argument similar to Claim 1 implies the following property:

Claim 3. In SubGame con-3.ρ, the replies to the first ρ queries are independent of the private information on the keys for the projective hash families $\mathbf{H}$, $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$. Moreover, the replies to the evaluation queries (which are not rejected) among the first ρ queries and the ciphertexts in $\mathcal{D}$ at the end of the ρ-th query are regular ciphertexts, except the challenge ciphertext C^* in $\mathcal{D}$ which depends on one value of the private evaluation algorithm for each of these projective hash families.

Let $R_\ell^{(\kappa.5.\rho)}$ denote the event in SubSubGame κ.5.ρ that the $(Q(\ell)+1-\rho)$-th query (exists and) is rejected by the enhanced rejection rule but is not rejected by the original rejection rule. Similarly, let $R_\ell^{(\text{con-3}.\rho)}$ denote the event in SubGame con-3.ρ that the ρ-th query (exists and) is rejected by the enhanced rejection rule but is not rejected by the original rejection rule. Then an argument similar to the case of SubSubGame κ.3.ρ implies that $|\Pr[T_\ell^{(\kappa.4)}] - \Pr[T_\ell^{(\kappa.5)}]| \leq \sum_{\rho=1}^{Q(\ell)} \Pr[R_\ell^{(\kappa.5.\rho)}]$ and $|\Pr[T_\ell^{(\text{con-2})}] - \Pr[T_\ell^{(\text{con-3})}]| \leq \sum_{\rho=1}^{Q(\ell)} \Pr[R_\ell^{(\text{con-3}.\rho)}]$.

In order to evaluate the quantities $\Pr[R_\ell^{(\kappa.3.\rho)}]$, $\Pr[R_\ell^{(\kappa.5.\rho)}]$ and $\Pr[R_\ell^{(\text{con-3}.\rho)}]$, we introduce further the following events:

- We define $R_\ell^{\langle 1\rangle(\kappa.3.\rho)}$ to be the event in SubSubGame κ.3.ρ that the ρ-th query is a decryption query C with $C = (x, e, \widehat{\pi}, \widetilde{\pi})$ and is in the find stage, RevHK has been queried before the ρ-th query, $x \notin L$ and $\widehat{\pi} = \widehat{H}_{\widehat{k}}(x)$. In a similar manner, we also define the events $R_\ell^{\langle 1\rangle(\kappa.5.\rho)}$ (where we focus on the $(Q(\ell)+1-\rho)$-th query instead of the ρ-th query) and $R_\ell^{\langle 1\rangle(\text{con-3}.\rho)}$.
- We define $R_\ell^{\langle 2\rangle(\kappa.3.\rho)}$ to be the event in SubSubGame κ.3.ρ that the ρ-th query is a decryption query C with $C = (x, e, \widehat{\pi}, \widetilde{\pi})$, RevHK has not been queried before the ρ-th query, $x \notin L$, $\widetilde{\pi} = \widetilde{H}_{\widetilde{k}}(x, e, \widehat{\pi})$, and either the ρ-th query is before the κ-th refreshing query, or the ρ-th query is after the κ-th refreshing query and the reply C_κ to the κ-th refreshing query is a regular ciphertext. In a similar manner, we also define the event $R_\ell^{\langle 2\rangle(\kappa.5.\rho)}$, where we focus on the $(Q(\ell)+1-\rho)$-th query instead of the ρ-th query. Moreover, we also define $R_\ell^{\langle 2\rangle(\text{con-3}.\rho)}$ to be the event in SubGame con-3.ρ that the ρ-th query is a decryption query C with $C = (x, e, \widehat{\pi}, \widetilde{\pi})$ and is in the find stage, RevHK has not been queried before the ρ-th query, $x \notin L$ and $\widetilde{\pi} = \widetilde{H}_{\widetilde{k}}(x, e, \widehat{\pi})$.
- We define $R_\ell^{\langle 3\rangle(\kappa.3.\rho)}$ to be the event in SubSubGame κ.3.ρ that the ρ-th query is an evaluation query (C', C'') with $C' = (x', e', \widehat{\pi}', \widetilde{\pi}') \notin \mathcal{D}$, RevHK has not been queried before the ρ-th query, $x' \notin L$, $\widetilde{\pi}' = \widetilde{H}_{\widetilde{k}}(x', e', \widehat{\pi}')$, and either the ρ-th query is the κ-th refreshing query or before the κ-th refreshing query, or the ρ-th query is after the κ-th refreshing query and the reply to the κ-th refreshing query is a regular ciphertext. In a similar manner, we also define the event $R_\ell^{\langle 3\rangle(\kappa.5.\rho)}$, where we focus on the $(Q(\ell)+1-\rho)$-th query instead of the ρ-th query. Moreover, we also define $R_\ell^{\langle 3\rangle(\text{con-3}.\rho)}$ to be the event

in SubGame con-3.ρ that the ρ-th query is an evaluation query (C', C'') with $C' = (x', e', \widehat{\pi}', \widetilde{\pi}')$ and is in the find stage, RevHK has not been queried before the ρ-th query, $x' \notin L$ and $\widetilde{\pi}' = \widetilde{H}_{\widetilde{k}}(x', e', \widehat{\pi}')$.

- We define the events $R_\ell^{\langle 4\rangle\,(\kappa.3.\rho)}$, $R_\ell^{\langle 4\rangle\,(\kappa.5.\rho)}$ and $R_\ell^{\langle 4\rangle\,(\text{con-3}.\rho)}$ in a way similar to the events $R_\ell^{\langle 3\rangle}$ above, where we focus on the second part $C'' \notin \mathcal{D}$ of the input for the evaluation query instead of the first part C'.

- We define $R_\ell^{\langle 5\rangle\,(\kappa.3.\rho)}$ to be the event in SubSubGame $\kappa.3.\rho$ that the ρ-th query is a decryption query C with $C = (x, e, \widehat{\pi}, \widetilde{\pi}) \notin \mathcal{D}$ and is after the κ-th refreshing query, RevHK has not been queried before the ρ-th query, $x \notin L$, $\widetilde{\pi} = \widetilde{H}_{\widetilde{k}}(x, e, \widehat{\pi})$, and the reply to the κ-th refreshing query is an irregular ciphertext. In a similar manner, we also define the event $R_\ell^{\langle 5\rangle\,(\kappa.5.\rho)}$, where we focus on the $(Q(\ell)+1-\rho)$-th query instead of the ρ-th query. Moreover, we also define $R_\ell^{\langle 5\rangle\,(\text{con-3}.\rho)}$ to be the event in SubGame con-3.ρ that the ρ-th query is a decryption query C with $C = (x, e, \widehat{\pi}, \widetilde{\pi}) \notin \mathcal{D}$ and is in the guess stage, RevHK has not been queried before the ρ-th query, $x \notin L$ and $\widetilde{\pi} = \widetilde{H}_{\widetilde{k}}(x, e, \widehat{\pi})$.

- We define $R_\ell^{\langle 6\rangle\,(\kappa.3.\rho)}$ to be the event in SubSubGame $\kappa.3.\rho$ that the ρ-th query is an evaluation query (C', C'') with $C' = (x', e', \widehat{\pi}', \widetilde{\pi}') \notin \mathcal{D}$, the ρ-th query is after the κ-th refreshing query, RevHK has not been queried before the ρ-th query, $x' \notin L$, $\widetilde{\pi}' = \widetilde{H}_{\widetilde{k}}(x', e', \widehat{\pi}')$, and the reply to the κ-th refreshing query is an irregular ciphertext. In a similar manner, we also define the event $R_\ell^{\langle 6\rangle\,(\kappa.5.\rho)}$, where we focus on the $(Q(\ell)+1-\rho)$-th query instead of the ρ-th query. Moreover, we also define $R_\ell^{\langle 6\rangle\,(\text{con-3}.\rho)}$ to be the event in SubGame con-3.ρ that the ρ-th query is an evaluation query (C', C'') with $C' = (x', e', \widehat{\pi}', \widetilde{\pi}') \notin \mathcal{D}$, the ρ-th query is in the guess stage, RevHK has not been queried before the ρ-th query, $x' \notin L$ and $\widetilde{\pi}' = \widetilde{H}_{\widetilde{k}}(x', e', \widehat{\pi}')$.

- We define the events $R_\ell^{\langle 7\rangle\,(\kappa.3.\rho)}$, $R_\ell^{\langle 7\rangle\,(\kappa.5.\rho)}$ and $R_\ell^{\langle 7\rangle\,(\text{con-3}.\rho)}$ in a way similar to the events $R_\ell^{\langle 6\rangle}$ above, where we focus on the second part $C'' \notin \mathcal{D}$ of the input for the evaluation query instead of the first part C'.

By the definitions of the events, we have $\Pr[R_\ell^{(\kappa.3.\rho)}] \le \sum_{i=1}^{7} \Pr[R_\ell^{\langle i\rangle\,(\kappa.3.\rho)}]$, and similar inequalities hold for $R_\ell^{(\kappa.5.\rho)}$ and $R_\ell^{(\text{con-3}.\rho)}$. Therefore, we have

$$|\delta_3| \le \sum_{i=1}^{7} \sum_{\rho=1}^{Q(\ell)} \left(\sum_{\kappa=1}^{Q(\ell)} \left(\Pr[R_\ell^{\langle i\rangle\,(\kappa.3.\rho)}] + \Pr[R_\ell^{\langle i\rangle\,(\kappa.5.\rho)}]\right) + \Pr[R_\ell^{\langle i\rangle\,(\text{con-3}.\rho)}] \right) \ .$$

We evaluate the quantities in the right-hand side of the inequality. Here, we put $\overline{\rho} = \rho$ for the case of events $R_\ell^{(\kappa.3.\rho)}$ and $R_\ell^{(\text{con-3}.\rho)}$, and $\overline{\rho} = Q(\ell) + 1 - \rho$ for the case of events $R_\ell^{(\kappa.5.\rho)}$.

For the events $R_\ell^{\langle 1\rangle}$, Claim 1 and Claim 3 imply that the probability that $x \notin L$ but $\widehat{H}_{\widehat{k}}(x) = \widehat{\pi}$ as in the event is bounded by a negligible value common to all κ and ρ owing to the universal$_1$ property of $\widehat{\mathbf{H}}$, since the private information on $\widehat{k}$ is not used in the game before the $\overline{\rho}$-th query. Hence, the sum of $\Pr[R_\ell^{\langle 1\rangle\,(\kappa.3.\rho)}]$, $\Pr[R_\ell^{\langle 1\rangle\,(\kappa.5.\rho)}]$ and $\Pr[R_\ell^{\langle 1\rangle\,(\text{con-3}.\rho)}]$ over all κ and ρ is negligible.

For the case of Assumption I, a similar argument based on Claim 1 and Claim 3 implies that the sum of $\Pr[R_\ell^{\langle i\rangle\,(\kappa.3.\rho)}]$, $\Pr[R_\ell^{\langle i\rangle\,(\kappa.5.\rho)}]$ and $\Pr[R_\ell^{\langle i\rangle\,(\text{con-3}.\rho)}]$ over all $i \in \{2, 3, 4\}$, κ and ρ is negligible owing to the universal$_1$ property of $\widetilde{\mathbf{H}}$, since the private information on $\widetilde{k}$ is not used in the game before the $\overline{\rho}$-th query. Similarly, Claim 1 and Claim 3 imply that the sum of $\Pr[R_\ell^{\langle i\rangle\,(\kappa.3.\rho)}]$, $\Pr[R_\ell^{\langle i\rangle\,(\kappa.5.\rho)}]$ and $\Pr[R_\ell^{\langle i\rangle\,(\text{con-3}.\rho)}]$ over all $i \in \{5, 6, 7\}$, κ and ρ is negligible owing to the universal$_2$ property of $\widetilde{\mathbf{H}}$, since the private information on $\widetilde{k}$ is not used in the game before the $\overline{\rho}$-th query except for the computation of the fourth component of the reply to the κ-th refreshing query (for the case of events $R_\ell^{\langle i\rangle\,(\kappa.3.\rho)}$ and $R_\ell^{\langle i\rangle\,(\kappa.5.\rho)}$) or the computation of the fourth component of the challenge ciphertext (for the case of event $R_\ell^{\langle i\rangle\,(\text{con-3}.\rho)}$). Summarizing, $|\delta_3|$ is negligible for the case of Assumption I.

From now, we consider the other cases of Assumption A and Assumption U. In these cases, we reduce the evaluation of the probabilities $\Pr[R_\ell^{\langle i\rangle\,(\kappa.3.\rho)}]$, $\Pr[R_\ell^{\langle i\rangle\,(\kappa.5.\rho)}]$ and $\Pr[R_\ell^{\langle i\rangle\,(\text{con-}3.\rho)}]$ for $2 \leq i \leq 7$ to evaluation of the advantage of some adversary for the security game of the first-adaptive or first-uniform computationally universal$_2$ property for $\widetilde{\mathbf{H}}$. Let Hash denote the oracle in the security game for $\widetilde{\mathbf{H}}$. The adversary will be defined in such a way that it simulates the underlying games for the events $R_\ell^{\langle i\rangle\,(\kappa.3.\rho)}$, $R_\ell^{\langle i\rangle\,(\kappa.5.\rho)}$ and $R_\ell^{\langle i\rangle\,(\text{con-}3.\rho)}$ (which we call an internal game). Here we note that, for the internal game, the behavior of the decryption oracle under the enhanced rejection rule can be efficiently simulated by using Hash, secret key k for $\mathbf{H}$ and secret key $\widehat{k}$ for $\widehat{\mathbf{H}}$; it can be checked whether the query is regular or not by an oracle query to Hash (note that Hash replies $\perp$ if and only if the first component of the input to the oracle is in $X \setminus L$), and the value of $\widetilde{H}$ for any regular input can be obtained by querying it to Hash. Similarly, Claim 1 and Claim 3 imply that the behavior of the evaluation oracle under the enhanced rejection rule can be efficiently simulated by using Hash, k and $\widehat{k}$ except for the κ-th refreshing process for the case of events $R_\ell^{\langle i\rangle\,(\kappa.3.\rho)}$ and $R_\ell^{\langle i\rangle\,(\kappa.5.\rho)}$ (we note that, in any refreshing process under the enhanced rejection rule, every ciphertext appearing during the process has its fourth component being consistent with the value of $\widetilde{H}$ for the first three components, therefore the consistency checks using $\widetilde{H}$ during the process can be omitted). The computation of the challenge ciphertext for the case of events $R_\ell^{\langle i\rangle\,(\kappa.3.\rho)}$ and $R_\ell^{\langle i\rangle\,(\kappa.5.\rho)}$ can be efficiently simulated by using Hash, k and $\widehat{k}$ as well. In the argument below, we say that the adversary **aborts the game**, to mean the following situation:

- For the case of Assumption A, the adversary submits $(0,0,0) \in X \times \Pi \times \widehat{\Pi}$ to the challenger unless an element has been submitted, and then outputs $((0,0,0),0) \in (X \times \Pi \times \widehat{\Pi}) \times \widetilde{\Pi}$.
- For the case of Assumption U, the adversary outputs $((0,0,0),0) \in (X \times \Pi \times \widehat{\Pi}) \times \widetilde{\Pi}$.

Hence, when the adversary aborts the game, it never wins the security game for $\widetilde{\mathbf{H}}$.

Based on the argument above, first, in order to evaluate the probabilities for $2 \leq i \leq 4$, we define an adversary $\mathcal{A}_{3,1}$ for the security game for $\widetilde{\mathbf{H}}$ as follows:

Adversary $\mathcal{A}_{3,1}$: First, we specify the input for $\mathcal{A}_{3,1}$ as follows:

- For the case of Assumption A, the input for $\mathcal{A}_{3,1}$ is a public key $\widetilde{s} = \widetilde{\alpha}(\widetilde{k})$ corresponding to a key $\widetilde{k}$ for $\widetilde{\mathbf{H}}$, as well as the underlying parameters 1^ℓ and $\Lambda = \Lambda[X, X', L, W, R]$.
- For the case of Assumption U, the input for $\mathcal{A}_{3,1}$ is a public key $\widetilde{s} = \widetilde{\alpha}(\widetilde{k})$ corresponding to a key $\widetilde{k}$ for $\widetilde{\mathbf{H}}$, a uniformly random element $(x^{**}, e^{**}, \widehat{\pi}^{**})$ of $(X' \setminus L) \times \Pi \times \widehat{\Pi}$ and the value $\pi^{**} = \widetilde{H}_{\widetilde{k}}(x^{**}, e^{**}, \widehat{\pi}^{**})$, as well as the underlying parameters 1^ℓ and $\Lambda = \Lambda[X, X', L, W, R]$.

Given the input as above, the adversary first generates the keys k, $\widehat{k}$, $s = \alpha(k)$ and $\widehat{s} = \widehat{\alpha}(\widehat{k})$ by using $\mathsf{HPS.param}(1^\ell, \Lambda)$ and $\widehat{\mathsf{HPS}}\mathsf{.param}(1^\ell, \Lambda)$. Secondly, the adversary chooses an internal game from SubSubGame $\kappa.3.\rho$, SubSubGame $\kappa.5.\rho$ (for $1 \leq \kappa \leq Q(\ell)$ and $1 \leq \rho \leq Q(\ell)$) and SubGame con-3.ρ (for $1 \leq \rho \leq Q(\ell)$), $(2Q(\ell)+1)Q(\ell)$ candidates in total, and the adversary chooses a mode from dec, eval_1 and eval_2. Then the adversary performs the followings:

- For the case that SubSubGame $\kappa.3.\rho$ was chosen as the internal game, the adversary chooses an element $x^{*(\kappa)} \in X' \setminus L$ uniformly at random, and simulates the internal game until the ρ-th query. Here, if RevHK is queried in the internal game, then the adversary aborts the game. On the other hand, by using Hash, k and $\widehat{k}$ as discussed above, the adversary simulates the decryption oracle before the ρ-th query, the evaluation oracle before the ρ-th query except for the κ-th refreshing query, and the computation of the challenge ciphertext. Moreover, the adversary simulates the κ-th refreshing process in the following manner:
  - The adversary computes the first three components of the reply $C_\kappa = (x_\kappa, e_\kappa, \widehat{\pi}_\kappa, \widetilde{\pi}_\kappa)$ to the κ-th refreshing query by using the element $x^{*(\kappa)}$ as above, the key k for $\mathbf{H}$ and the key $\widehat{k}$ for $\widehat{\mathbf{H}}$.

  - Then the adversary queries $(x_\kappa, e_\kappa, \widehat{\pi}_\kappa)$ to Hash and receives the reply. Now, the adversary aborts the game if the reply by Hash is $\perp$; while, if the reply by Hash is an element of $\widetilde{\Pi}$, then the adversary sets $\widetilde{\pi}$ to be the reply by Hash.

  Finally, if the internal game ends until the ρ-th query is performed, then the adversary aborts the game. For the other case, at the ρ-th query in the internal game, for the case of Assumption A, the adversary chooses $x^{**} \in X' \setminus L$, $e^{**} \in \Pi$ and $\widehat{\pi}^{**} \in \widehat{\Pi}$ uniformly at random and submits $(x^{**}, e^{**}, \widehat{\pi}^{**})$ to the challenger of the security game for $\widetilde{\mathbf{H}}$. Then the adversary performs as follows:

  - In the case that the mode dec was chosen, the adversary aborts the game unless the ρ-th query is a decryption query. On the other hand, for the case that the ρ-th query is a decryption query $C = (x, e, \widehat{\pi}, \widetilde{\pi})$, if $C \in \mathcal{D}$, then the adversary aborts the game; while if $C \notin \mathcal{D}$, then the adversary outputs $(x, e, \widehat{\pi}) \in X \times \Pi \times \widehat{\Pi}$ and $\widetilde{\pi} \in \widetilde{\Pi}$.
  - In the case that the mode eval_1 was chosen, the adversary aborts the game unless the ρ-th query is an evaluation query. On the other hand, for the case that the ρ-th query is an evaluation query (C', C'') with $C' = (x', e', \widehat{\pi}', \widetilde{\pi}')$, if $C' \in \mathcal{D}$, then the adversary aborts the game; while if $C'' \notin \mathcal{D}$, then the adversary outputs $(x', e', \widehat{\pi}') \in X \times \Pi \times \widehat{\Pi}$ and $\widetilde{\pi}' \in \widetilde{\Pi}$.
  - In the case that the mode eval_2 was chosen, the adversary aborts the game unless the ρ-th query is an evaluation query. On the other hand, for the case that the ρ-th query is an evaluation query (C', C'') with $C'' = (x'', e'', \widehat{\pi}'', \widetilde{\pi}'')$, if $C'' \in \mathcal{D}$, then the adversary aborts the game; while if $C'' \notin \mathcal{D}$, then the adversary outputs $(x'', e'', \widehat{\pi}'') \in X \times \Pi \times \widehat{\Pi}$ and $\widetilde{\pi}'' \in \widetilde{\Pi}$.

- For the case that SubSubGame $\kappa.5.\rho$ was chosen, the adversary performs in a similar manner to the case of SubSubGame $\kappa.3.\rho$ above by simulating SubSubGame $\kappa.5.\rho$, where the $(Q(\ell) + 1 - \rho)$-th query in the present case plays the role of the ρ-th query in the case of SubSubGame $\kappa.3.\rho$.

- For the case that SubGame con-3.ρ was chosen, the adversary performs in a similar manner to the case of SubSubGame $\kappa.3.\rho$ above by simulating SubGame con-3.ρ. Here, the differences from the previous case are the followings: For the simulation of a refreshing process, the adversary performs as in the case $\kappa' \neq \kappa$ of the description above. On the other hand, if the challenge phase comes before the ρ-th query, then the adversary aborts the game.

By the construction of the adversary $\mathcal{A}_{3,1}$, when SubSubGame $\kappa.3.\rho$ and the mode dec (respectively, eval_1 and eval_2) are chosen, $\mathcal{A}_{3,1}$ wins the security game for $\widetilde{\mathbf{H}}$ if and only if the event $R_\ell^{\langle i \rangle\,(\kappa.3.\rho)}$ for $i = 2$ (respectively, $i = 3$ and $i = 4$) occurs in the internal game and $(x^{**}, e^{**}, \widehat{\pi}^{**})$ is different from the first part (in $X \times \Pi \times \widehat{\Pi}$) of the output by $\mathcal{A}_{3,1}$ (note that, when the κ-th refreshing query is before the ρ-th query, the condition that the reply to the κ-th refreshing query is a regular ciphertext is guaranteed by the property that the oracle Hash in the simulation of the κ-th refreshing process does not reply $\perp$). The same holds for $R_\ell^{\langle i \rangle\,(\kappa.5.\rho)}$ and $R_\ell^{\langle i \rangle\,(\text{con-3}.\rho)}$ instead of $R_\ell^{\langle i \rangle\,(\kappa.3.\rho)}$. Since the probability that the uniformly random $(x^{**}, e^{**}, \widehat{\pi}^{**})$ coincides with the first part of the output of $\mathcal{A}_{3,1}$ (which is independent of $(x^{**}, e^{**}, \widehat{\pi}^{**})$) is negligible (note that $1/|X' \setminus L|$ is negligible, since otherwise the subset membership problem $\mathbf{M}$ becomes not hard relative to X'), the advantage $Adv_{\mathcal{A}_{3,1}}$ of $\mathcal{A}_{3,1}$ satisfies that the difference

$$\left| Adv_{\mathcal{A}_{3,1}} - \frac{1}{3 \cdot Q'} \sum_{i=2}^{4} \sum_{\rho=1}^{Q(\ell)} \left(\sum_{\kappa=1}^{Q(\ell)} \left(\Pr[R_\ell^{\langle i \rangle\,(\kappa.3.\rho)}] + \Pr[R_\ell^{\langle i \rangle\,(\kappa.5.\rho)}] \right) + \Pr[R_\ell^{\langle i \rangle\,(\text{con-3}.\rho)}] \right) \right| ,$$

where $Q' = (2Q(\ell) + 1)Q(\ell)$, is negligible. Moreover, only the steps in $\mathcal{A}_{3,1}$ that may be not efficient are to choose some elements of $X' \setminus L$ uniformly at random. Owing to the assumption that $X' \setminus L$ is approximately samplable relative to X, we define $\mathcal{A}'_{3,1}$ by replacing the uniform distribution on $X' \setminus L$ sampled in $\mathcal{A}_{3,1}$ with an efficiently samplable distribution on X with negligible statistical distance. Then $|Adv_{\mathcal{A}'_{3,1}} - Adv_{\mathcal{A}_{3,1}}|$ is negligible, while $Adv_{\mathcal{A}'_{3,1}}$ is negligible since $\mathcal{A}'_{3,1}$ is PPT. By these arguments, it follows (since Q' is a polynomial) that the sum of $\Pr[R_\ell^{\langle i\rangle(\kappa.3.\rho)}]$, $\Pr[R_\ell^{\langle i\rangle(\kappa.5.\rho)}]$ and $\Pr[R_\ell^{\langle i\rangle(\text{con-3}.\rho)}]$ for all $i \in \{2,3,4\}$, κ and ρ is negligible as well.

Secondly, in order to evaluate the probabilities $\Pr[R_\ell^{\langle i\rangle(\kappa.3.\rho)}]$, $\Pr[R_\ell^{\langle i\rangle(\kappa.5.\rho)}]$ and $\Pr[R_\ell^{\langle i\rangle(\text{con-3}.\rho)}]$ for $i \in \{5,6,7\}$, we define an adversary $\mathcal{A}_{3,2}$ for the security game for $\widetilde{\mathbf{H}}$ as follows:

Adversary $\mathcal{A}_{3,2}$: The basic construction of $\mathcal{A}_{3,2}$ is similar to $\mathcal{A}_{3,1}$; here we only describe the differences from the construction of $\mathcal{A}_{3,1}$. First, the adversary omits the submission of $(x^{**}, e^{**}, \widehat{\pi}^{**})$ which the previous adversary $\mathcal{A}_{3,1}$ performs for the case of Assumption A. Secondly, for the case of SubSubGame $\kappa.3.\rho$, the simulation of the κ-th refreshing process is now performed as follows:

- To compute the reply $C_\kappa = (x_\kappa, e_\kappa, \widehat{\pi}_\kappa, \widetilde{\pi}_\kappa)$ to the κ-th refreshing query, the adversary first computes x_κ from the given $x^{*(\kappa)}$ and the integer $\lambda_{\mathcal{D}}(\kappa)$ as in Claim 1. Namely, x_κ is the sum of $\lambda_{\mathcal{D}}(\kappa) \cdot x^{*(\kappa)}$ and an element of L independent of $x^{*(\kappa)}$.
- The adversary computes e_κ as in Claim 1. Namely, e_κ is the sum of $\lambda_{\mathcal{D}}(\kappa) \cdot (M_\beta^* + H_k(x^{*(\kappa)})) = \lambda_{\mathcal{D}}(\kappa) \cdot M_\beta^* + H_k(\lambda_{\mathcal{D}}(\kappa) \cdot x^{*(\kappa)})$, an integer linear combination of elements $H_k(\overline{x})$ for some $\overline{x} \in L$ independent of $x^{*(\kappa)}$, and an integer linear combination of the second components of ciphertexts listed in $\mathcal{D}'$.
- The adversary computes $\widehat{\pi}_\kappa$ as in Claim 1. Namely, $\widehat{\pi}_\kappa = \widehat{H}_{\widehat{k}}(x_\kappa) + \widehat{\Delta}(C_\kappa)$ and $\widehat{\Delta}(C_\kappa)$ is computed without the private information on $\widehat{k}$ as an integer linear combination of $\widehat{\Delta}(D'_h)$ and $\widehat{\Delta}(D''_h)$ for ciphertexts D'_h and D''_h listed in $\mathcal{D}'$.
- Then the adversary queries $(x_\kappa, e_\kappa, \widehat{\pi}_\kappa)$ to Hash. If the reply by Hash is not $\perp$, then the adversary aborts the game. Otherwise:
  - For the case of Assumption A, the adversary submits $(x_\kappa, e_\kappa, \widehat{\pi}_\kappa)$ to the challenger in the security game for $\widetilde{\mathbf{H}}$, and sets $\widetilde{\pi}_\kappa$ to be the reply by the challenger. Then the adversary returns the C_κ as the reply to the κ-th refreshing query.
  - For the case of Assumption U, the adversary returns $(x^{**}, e^{**}, \widehat{\pi}^{**}, \widetilde{\pi}^{**})$, instead of the C_κ above, as the reply to the κ-th refreshing query.

For the case of SubSubGame $\kappa.5.\rho$, the simulation of the κ-th refreshing process is similar as above, where the adversary uses, instead of M_β^*, a value $\pi^\dagger \in \Pi$ chosen according to the probability distribution which is negligibly close to the uniform distribution on Π' specified by the assumption that Π' is approximately samplable relative to Π. Moreover, for the cases of SubSubGame $\kappa.3.\rho$ and SubSubGame $\kappa.5.\rho$, the adversary aborts the game unless the $\overline{\rho}$-th query (where $\overline{\rho} = \rho$ in the case of SubSubGame $\kappa.3.\rho$ and $\overline{\rho} = Q(\ell) + 1 - \rho$ in the case of SubSubGame $\kappa.5.\rho$) comes after the κ-th refreshing query. On the other hand, for the case of SubGame con-3.ρ, the adversary aborts the game in the case that the challenge phase does not come before the ρ-th query (instead of the case that the challenge phase comes before the ρ-th query, as in the previous adversary $\mathcal{A}_{3,1}$). Then, for the case of SubGame con-3.ρ, the adversary simulates the challenge phase in the following manner:

- Given the challenge plaintexts (M_0^*, M_1^*), the adversary chooses the challenge bit β uniformly at random. Then, by using the uniformly random element $x^* \in X' \setminus L$ which is chosen by the adversary at the beginning of the internal game, the adversary computes $e^* = M_\beta^* + H_k(x^*)$ and $\widehat{\pi}^* = \widehat{H}_{\widehat{k}}(x^*)$.
- For the case of Assumption A, the adversary submits $(x^*, e^*, \widehat{\pi}^*)$ to the challenger in the security game for $\widetilde{\mathbf{H}}$, and sets $\widetilde{\pi}^*$ to be the reply by the challenger. Then the adversary returns $C^* = (x^*, e^*, \widehat{\pi}^*, \widetilde{\pi}^*)$ as the challenge ciphertext.
- For the case of Assumption U, the adversary returns $(x^{**}, e^{**}, \widehat{\pi}^{**}, \widetilde{\pi}^{**})$, instead of the elements x^*, e^* and $\widehat{\pi}^*$ above, as the challenge ciphertext.

By the construction of the adversary $\mathcal{A}_{3,2}$, for the case of Assumption A, when SubSubGame $\kappa.3.\rho$ and the mode dec (respectively, eval_1 and eval_2) are chosen, $\mathcal{A}_{3,2}$ wins the security game for $\widetilde{\mathbf{H}}$ if and only if the event $R_\ell^{\langle i\rangle(\kappa.3.\rho)}$ for $i=5$ (respectively, $i=6$ and $i=7$) occurs in the internal game. The same holds for $R_\ell^{\langle i\rangle(\kappa.5.\rho)}$ and $R_\ell^{\langle i\rangle(\text{con-3}.\rho)}$ instead of $R_\ell^{\langle i\rangle(\kappa.3.\rho)}$. Therefore, since $X' \setminus L$ is approximately samplable relative to X by the assumption, the same argument as the case of adversary $\mathcal{A}_{3,1}$ implies that the sum of $\Pr[R_\ell^{\langle i\rangle(\kappa.3.\rho)}]$, $\Pr[R_\ell^{\langle i\rangle(\kappa.5.\rho)}]$ and $\Pr[R_\ell^{\langle i\rangle(\text{con-3}.\rho)}]$ for all $i \in \{5,6,7\}$, κ and ρ is negligible.

From now, we consider the case of Assumption U. First we note that, for the cases of SubSubGame $\kappa.3.\rho$ and SubSubGame $\kappa.5.\rho$, when the oracle Hash used in the simulation of the κ-th refreshing process returns $\perp$ (i.e., $x_\kappa \in X \setminus L$), Claim 2 implies that the elements $\lambda_{\mathcal{D}}(\kappa) \cdot x^{*(\kappa)}$ and x_κ are uniformly random over $X' \setminus L$. Claim 2 also implies that whether the oracle Hash returns $\perp$ or not is determined solely from the value $\lambda_{\mathcal{D}}(\kappa)$ and is independent of $x^{*(\kappa)}$. Moreover, Claim 1 implies that the private information on k and $\widehat{k}$ are not used during the internal game except for computing the values $H_k(\lambda_{\mathcal{D}}(\kappa) \cdot x^{*(\kappa)})$ and $\widehat{H}_{\widehat{k}}(x_\kappa)$ in the calculation of e_κ and $\widehat{\pi}_\kappa$, respectively. Therefore, since $\Pi' = \Pi$, $\mathbf{H}$ is smooth relative to (X',Π') and $\widehat{\mathbf{H}}$ is smooth relative to $(X',\widehat{\Pi})$, the distributions of $H_k(\lambda_{\mathcal{D}}(\kappa) \cdot x^{*(\kappa)})$ and $\widehat{H}_{\widehat{k}}(x_\kappa)$ have negligible statistical distances from the uniform distributions over Π and $\widehat{\Pi}$, respectively (note that, for any finite abelian group, the sum of a uniformly random element and any element is uniformly random over the group). Since the differences $e_\kappa - H_k(\lambda_{\mathcal{D}}(\kappa) \cdot x^{*(\kappa)})$ and $\widehat{\pi}_\kappa - \widehat{H}_{\widehat{k}}(x_\kappa)$ are independent of $x^{*(\kappa)}$ by Claim 1, it follows that the distribution of $(x_\kappa, e_\kappa, \widehat{\pi}_\kappa)$ has negligible statistical distance from the uniform distribution of $(x^{**}, e^{**}, \widehat{\pi}^{**})$ over $(X' \setminus L) \times \Pi \times \widehat{\Pi}$. Similarly, for the case of SubGame con-3.ρ, Claim 3 and the smoothness of $\mathbf{H}$ and $\widehat{\mathbf{H}}$ as above imply that the distribution of the first three components $(x^*, e^*, \widehat{\pi}^*)$ of the challenge ciphertext in the internal game has negligible statistical distance from the uniform distribution of $(x^{**}, e^{**}, \widehat{\pi}^{**})$ over $(X' \setminus L) \times \Pi \times \widehat{\Pi}$. This implies that, when the tuple $(x_\kappa, e_\kappa, \widehat{\pi}_\kappa)$ for the cases of SubSubGame $\kappa.3.\rho$ and SubSubGame $\kappa.5.\rho$, or the tuple $(x^*, e^*, \widehat{\pi}^*)$ for the case of SubGame con-3.ρ, is replaced with the $(x^{**}, e^{**}, \widehat{\pi}^{**})$, the differences induced to the advantage of $\mathcal{A}_{3,2}$ and to the probabilities $\Pr[R_\ell^{\langle i\rangle(\kappa.3.\rho)}]$, $\Pr[R_\ell^{\langle i\rangle(\kappa.5.\rho)}]$ and $\Pr[R_\ell^{\langle i\rangle(\text{con-3}.\rho)}]$ by the modification are negligible. By the definition of $\mathcal{A}_{3,2}$, after the modification, the adversary wins the security game for $\widetilde{\mathbf{H}}$ if and only if the event $R_\ell^{\langle i\rangle(\kappa.3.\rho)}$, $R_\ell^{\langle i\rangle(\kappa.5.\rho)}$ or $R_\ell^{\langle i\rangle(\text{con-3}.\rho)}$ corresponding to the choices of the internal game and the mode (dec, eval_1 or eval_2) occurs. Moreover, owing to the assumption that $X' \setminus L$ is approximately samplable relative to X, the uniform distribution over $X' \setminus L$ which is sampled by the adversary can be replaced with an efficiently samplable distribution over X with negligible statistical distance, and the adversary becomes PPT after the replacement and has negligible advantage owing to the first-uniform computationally universal$_2$ property of $\widetilde{\mathbf{H}}$ relative to $X' \times \Pi \times \widehat{\Pi}$. By the results above, the same argument as the case of Assumption A above implies that the sum of $\Pr[R_\ell^{\langle i\rangle(\kappa.3.\rho)}]$, $\Pr[R_\ell^{\langle i\rangle(\kappa.5.\rho)}]$ and $\Pr[R_\ell^{\langle i\rangle(\text{con-3}.\rho)}]$ for all $i \in \{5,6,7\}$, κ and ρ is negligible.

Summarizing, for any of the cases of Assumption A and Assumption U, the sum of $\Pr[R_\ell^{\langle i\rangle(\kappa.3.\rho)}]$, $\Pr[R_\ell^{\langle i\rangle(\kappa.5.\rho)}]$ and $\Pr[R_\ell^{\langle i\rangle(\text{con-3}.\rho)}]$ for all $1 \leq i \leq 7$, κ and ρ is negligible. Hence, $|\delta_3|$ is negligible as well, as desired.

Evaluation of δ_4: Finally, we evaluate the quantity δ_4. By Claim 1 in the evaluation of δ_3 above, in SubGame $\kappa.3$, the private information on the key k for $\mathbf{H}$ is not used during the game except for the computation of $e^{*(\kappa)} = M_\beta^* + H_k(x^{*(\kappa)})$ with $x^{*(\kappa)} \in X' \setminus L$ in the κ-th refreshing process. Therefore, by replacing the value $H_k(x^{*(\kappa)})$ above with $H_k(x^{*(\kappa)}) + \pi^{\dagger\dagger}$ where $\pi^{\dagger\dagger}$ is chosen uniformly at random from Π', only negligible difference is induced to the probability of the event T_ℓ owing to the smoothness of $\mathbf{H}$ relative to (X',Π'). Secondly, since the uniformly random $\pi^{\dagger\dagger} \in \Pi'$ is independent of $M_\beta^* \in \Pi'$, the element $M_\beta^* + \pi^{\dagger\dagger}$ is also uniformly random over Π'. Then, owing to the assumption that Π' is approximately samplable relative to Π, by replacing the value $M_\beta^* + \pi^{\dagger\dagger}$ above further with the element $\pi^\dagger \in \Pi$ chosen as in the definition of SubGame $\kappa.4$, only negligible difference is induced to the probability of the event T_ℓ. Now the resulting choice of $e^{*(\kappa)}$ is the same as in SubGame $\kappa.4$, therefore $|\Pr[T_\ell^{(\kappa.3)}] - \Pr[T_\ell^{(\kappa.4)}]|$ is negligible for any κ. Similarly, in Game con-3, by Claim 3 and the smoothness of $\mathbf{H}$ as above, only negligible difference is induced to the probability of the event T_ℓ when the value $M_\beta^* + H_k(x^*)$ appeared in the computation of the second component e^* of the challenge ciphertext is replaced with $\pi^\dagger + H_k(x^*)$ where $\pi^\dagger \in \Pi$ is chosen as in the definition of Game con-4. Hence, $|\Pr[T_\ell^{(\text{con-3})}] - \Pr[T_\ell^{(\text{con-4})}]|$ is negligible as well. Summarizing, it follows that $|\delta_4|$ is negligible, as desired.

By these results, we have $Adv_{\mathsf{KH\text{-}PKE},\mathcal{A}}^{KH\text{-}CCA}(\ell) \leq |\delta_1| + |\delta_2| + |\delta_3| + |\delta_4|$, while all of $|\delta_1|$, $|\delta_2|$, $|\delta_3|$ and $|\delta_4|$ are negligible, therefore the advantage of the adversary $\mathcal{A}$ for our proposed KH-PKE scheme is negligible as well. This completes the proof of Theorem 4.1. □

5 Instantiations of the Generic Construction

5.1 Cramer–Shoup Projective Hash Family

To instantiate the (computationally or information-theoretically) universal$_2$ hash proof system $\widetilde{\mathbf{P}}$ in our generic construction of KH-PKE given in Section 4, the construction of hash proof systems proposed by Cramer and Shoup [12, §7.43 Theorem 3] based on diverse group systems can be used. Here we recall the definition of the Cramer–Shoup (CS) hash proof system. In fact, we deal with not only the original construction based on an injective function as the internal function, but also its variants (already mentioned in [12]) where the internal function is generalized to more various classes of functions.

The construction of the CS projective hash family [12] is as follows. Let $\mathbf{G} = (\mathcal{H}, X, L, \Pi)$ be a diverse group system, and let $\{g_1, \dots, g_d\}$ be a fixed generating set of L. Let E be a finite set. Moreover, let $\Gamma\colon X \times E \to \{0, \dots, \widetilde{p} - 1\}^n$ be a function, where $\widetilde{p}$ is the smallest prime dividing $|X/L|$ (in the original construction, Γ is supposed to be an injective function; here we consider more general functions Γ). Then the CS projective hash family $\mathbf{H} = (H, K, X \times E, L \times E, \Pi, S, \alpha)$ is constructed as follows:

- We set $K = \mathcal{H}^{n+1}$, and for $\overrightarrow{k} = (k_0, k_1, \dots, k_n) \in K$ and $(x, e) \in X \times E$, the value of H is defined as follows, where we write $\Gamma(x, e) = (\gamma_1, \dots, \gamma_n) = (\gamma_1(x, e), \dots, \gamma_n(x, e))$:

$$H_{\overrightarrow{k}}(x, e) = k_0(x) + \sum_{i=1}^{n} \gamma_i k_i(x) \enspace .$$

- We set $S = \Pi^{(n+1)d}$, and for $\overrightarrow{k} = (k_0, k_1, \dots, k_n) \in K$, the value of α is defined by

$$\alpha(\overrightarrow{k}) = (k_0(g_1), \dots, k_0(g_d), k_1(g_1), \dots, k_1(g_d), \dots, k_n(g_1), \dots, k_n(g_d)) \enspace .$$

Now, given a public information $\overrightarrow{s} = \alpha(\overrightarrow{k})$, an element $(x, e) \in L \times E$ and an expression $x = \sum_{j=1}^{d} \omega_j g_j$ of x with the generating set $\{g_1, \dots, g_d\}$ of L (which is a witness of $(x, e) \in L \times E$), the private evaluation algorithm for the corresponding hash proof system can compute the value of H by

$$H_{\overrightarrow{k}}(x, e) = \sum_{j=1}^{d} \omega_j s_{0,j} + \sum_{i=1}^{n} \gamma_i(x, e) \sum_{j=1}^{d} \omega_j s_{i,j} \enspace ,$$

where $s_{i,j} = k_i(g_j)$ for $i \in \{0, 1, \dots, n\}$ and $j \in \{1, \dots, d\}$.

The following lemma is the key property for our argument below. We note that essentially the same argument appeared in [12]; here we include the proof for the sake of completeness.

Lemma 5.1. *For the CS projective hash family constructed as above, for $\overrightarrow{s} \in S$, $(x, e), (x^*, e^*) \in (X \setminus L) \times E$ and $\pi, \pi^* \in \Pi$, if $\Gamma(x, e) \neq \Gamma(x^*, e^*)$, then we have*

$$\Pr_{\overrightarrow{k} \xleftarrow{\$} K}[H_{\overrightarrow{k}}(x, e) = \pi \wedge H_{\overrightarrow{k}}(x^*, e^*) = \pi^* \wedge \alpha(\overrightarrow{k}) = \overrightarrow{s}] \leq \frac{1}{\widetilde{p}} \cdot \Pr_{\overrightarrow{k} \xleftarrow{\$} K}[H_{\overrightarrow{k}}(x^*, e^*) = \pi^* \wedge \alpha(\overrightarrow{k}) = \overrightarrow{s}] \enspace .$$

Proof. Since $\Gamma(x,e) \neq \Gamma(x^*,e^*)$, by symmetry, we may assume without loss of generality that $\gamma_n(x,e) \neq \gamma_n(x^*,e^*)$. Now the left-hand side of the inequality in the statement is equal to

$$\sum_{\overrightarrow{k} \in K(\overrightarrow{s})} \frac{1}{|\mathcal{H}|^{n+1}} \cdot \chi[k_0(x) + \gamma_n(x,e)k_n(x) = \overline{\pi} \wedge k_0(x^*) + \gamma_n(x^*,e^*)k_n(x^*) = \overline{\pi}^*] \enspace ,$$

where $K(\overrightarrow{s})$ denotes the set of all $\overrightarrow{k} \in K$ satisfying that $\alpha(\overrightarrow{k}) = \overrightarrow{s}$, we put $\overline{\pi} = \pi - \sum_{i=1}^{n-1} \gamma_i(x,e)k_i(x)$ and $\overline{\pi}^* = \pi^* - \sum_{i=1}^{n-1} \gamma_i(x^*,e^*)k_i(x^*)$, and $\chi[\cdot]$ denotes the characteristic function that returns 1 if the specified condition is satisfied and returns 0 otherwise. Similarly, the right-hand side of the inequality in the statement is equal to

$$\frac{1}{\widetilde{p}} \cdot \sum_{\overrightarrow{k} \in K(\overrightarrow{s})} \frac{1}{|\mathcal{H}|^{n+1}} \cdot \chi[k_0(x^*) + \gamma_n(x^*,e^*)k_n(x^*) = \overline{\pi}^*] \enspace .$$

Therefore, it suffices to show that, for any $k_1, \dots, k_{n-1} \in \mathcal{H}$, we have

$$\begin{aligned} &\sum_{(k_0,k_n) \in K'} \chi[k_0(x) + \gamma_n(x,e)k_n(x) = \overline{\pi} \wedge k_0(x^*) + \gamma_n(x^*,e^*)k_n(x^*) = \overline{\pi}^*] \\ &\qquad \leq \frac{1}{\widetilde{p}} \cdot \sum_{(k_0,k_n) \in K'} \chi[k_0(x^*) + \gamma_n(x^*,e^*)k_n(x^*) = \overline{\pi}^*] \enspace , \end{aligned}$$

where K' denotes the set of all $(k_0,k_n) \in \mathcal{H}^2$ satisfying that $k_i(g_j) = s_{i,j}$ for any $i \in \{0,n\}$ and $j \in \{1,\dots,d\}$.

The inequality above becomes trivial if $K' = \emptyset$; from now, we suppose that $K' \neq \emptyset$. We take an element (k_0^*, k_n^*) of K'. Let A denote the subgroup of $\mathcal{H}$ consisting of homomorphisms $\psi \colon X \to \Pi$ satisfying that $\psi(a) = 0$ for all $a \in L$. Then any element of K' is uniquely expressed as $(k_0^* + \psi_0, k_n^* + \psi_n)$ with $\psi_0, \psi_n \in A$. Moreover, we take an element ψ_* of A satisfying that $\psi_*(x) \neq 0$, which exists since the group system $\mathbf{G}$ is diverse and $x \notin L$. Let $\mathrm{ord}(\psi_*)$ denote the order of the group element $\psi_* \in A$. Then there exist elements $\psi_1, \dots, \psi_\ell \in A$ with $\ell = |A|/\mathrm{ord}(\psi_*)$ (namely, the representative elements of the cosets in the quotient group of A by the subgroup generated by ψ_*) satisfying that any element of A is uniquely expressed as $\psi_i + a\psi_*$ with $i \in \{1,\dots,\ell\}$ and $a \in \{0,1,\dots,\mathrm{ord}(\psi_*)-1\}$. Now if $k_0 = k_0^* + \psi_{i_0} + a_0\psi_*$ and $k_n = k_n^* + \psi_{i_n} + a_n\psi_*$, then we have

$$\begin{aligned} k_0(x) + \gamma_n(x,e)k_n(x) &= k_0^*(x) + \psi_{i_0}(x) + \gamma_n(x,e)(k_n^*(x) + \psi_{i_n}(x)) + (a_0 + \gamma_n(x,e)a_n)\psi_*(x) \enspace , \\ k_0(x^*) + \gamma_n(x^*,e^*)k_n(x^*) &= k_0^*(x^*) + \psi_{i_0}(x^*) + \gamma_n(x^*,e^*)(k_n^*(x^*) + \psi_{i_n}(x^*)) + (a_0 + \gamma_n(x,e)a_n)\psi_*(x^*) \enspace . \end{aligned}$$

Therefore, it suffices to show that, for any $k_1, \dots, k_{n-1} \in \mathcal{H}$ and any $i_0, i_n \in \{1,\dots,\ell\}$, we have

$$\begin{aligned} &\sum_{a_0,a_n=0}^{\mathrm{ord}(\psi_*)-1} \chi[(a_0 + \gamma_n(x,e)a_n)\psi_*(x) = \pi' \wedge (a_0 + \gamma_n(x^*,e^*)a_n)\psi_*(x^*) = \pi'^*] \\ &\qquad \leq \frac{1}{\widetilde{p}} \cdot \sum_{a_0,a_n=0}^{\mathrm{ord}(\psi_*)-1} \chi[(a_0 + \gamma_n(x^*,e^*)a_n)\psi_*(x^*) = \pi'^*] \enspace , \end{aligned}$$

where we put

$$\begin{aligned} \pi' &= \overline{\pi} - k_0^*(x) - \psi_{i_0}(x) - \gamma_n(x,e)(k_n^*(x) + \psi_{i_n}(x)) \enspace , \\ \pi'^* &= \overline{\pi}^* - k_0^*(x^*) - \psi_{i_0}(x^*) - \gamma_n(x^*,e^*)(k_n^*(x^*) + \psi_{i_n}(x^*)) \enspace . \end{aligned}$$

We show that $\psi_*(a \cdot x) \neq 0$ for any integer $a \neq 0$ with $|a| < \widetilde{p}$. First, a is coprime to $|X/L|$ by the definition of $\widetilde{p}$, therefore we have $b_1 a = b_2|X/L| + 1$ for some integers b_1, b_2. Now we have $\psi_*(b_2|X/L| \cdot x) = 0$ since $|X/L| \cdot x \in L$ (note that the order of the image of x in the quotient group X/L is a divisor of $|X/L|$), while $\psi_*(x) \neq 0$ by the choice of ψ_*. This implies that $\psi_*(b_1 a \cdot x) \neq 0$, therefore $\psi_*(a \cdot x) \neq 0$, as desired.

The previous paragraph implies that $\mathrm{ord}(\psi_*) \geq \widetilde{p}$, since $\psi_*(\mathrm{ord}(\psi_*) \cdot x) = (\mathrm{ord}(\psi_*) \cdot \psi_*)(x) = 0$. Now, since $\gamma_n(x,e), \gamma_n(x^*,e^*) \in \{0,1,\ldots,\widetilde{p}-1\}$ and $\gamma_n(x,e) \neq \gamma_n(x^*,e^*)$, the matrix $\begin{pmatrix} 1 & \gamma_n(x,e) \\ 1 & \gamma_n(x^*,e^*) \end{pmatrix}$ is non-singular, where the components are considered modulo $\mathrm{ord}(\psi_*)$. This implies that, when a_0 and a_n run over $\{0,1,\ldots,\mathrm{ord}(\psi_*)-1\}$, the pair of $(a_0 + \gamma_n(x,e)a_n \bmod \mathrm{ord}(\psi_*))$ and $(a_0 + \gamma_n(x^*,e^*)a_n \bmod \mathrm{ord}(\psi_*))$ distributes uniformly on $\{0,1,\ldots,\mathrm{ord}(\psi_*)-1\}^2$. Therefore, it suffices to show that, for any $k_1,\ldots,k_{n-1} \in \mathcal{H}$ and any $i_0, i_n \in \{1,\ldots,\ell\}$, we have

$$\sum_{a,a'=0}^{\mathrm{ord}(\psi_*)-1} \chi[a\psi_*(x) = \pi' \wedge a'\psi_*(x^*) = \pi'^*] \leq \frac{1}{\widetilde{p}} \cdot \sum_{a,a'=0}^{\mathrm{ord}(\psi_*)-1} \chi[a'\psi_*(x^*) = \pi'^*] \ .$$

Now we note that, the condition $a\psi_*(x) = \pi'$ is satisfied by at most $\mathrm{ord}(\psi_*)/\widetilde{p}$ integers $a \in \{0,1,\ldots,\mathrm{ord}(\psi_*)-1\}$. Indeed, if the number of such a is larger than $\mathrm{ord}(\psi_*)/\widetilde{p}$, then the pigeonhole principle implies that we have $a_1\psi_*(x) = a_2\psi_*(x) = \pi'$ for some integers $a_1 < a_2$ with $a_2 - a_1 < \widetilde{p}$. However, this implies that $\psi_*((a_2 - a_1) \cdot x) = (a_2 - a_1)\psi_*(x) = 0$, contradicting the previous paragraph. Hence, we have

$$\begin{aligned} \sum_{a,a'=0}^{\mathrm{ord}(\psi_*)-1} \chi[a\psi_*(x) = \pi' \wedge a'\psi_*(x^*) = \pi'^*] &\leq \sum_{a'=0}^{\mathrm{ord}(\psi_*)-1} \frac{\mathrm{ord}(\psi_*)}{\widetilde{p}} \chi[a'\psi_*(x^*) = \pi'^*] \\ &= \frac{1}{\widetilde{p}} \cdot \sum_{a,a'=0}^{\mathrm{ord}(\psi_*)-1} \chi[a'\psi_*(x^*) = \pi'^*] \ , \end{aligned}$$

as desired. This completes the proof of Lemma 5.1. □

Owing to Lemma 5.1, we will show that the CS projective hash family is (information-theoretically or computationally) universal$_2$, if the internal function Γ satisfies some appropriate property. First, we recall the notions of collision resistant (CR) hash family and target collision resistant (TCR) hash family.

Definition 5.1 (Collision Resistant Hash Family). *Let $\{f_{hk} \mid hk \in \mathcal{HK}\}$ be a family of hash functions $f_{hk} \colon \mathcal{X} \to \mathcal{Y}$ indexed by a hash key $hk \in \mathcal{HK}$. We say that the family is* collision resistant *(*CR*), if for any PPT adversary $\mathcal{A}$, its advantage $Adv^{\mathsf{CR}}_{\mathcal{A}}(\ell)$ defined by*

$$Adv^{\mathsf{CR}}_{\mathcal{A}}(\ell) = \Pr[hk \xleftarrow{\$} \mathcal{HK}; (x,x^*) \leftarrow \mathcal{A}(1^\ell, hk) \colon x \neq x^* \wedge f_{hk}(x) = f_{hk}(x^*)]$$

is negligible in the security parameter ℓ.

Definition 5.2 (Target Collision Resistant Hash Family). *Let $\{f_{hk} \mid hk \in \mathcal{HK}\}$ be a family of hash functions $f_{hk} \colon \mathcal{X} \to \mathcal{Y}$ indexed by a hash key $hk \in \mathcal{HK}$. Let $\mathcal{X}' \subset \mathcal{X}$. We say that the family is* target collision resistant *(*TCR*) relative to $\mathcal{X}'$, if for any PPT adversary $\mathcal{A}$, its advantage $Adv^{\mathsf{TCR}}_{\mathcal{A}}(\ell)$ defined by*

$$Adv^{\mathsf{TCR}}_{\mathcal{A}}(\ell) = \Pr[x^* \xleftarrow{\$} \mathcal{X}'; hk \xleftarrow{\$} \mathcal{HK}; x \leftarrow \mathcal{A}(1^\ell, hk, x^*) \colon x \neq x^* \wedge f_{hk}(x) = f_{hk}(x^*)]$$

is negligible in the security parameter ℓ. When $\mathcal{X}' = \mathcal{X}$, we simply say that the family of hash functions is target collision resistant*.*

On the other hand, we also introduce a simple but useful technique to improve the efficiency; we can "compress" the output of the CS projective hash family by using a "smooth" function. The smoothnessof a function defined below is a statistical property that roughly ensures that the "min-entropy" of the output of the function (for uniformly random input) is sufficiently high, and thus it is information-theoretically hard to guess the output.[2] The definition is as follows.

[2] Note that this notion is (somewhat similar to but) different from the smoothness of a projective hash family.

Definition 5.3 (Smooth Function)**.** *Let $f : \mathcal{X} \rightarrow \mathcal{Y}$ be a hash function. We say that f is* ϵ-smooth*, if the quantity* $\mathsf{Smth}_f := \max_{y \in \mathcal{Y}} \Pr_{x \xleftarrow{\$} \mathcal{X}} [f(x) = y]$ *is not larger than ϵ. We say that f is* smooth*, if it is ϵ-smooth for a negligible ϵ.*

We note that, besides the injective functions (with superpolynomially large domain) which are trivially smooth, the smoothness is in fact satisfied by several famous cryptographic functions such as OWFs, always second-preimage resistant (aSec secure) hash functions [34], and KDFs [14]; see Appendix A. Interestingly, the universal$_2$ property of the CS projective hash family is preserved by hashing its output to be a shorter element. Intuitively, for the CS projective hash family, the bound of the advantage of adversaries for the universal$_2$ property is closely related to the parameter for the underlying subset membership problem $\mathbf{M}$, therefore the bound cannot be freely selected (e.g., the order of the group should be larger than a certain threshold relevant to the desired security level) since $\mathbf{M}$ must be hard. Our proposed technique provides a way to reduce the output size of the projective hash family, while the too strong bound of the universal$_2$ advantage is increased but is still reasonably strong. It is a bit surprising that this technique can be also applied to the original Cramer–Shoup scheme, but to the best of our knowledge, it has never explicitly been stated in the literature. When applying our technique to the Cramer–Shoup scheme, ciphertext length of the resulting scheme becomes the same as that of the Kurosawa–Desmedt (KD) scheme [29] which is the best known DDH-based PKE scheme. We should also note that this technique is not applicable to other similar schemes such as the Cash–Kiltz–Shoup [9], Hanaoka–Kurosawa [22], and Kiltz schemes [27]. This fact is primarily due to the structure of HPS-based constructions, and thus, it is difficult to apply the above technique to PKE schemes from other methodology, e.g. [7, 22, 26].

We describe the technique discussed above. For any projective hash family $\mathbf{H}$ and any smooth function f with domain including Π, we define the composition $f \circ \mathbf{H}$ to be the projective hash family obtained from $\mathbf{H}$ by taking the composition $f \circ H_k$ for the function H_k, $k \in K$. We will show that, for the case that $\mathbf{H}$ is the CS projective hash family, $f \circ \mathbf{H}$ is (information-theoretically or computationally) universal$_2$ provided some appropriate conditions are satisfied. For the purpose, we require a trapdoor property for the underlying subset membership problem, formalized as follows.

Definition 5.4 (Trapdoor Subset Membership Problem)**.** *We say that a subset membership problem $\mathbf{M}$ is a* trapdoor subset membership problem*, if it is endowed with an additional* trapdoor mode *as well as the ordinary mode, satisfying the followings: (1) In the trapdoor mode, the instance sampling algorithm takes as input 1^ℓ and returns $\Lambda = \Lambda[X, X', L, W, R] \in [I_\ell]$ and a trapdoor element τ, where the distribution of Λ in the trapdoor mode is identical to that of Λ in the ordinary mode; (2) there exists a PPT algorithm that takes the trapdoor τ and an element $x \in X$ as input and decides whether $x \in L$ or not.*

We say that a trapdoor subset membership problem $\mathbf{M}$ is hard (relative to X'), if it is hard (relative to X') in the ordinary mode as a subset membership problem.

Based on the definitions above, we give the following result:

Proposition 5.1. *Let $\mathbf{H}$ be the CS projective hash family constructed as above. Let $f \colon \Pi \rightarrow \mathcal{Y}$ be an ϵ-smooth hash function.*

1. *If the function Γ is injective, then $f \circ \mathbf{H}$ is $(|\Pi|\epsilon/\widetilde{p})$-universal$_2$.*

2. *If Γ is sampled from a CR hash family, the subset membership problem associated to $\mathbf{H}$ is a trapdoor subset membership problem, and $|\Pi|\epsilon/\widetilde{p}$ is negligible, then $f \circ \mathbf{H}$ is first-adaptive computationally universal$_2$.*

3. *If Γ is sampled from a TCR hash family relative to a subset $X' \times E \subset X \times E$, the subset membership problem associated to $\mathbf{H}$ is a trapdoor subset membership problem, and $|\Pi|\epsilon/\widetilde{p}$ and $|X' \cap L|/|X'|$ are negligible, then $f \circ \mathbf{H}$ is first-uniform computationally universal$_2$ relative to $X' \times E$.*

Proof. For the first part, let $(x,e),(x^*,e^*) \in (X \setminus L) \times E$ with $(x,e) \neq (x^*,e^*)$, let $\overrightarrow{s} \in S$, and let $y, y^* \in \mathcal{Y}$. Then we have

$$\begin{aligned}
&\Pr_{\overrightarrow{k} \xleftarrow{\$} K}[f \circ H_{\overrightarrow{k}}(x,e) = y \wedge f \circ H_{\overrightarrow{k}}(x^*,e^*) = y^* \wedge \alpha(\overrightarrow{k}) = \overrightarrow{s}] \\
&= \sum_{\pi \in f^{-1}(y), \pi^* \in f^{-1}(y^*)} \Pr_{\overrightarrow{k} \xleftarrow{\$} K}[H_{\overrightarrow{k}}(x,e) = \pi \wedge H_{\overrightarrow{k}}(x^*,e^*) = \pi^* \wedge \alpha(\overrightarrow{k}) = \overrightarrow{s}] \enspace .
\end{aligned}$$

Since Γ is injective, we have $\Gamma(x,e) \neq \Gamma(x^*,e^*)$, therefore Lemma 5.1 implies that the right-hand side of the last equality is not larger than

$$\begin{aligned}
&\sum_{\pi \in f^{-1}(y), \pi^* \in f^{-1}(y^*)} \frac{1}{\widetilde{p}} \cdot \Pr_{\overrightarrow{k} \xleftarrow{\$} K}[H_{\overrightarrow{k}}(x^*,e^*) = \pi^* \wedge \alpha(\overrightarrow{k}) = \overrightarrow{s}] \\
&= \frac{|f^{-1}(y)|}{\widetilde{p}} \cdot \Pr_{\overrightarrow{k} \xleftarrow{\$} K}[f \circ H_{\overrightarrow{k}}(x^*,e^*) = y^* \wedge \alpha(\overrightarrow{k}) = \overrightarrow{s}] \enspace .
\end{aligned}$$

This implies that $f \circ \mathbf{H}$ is $(\max_{y \in \mathcal{Y}} |f^{-1}(y)|/\widetilde{p})$-universal$_2$. Moreover, we have $\mathsf{Smth}_f = \max_{y \in \mathcal{Y}} |f^{-1}(y)|/|\Pi|$, therefore $\max_{y \in \mathcal{Y}} |f^{-1}(y)| \leq |\Pi|\epsilon$ since f is ϵ-smooth. Hence $f \circ \mathbf{H}$ is $(|\Pi|\epsilon/\widetilde{p})$-universal$_2$, as desired.

For the second part of the claim, let $\mathcal{A}$ be an adversary for the first-adaptive computationally universal$_2$ game of $f \circ \mathbf{H}$. Let τ denote the trapdoor element for the subset membership problem associated to $\mathbf{H}$ generated in its trapdoor mode. Then we construct an adversary $\mathcal{A}^\dagger$ for the CR property for Γ in the following manner. The adversary $\mathcal{A}^\dagger$ first generates the key $\overrightarrow{k} \in K$ uniformly at random, computes $\overrightarrow{s} = \alpha(\overrightarrow{k})$, and then executes the adversary $\mathcal{A}$ with input $\overrightarrow{s}$. In the simulation of the first-adaptive computationally universal$_2$ game, $\mathcal{A}^\dagger$ emulates the oracle Hash by using the trapdoor element τ (to efficiently decide whether the query (x,e) is in $L \times E$ or not) and the key $\overrightarrow{k}$ (to compute the reply $H_{\overrightarrow{k}}(x,e)$ to the query). Similarly, given an element $(x^*,e^*) \in X \times E$ submitted by $\mathcal{A}$, $\mathcal{A}^\dagger$ computes the reply $y^* = f \circ H_{\overrightarrow{k}}(x^*,e^*)$ by using $\overrightarrow{k}$. Finally, $\mathcal{A}^\dagger$ receives an output $((x,e),y) \in (X \times E) \times \mathcal{Y}$ of $\mathcal{A}$, and outputs (x,e) and (x^*,e^*). Now let T (respectively, T') denote the event that $\Gamma(x,e) = \Gamma(x^*,e^*)$ (respectively, $\Gamma(x,e) \neq \Gamma(x^*,e^*)$) and $\mathcal{A}$ wins the first-adaptive computationally universal$_2$ game. Then we have $Adv_{\mathcal{A}}^{\mathsf{AComp.Univ}_2}(\ell) = \Pr[T] + \Pr[T']$ and $\Pr[T] \leq Adv_{\mathcal{A}^\dagger}^{\mathsf{CR}}(\ell)$. Moreover, the same argument as the previous paragraph based on Lemma 5.1 implies that

$$\Pr[T'] = \Pr[\Gamma(x,e) \neq \Gamma(x^*,e^*) \wedge H_{\overrightarrow{k}}(x,e) = y \mid H_{\overrightarrow{k}}(x^*,e^*) = y^* \wedge \alpha(\overrightarrow{k}) = \overrightarrow{s}] \leq \frac{|\Pi|\epsilon}{\widetilde{p}} \enspace ,$$

which is negligible by the assumption. Hence, $Adv_{\mathcal{A}^\dagger}^{\mathsf{CR}}(\ell)$ is non-negligible provided $Adv_{\mathcal{A}}^{\mathsf{AComp.Univ}_2}(\ell)$ is non-negligible. This completes the proof of the second part of the claim.

Similarly, for the third part of the claim, let $\mathcal{A}$ be an adversary for the first-uniform computationally universal$_2$ game of $f \circ \mathbf{H}$ relative to $X' \times E$. Let τ denote the trapdoor element for the subset membership problem associated to $\mathbf{H}$ generated in its trapdoor mode. Then we construct an adversary $\mathcal{A}^\dagger$ for the TCR property for Γ relative to $X' \times E$ in the following manner. Given an input $(x^*,e^*) \in X' \times E$, the adversary $\mathcal{A}^\dagger$ first generates the key $\overrightarrow{k} \in K$ uniformly at random, computes $\overrightarrow{s} = \alpha(\overrightarrow{k})$, and then executes the adversary $\mathcal{A}$ with input (x^*,e^*), $\overrightarrow{s}$ and $y^* = f \circ H_{\overrightarrow{k}}(x^*,e^*)$. Here $\mathcal{A}^\dagger$ efficiently simulates the first-uniform computationally universal$_2$ game by using τ and $\overrightarrow{k}$ in the same manner as the previous paragraph. Finally, $\mathcal{A}^\dagger$ receives an output $((x,e),y) \in (X \times E) \times \mathcal{Y}$ of $\mathcal{A}$, and outputs (x,e) and (x^*,e^*). We define the events T and T' in the same manner as in the previous paragraph. Moreover, let T_0 and T'_0 denote the same events as T and T', respectively, except that the input (x^*,e^*) for $\mathcal{A}^\dagger$ is chosen uniformly at random from $(X' \setminus L) \times E$ instead of $X' \times E$. Then we have $Adv_{\mathcal{A}}^{\mathsf{UComp.Univ}_2}(\ell) = \Pr[T_0] + \Pr[T'_0]$. On the other hand, by the assumption that $|X' \cap L|/|X'|$ is negligible, it follows that the uniform distributions on $X' \times E$ and on $(X' \setminus L) \times E$ have negligible statistical distance, therefore $|\Pr[T] - \Pr[T_0]|$ is negligible. Moreover, we have $\Pr[T] \leq Adv_{\mathcal{A}^\dagger}^{\mathsf{TCR}}(\ell)$, while the same argument as the previous paragraph implies that $\Pr[T'_0] \leq |\Pi|\epsilon/\widetilde{p}$, which is negligible by the assumption. Hence, $Adv_{\mathcal{A}^\dagger}^{\mathsf{TCR}}(\ell)$ is non-negligible provided $Adv_{\mathcal{A}}^{\mathsf{UComp.Univ}_2}(\ell)$ is non-negligible. This completes the proof of Proposition 5.1. □

5.2 Instantiation of KH-PKE from Diverse Group Systems

Here we give an instantiation of our generic construction of KH-PKE schemes proposed in Section 4, based on a general diverse group system and the corresponding CS projective hash family. Let $\mathbf{G} = (\mathcal{H}, X, L, \Pi)$ be a diverse group system for which the associated subset membership problem is hard. Let $\widetilde{p}$ denote the smallest prime dividing $|X/L|$. Suppose that $\epsilon' := 1/\widetilde{p}$ and $\epsilon := (\epsilon'|\Pi| - 1)(|\Pi| - 1)/2$ are both negligible. In the setting, we define the three hash proof systems $\mathbf{P}$, $\widehat{\mathbf{P}}$ and $\widetilde{\mathbf{P}}$ used by our generic construction in the following manner:

- We set $\mathbf{H}$ to be the homomorphic projective hash family constructed from $\mathbf{G}$ as mentioned in Section 2, and set $\mathbf{P}$ to be the corresponding hash proof system. Then $\mathbf{P}$ is ϵ-smooth by Lemma 2.3 and Lemma 2.2, and ϵ is negligible as above.

- We set $\widehat{\mathbf{H}} = \mathbf{H}$ and $\widehat{\mathbf{P}} = \mathbf{P}$. Then $\widehat{\mathbf{P}}$ is homomorphic and ϵ'-universal$_1$ by Lemma 2.3, and ϵ' is negligible as above.

- Let $f\colon \Pi \to \mathcal{Y}$ be an ϵ_{smth}-smooth function, and suppose that $\epsilon'' := |\Pi|\epsilon_{\mathsf{smth}}/\widetilde{p}$ is negligible. We put $E := \Pi^2$, and let $\Gamma\colon X \times E \to \{0, 1, \dots, \widetilde{p} - 1\}^n$ be any injective function. We set $\widetilde{\mathbf{H}}$ to be the composition of f and the CS projective hash family constructed from the diverse group system $\mathbf{G}$ and the internal function Γ, and set $\widetilde{\mathbf{P}}$ to be the corresponding hash proof system. Then $\widetilde{\mathbf{P}}$ is (information-theoretically) ϵ''-universal$_2$ by Lemma 2.3 and Proposition 5.1, and ϵ'' is negligible as above.

Then the conditions of Theorem 4.1 with Assumption I are satisfied, therefore the resulting instantiation of our KH-PKE scheme is KH-CCA secure.

For example, we can use any $\mathbf{G}$ satisfying that $|\Pi| = |X/L|$ and it is an exponentially large prime $\widetilde{p}$, and the identity map $\Pi \to \Pi$ as the smooth function f. Then we have $\epsilon = 0$, $\epsilon' = 1/\widetilde{p}$, $\epsilon_{\mathsf{smth}} = 1/|\Pi|$ and $\epsilon'' = 1/\widetilde{p}$, therefore all of ϵ, ϵ' and ϵ'' are negligible, as desired.

5.3 DDH-Based Instantiation of KH-PKE

From now, we give instantiations of our KH-PKE schemes based on some standard computational assumptions. First, we describe the instantiation based on the DDH assumption. We recall the definition of the DDH assumption.

Definition 5.5 (The Decisional Diffie–Hellman (DDH) Assumption)**.** *Let $\mathbb{G}$ be a multiplicative cyclic group of prime order p. We say that* the DDH assumption holds in $\mathbb{G}$, *if for any PPT algorithm $\mathcal{A}$, the advantage $Adv^{DDH}_{\mathbb{G},\mathcal{A}}(\ell) := |\Pr[\mathcal{A}(g_0, g_1, g_0^r, g_1^r) = 0] - \Pr[\mathcal{A}(g_0, g_1, g_0^r, g_1^{r'}) = 0]|$ is negligible, where g_0 and g_1 are chosen from $\mathbb{G}$ uniformly at random, and r and r' are chosen from $\mathbb{Z}_p$ uniformly at random.*

In order to construct the DDH-based instantiation, we define a trapdoor subset membership problem $\mathbf{M}$ and a diverse group system $\mathbf{G}$ in the following manner. Let $\mathbb{G}$ be a cyclic group of prime order p for which the DDH assumption holds. In particular, $1/p$ is negligible (since otherwise the DDH assumption is not satisfied), therefore the uniform distributions on $\mathbb{G}$ and on $\mathbb{G} \setminus \{1\}$ have negligible statistical distance.

- The instance sampling algorithm for $\mathbf{M}$ chooses two generators $g_0, g_1 \in \mathbb{G} \setminus \{1\}$ of $\mathbb{G}$ uniformly at random, sets $X := \mathbb{G}^2$, $L := \{(g_0^i, g_1^i) \in X \mid i \in \mathbb{Z}_p\} \simeq \mathbb{G}$ which is generated by (g_0, g_1), $W := \mathbb{Z}_p$, and defines the relation R in such a way that, for $(x_0, x_1) \in X$ and $\omega \in W$, we have $((x_0, x_1), \omega) \in R$ if and only if $x_0 = g_0^\omega$ and $x_1 = g_1^\omega$. On the other hand, the subset sampling algorithm first chooses $\omega \in W$ uniformly at random, and then outputs $(g_0^\omega, g_1^\omega) \in L$ and the $\omega \in W$. The construction of $\mathbf{M}$ satisfies the condition for a hard subset membership problem, where the hardness follows immediately from the DDH assumption on $\mathbb{G}$.

- In the trapdoor mode for $\mathbf{M}$, the algorithm chooses the g_0 and g_1 above in such a way that g_0 is chosen first; secondly $\tau \in \mathbb{Z}_p \setminus \{0\}$ is chosen uniformly at random, which is the trapdoor element; and then g_1

```
KeyGen$(1^\ell)$ :
  $hk \xleftarrow{\$} \mathcal{HK}$;  $g_0, g_1 \xleftarrow{\$} \mathbb{G}$
  $k_0, k_1, \widehat{k}_0, \widehat{k}_1, \widetilde{k}_{0,0}, \widetilde{k}_{0,1}, \widetilde{k}_{1,0}, \widetilde{k}_{1,1} \xleftarrow{\$} \mathbb{Z}_p$
  $s \leftarrow g_0^{k_0} g_1^{k_1}$;  $\widehat{s} \leftarrow g_0^{\widehat{k}_0} g_1^{\widehat{k}_1}$
  $\widetilde{s}_0 \leftarrow g_0^{\widetilde{k}_{0,0}} g_1^{\widetilde{k}_{0,1}}$;  $\widetilde{s}_1 \leftarrow g_0^{\widetilde{k}_{1,0}} g_1^{\widetilde{k}_{1,1}}$
  $pk \leftarrow (hk, f, g_0, g_1, s, \widehat{s}, \widetilde{s}_0, \widetilde{s}_1)$
  $sk_d \leftarrow (k_0, k_1, \widehat{k}_0, \widehat{k}_1, \widetilde{k}_{0,0}, \widetilde{k}_{0,1}, \widetilde{k}_{1,0}, \widetilde{k}_{1,1})$
  $sk_h \leftarrow (\widetilde{k}_{0,0}, \widetilde{k}_{0,1}, \widetilde{k}_{1,0}, \widetilde{k}_{1,1})$
  Return $(pk, sk_d, sk_h)$
```

```
Dec$(sk_d, C)$ :
  Parse $C$ as $(x_0, x_1, e, \widehat{\pi}, y)$
  $\widehat{\pi}' \leftarrow x_0^{\widehat{k}_0} x_1^{\widehat{k}_1}$
  $\gamma \leftarrow \Gamma_{hk}(x_0, x_1, e, \widehat{\pi})$
  $y' \leftarrow f(x_0^{\widetilde{k}_{0,0}+\gamma\widetilde{k}_{1,0}} x_1^{\widetilde{k}_{0,1}+\gamma\widetilde{k}_{1,1}})$
  If either $\widehat{\pi} \neq \widehat{\pi}'$ or $y \neq y'$ then
     return $\perp$
  $\pi \leftarrow x_0^{k_0} x_1^{k_1}$
  Return $M \leftarrow e/\pi$
```

```
Enc$(pk, M)$ (for $M \in \mathcal{M} := \mathbb{G}$) :
  $\omega \xleftarrow{\$} \mathbb{Z}_p$;  $x_0 \leftarrow g_0^\omega$;  $x_1 \leftarrow g_1^\omega$
  $\pi \leftarrow s^\omega$;  $e \leftarrow M \cdot \pi$
  $\widehat{\pi} \leftarrow \widehat{s}^\omega$
  $\gamma \leftarrow \Gamma_{hk}(x_0, x_1, e, \widehat{\pi})$
  $y \leftarrow f((\widetilde{s}_0 \cdot \widetilde{s}_1^\gamma)^\omega)$
  Return $C \leftarrow (x_0, x_1, e, \widehat{\pi}, y)$
```

```
Eval$(sk_h, C_1, C_2)$ :
  Parse $C_b$ as $(x_{b,0}, x_{b,1}, e_b, \widehat{\pi}_b, y_b)$ for $b = 1, 2$
  $\gamma_b \leftarrow \Gamma_{hk}(x_{b,0}, x_{b,1}, e_b, \widehat{\pi}_b)$ for $b = 1, 2$
  $y'_b \leftarrow f(x_{b,0}^{\widetilde{k}_{0,0}+\gamma_b\widetilde{k}_{1,0}} x_{b,1}^{\widetilde{k}_{0,1}+\gamma_b\widetilde{k}_{1,1}})$ for $b = 1, 2$
  If either $y_1 \neq y'_1$ or $y_2 \neq y'_2$ then
     return $\perp$
  $\omega \xleftarrow{\$} \mathbb{Z}_p$
  $x_0 \leftarrow x_{1,0} x_{2,0} g_0^\omega$;  $x_1 \leftarrow x_{1,1} x_{2,1} g_1^\omega$
  $e \leftarrow e_1 e_2 s^\omega$;  $\widehat{\pi} \leftarrow \widehat{\pi}_1 \widehat{\pi}_2 \widehat{s}^\omega$
  $\gamma \leftarrow \Gamma_{hk}(x_0, x_1, e, \widehat{\pi})$
  $y \leftarrow f(x_0^{\widetilde{k}_{0,0}+\gamma\widetilde{k}_{1,0}} x_1^{\widetilde{k}_{0,1}+\gamma\widetilde{k}_{1,1}})$
  Return $C \leftarrow (x_0, x_1, e, \widehat{\pi}, y)$
```

Figure 2: DDH-based instantiation of our KH-PKE scheme (here $\mathbb{G}$ is a cyclic group of prime order p satisfying the DDH assumption; $\{\Gamma = \Gamma_{hk} \colon \mathbb{G}^4 \to \{0, 1, \ldots, p-1\} \mid hk \in \mathcal{HK}\}$ is a TCR hash family; and $f \colon \mathbb{G} \to \mathcal{Y}$ is a smooth function)

is defined by $g_1 := g_0^\tau$. Then, by using τ, it can be efficiently decided whether a given $(x_0, x_1) \in X$ is in L or not, by checking if $x_1 = x_0^\tau$. Hence, $\mathbf{M}$ is a hard trapdoor subset membership problem.

- To define the corresponding diverse group system $\mathbf{G}$, we set $\Pi := \mathbb{G}$, and define $\mathcal{H}$ to be the set of homomorphisms $H_{k_0,k_1} \colon X \to \Pi$, indexed by $(k_0, k_1) \in \mathbb{Z}_p^2$, satisfying that $H_{k_0,k_1}(x_0, x_1) := x_0^{k_0} x_1^{k_1}$ for any $(x_0, x_1) \in X$. Then $\mathbf{G}$ is diverse; indeed, for any $(x_0, x_1) = (g_0^i, g_1^j) \in X$, by putting $g_1 = g_0^\tau$, we have $H_{-\tau,1}(x_0, x_1) = g_0^{(j-i)\tau} = 1$ if and only if $j \equiv i \bmod p$, i.e., $(x_0, x_1) \in L$.

By the construction, we have $|X/L| = |\Pi| = p$, therefore the homomorphic hash proof system $\mathbf{P} = \widehat{\mathbf{P}}$ associated to the $\mathbf{M}$ and $\mathbf{G}$ is $(1/p)$-universal$_1$ (by Lemma 2.3) and 0-smooth (by Lemma 2.2). On the other hand, let $\Gamma = \Gamma_{hk} \colon X \times \Pi^2 \to \{0, 1, \ldots, p-1\}$ be a function indexed by $hk \in \mathcal{HK}$ sampled from a TCR hash family. Let $f \colon \Pi \to \mathcal{Y}$ be an ϵ_{smth}-smooth function, where ϵ_{smth} is negligible. Then, since $\mathbf{M}$ is a trapdoor subset membership problem and the values $|\Pi|\epsilon_{\mathsf{smth}}/p = \epsilon_{\mathsf{smth}}$ and $|L|/|X| = 1/p$ are negligible, Proposition 5.1 implies that the composition (denoted by $\widetilde{\mathbf{P}}$) of f and the CS hash proof system constructed from the diverse group system $\mathbf{G}$ and the internal function Γ is a first-uniform computationally universal$_2$ hash proof system.

Now we show that the conditions of Theorem 4.1 with Assumption U (where $X' = X$ and $\Pi' = \Pi$) are satisfied (note that the last two conditions in Assumption U are now trivial, since $X' = X$). First, since $|L|/|X| = 1/p$ is negligible, the uniform distributions on X and on $X \setminus L$ have negligible statistical distance, therefore $X \setminus L$ is approximately samplable relative to X. Secondly, for the condition for critical integers, since $|X/L| = p$, we have $o(\Lambda) = p$. On the other hand, we have $|X| = p^2$, therefore any positive integer that is not coprime to $|X|$ is a multiple of p. This implies that there exist no critical integers, therefore the condition for critical integers in Assumption U is automatically satisfied. Hence, all the conditions for Theorem 4.1 with Assumption U are satisfied, therefore the resulting instantiation of our KH-PKE scheme is KH-CCA secure. We write down the instantiation of the KH-PKE scheme in Figure 2.

Efficiency Comparison In Table 1, we give an efficiency comparison of our DDH-based KH-PKE scheme with the CS PKE [11], the KD PKE [29], and the naive construction (see Section 1). We note that the latter

Table 1: Comparison among the Cramer–Shoup (CS) scheme, the Kurosawa–Desmedt (KD) scheme, the KD + CS-lite (using the double encryption) scheme, and our DDH-based KH-PKE scheme (here $|C| - |M|$ denotes ciphertext overhead; $|g|$ denotes the size of an element in the underlying group $\mathbb{G}$; exp denotes exponentiation; and we count 1 multi-exp as 1.2 regular exp, and the size of MAC and the output length of f as ℓ and $n = n(\ell)$, respectively)

	$\lvert C\rvert - \lvert M\rvert$	Cost (Enc)	Cost (Dec)	KH property
CS [11]	$3\lvert g\rvert$	4.2 exp	2.4 exp	No
KD [29]	$2\lvert g\rvert + \ell$	3.2 exp	1.2 exp	No
KD+CS-lite Double Enc	$5\lvert g\rvert + \ell$	7.2 exp	3.6 exp	No?
Our DDH-based KH-PKE	$3\lvert g\rvert + n$	5.4 exp	3.6 exp	Yes

three schemes do not possess the keyed-homomorphic property and/or the KH-CCA security. As seen in Table 1, our scheme is comparably efficient to the best known DDH-based (standard) PKE schemes, i.e. the CS and the KD schemes, in terms of computational costs. The ciphertext size of our construction is dependent on how large the output length n of the smooth function f is. However, as analyzed in Appendix A, if we assume that f is a OWF, an aSec hash function, or a KDF, that is secure against non-uniform adversaries, then n can be as small as ℓ for ℓ-bit security. In this case, the ciphertext overhead of our scheme is only ℓ-bit longer than that of the CS scheme for ℓ-bit security.[3] Then, for 128-bit security, ciphertext overhead of our scheme is 896-bit while that of the Cramer–Shoup scheme is 768-bit (assuming that these schemes are implemented over elliptic curves).

It is somewhat surprising that it is possible to realize KH property with only significantly small additional cost. Furthermore, comparing with the naive construction (from KD and CS(-lite)) which appears to have KH property (but does not satisfy KH-CCA security), we see that our scheme is more efficient. This means that our methodology does not only yield KH property (and KH-CCA security) but also significantly high efficiency.

5.4 DCR-Based Instantiation of KH-PKE

Here we describe the instantiation of our KH-PKE scheme based on the DCR assumption. First, we recall the definition of the DCR assumption.

Definition 5.6 (The Decisional Composite Residuosity (DCR) Assumption [33]). *Let p, q, p', q' be distinct odd primes with $p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are both $\lambda = \lambda(\ell)$ bits in length. Let $N = pq$. We say that* the DCR assumption holds in $\mathbb{Z}^*_{N^2}$, *if for any PPT adversary $\mathcal{A}$, the advantage $Adv^{DCR}_{N,\mathcal{A}}(1^\ell) := |\Pr[\mathcal{A}(g, N) = 0] - \Pr[\mathcal{A}(g^N, N) = 0]|$ is negligible, where g is a uniformly random element of $\mathbb{Z}^*_{N^2}$.*

In order to construct the DCR-based instantiation, we note the following immediate consequence of the DCR assumption:

Lemma 5.2. *Let p, q, p', q' and $N = pq$ be as in the definition of the DCR assumption. If the DCR assumption holds in $\mathbb{Z}^*_{N^2}$, then $|\Pr[\mathcal{A}(g^2, N) = 0] - \Pr[\mathcal{A}(g^{2N}, N) = 0]|$ is negligible for any PPT adversary $\mathcal{A}$, where g is a uniformly random element of $\mathbb{Z}^*_{N^2}$.*

We define a trapdoor subset membership problem $\mathbf{M}$ and a diverse group system $\mathbf{G}$ as follows.

- The instance sampling algorithm for $\mathbf{M}$ chooses the primes p, q, p' and q' as in the DCR assumption, puts $N := pq$, and sets $X := \{g^2 \mid g \in \mathbb{Z}^*_{N^2}\}$ and $L := \{g^{2N} \mid g \in \mathbb{Z}^*_{N^2}\}$. By the choice of N, we have

[3] Even if this "non-uniform" security assumption is not justified (and only security against uniform PPT adversaries is assumed), n can still be as small as at most 2ℓ-bit, which is still smaller than (or in some group equal to) the size of an element in the group $\mathbb{G}$.See our analysis of smoothness of these cryptographic functions in Appendix A.

$\mathbb{Z}_{N^2}^* \simeq \mathbb{Z}_{p^2}^* \times \mathbb{Z}_{q^2}^* \simeq (C_p \times C_2 \times C_{p'}) \times (C_q \times C_2 \times C_{q'})$ where C_n denotes the multiplicative cyclic group of order n, therefore $X \simeq C_p \times C_{p'} \times C_q \times C_{q'}$ and $L \simeq C_{p'} \times C_{q'}$. Let ι denote the isomorphism from X to $C_p \times C_{p'} \times C_q \times C_{q'}$. Moreover, X' is defined to be the subset of X consisting of elements $\iota^{-1}(y)$ with $y = (y_p, y_{p'}, y_q, y_{q'}) \in C_p \times C_{p'} \times C_q \times C_{q'}$ satisfying either $y_p \neq 1$ and $y_q \neq 1$, or $y_p = y_q = 1$. Now let g_* be a generator of L, which can be approximately sampled by $g_* = g^{2N}$ where $g \in \mathbb{Z}_{N^2}^*$ is chosen uniformly at random.[4] Then the instance sampling algorithm sets $W := \{1, \dots, \lfloor N/4 \rfloor\}$, and defines the relation R in such a way that, for $(x, i) \in X \times W$, we have $(x, i) \in R$ if and only if $x = g_*^i$. On the other hand, the subset sampling algorithm first chooses $\omega \in W$ uniformly at random, and then outputs $g_*^\omega \in L$ together with $\omega \in W$ as the witness.[5] Now we have $|X| = pqp'q'$ and $|X' \setminus L| = (p-1)(q-1)p'q'$, therefore $|X' \setminus L|/|X|$ is overwhelming and the three uniform distributions on X, on $X \setminus L$ and on $X' \setminus L$ have negligible statistical distances from each other. By this and Lemma 5.2, it follows that the construction of $\mathbf{M}$ satisfies the condition for a hard subset membership problem relative to X' provided the DDH assumption holds in $\mathbb{Z}_{N^2}^*$. Moreover, $X' \setminus L$ is approximately samplable relative to X.

- In the trapdoor mode for $\mathbf{M}$, the algorithm also outputs $\tau := p'q'$ as the trapdoor element. Then, by using τ, it can be efficiently decided whether a given $x \in X$ is in L or not, by checking if $x^\tau = 1$. Hence, $\mathbf{M}$ is a hard trapdoor subset membership problem (relative to X').

- To define the corresponding diverse group system $\mathbf{G}$, we set $\Pi := X$, and define $\mathcal{H}$ to be the set of homomorphisms $H_k \colon X \to \Pi$, indexed by $k \in K := \mathbb{Z}_{pqp'q'}$, satisfying that $H_k(x) := x^k$ for any $x \in X$. Then $\mathbf{G}$ is diverse; indeed, for any $x \in X$, we have $H_{p'q'}(x) = x^{p'q'} = 1$ if and only if $x \in L$.

By the construction, we have $|X/L| = pq$ and $|\Pi| = pqp'q'$, therefore $\widetilde{p} = \min\{p, q\}$. This implies that, by setting both $\mathbf{P}$ and $\widehat{\mathbf{P}}$ to be the homomorphic HPS associated to $\mathbf{G}$, $\widehat{\mathbf{P}}$ is $(1/\widetilde{p})$-universal$_1$ by Lemma 2.3, and $1/\widetilde{p}$ is negligible. On the other hand, for the HPS $\mathbf{P}$, we define the subgroup Π' of $\Pi = X$ by $\Pi' := \iota^{-1}(C_p \times 1 \times C_q \times 1)$. We note that $\Pi' = \{x \in X \mid x^N = 1\}$ and it is generated by $1 + N \in \mathbb{Z}_{N^2}^*$, therefore a uniformly random element of Π' can be efficiently chosen (in particular, Π' is approximately samplable relative to Π). Now we have the following:

Lemma 5.3. *In the setting, $\mathbf{P}$ is 0-smooth relative to (X', Π').*

Proof. Let $k \in K$ and $x \in X' \setminus L$. Write $k = \lambda_1 p'q' + \lambda_2$ with $\lambda_1 \in \{0, 1, \dots, pq-1\}$ and $\lambda_2 \in \{0, 1, \dots, p'q'-1\}$, and $\iota(x) = y = (y_p, y_{p'}, y_q, y_{q'})$. Then we have $y_p \neq 1$ and $y_q \neq 1$ by the definition of X' and L. Put $y_{p,q} := (y_p, 1, p_q, 1)$ and $y_{p',q'} := (1, y_{p'}, 1, y_{q'})$. On the other hand, we have $s = \alpha(k) = g_*^k = g_*^{\lambda_2}$ since $g_* \in L$, therefore λ_2 is uniquely determined from s since g_* is a generator of L. Now we have

$$y^k = y^{\lambda_1 p'q'} y^{\lambda_2} = y_{p,q}^{\lambda_1 p'q'} y_{p',q'}^{\lambda_1 p'q'} y^{\lambda_2} = (y_{p,q}^{p'q'})^{\lambda_1} y^{\lambda_2} \enspace ,$$

therefore $x^k = \iota^{-1}(y_{p,q}^{p'q'})^{\lambda_1} x^{\lambda_2}$. Since $y_p \neq 1$ and $y_q \neq 1$, $y_{p,q}^{p'q'}$ is a generator of $C_p \times 1 \times C_q \times 1$. Hence, when k is chosen uniformly at random subject to the condition $\alpha(k) = s$ for a given s, λ_1 is uniformly random while λ_2 is fixed, therefore x^k is the product of the fixed element x^{λ_2} of Π and a uniformly random element $\iota^{-1}(y_{p,q}^{p'q'})^{\lambda_1}$ of Π'. This implies that $\mathbf{P}$ is 0-smooth relative to (X', Π'), as desired. □

Moreover, let $\Gamma = \Gamma_{hk} \colon X \times \Pi^2 \to \{0, 1, \dots, \widetilde{p} - 1\}$ be a function indexed by $hk \in \mathcal{HK}$ sampled from a CR hash family. Let $f \colon \Pi \to \mathcal{Y}$ be an ϵ_{smth}-smooth function, where ϵ_{smth} satisfies that the value $|\Pi|\epsilon_{\mathsf{smth}}/\widetilde{p} = pqp'q'\epsilon_{\mathsf{smth}}/\min\{p, q\}$ is negligible (for example, f may be an identity mapping $\Pi \to \Pi$; then we have $\epsilon_{\mathsf{smth}} = 1/|\Pi|$ and $|\Pi|\epsilon_{\mathsf{smth}}/\widetilde{p} = 1/\widetilde{p}$ is negligible, as desired). Then, since $\mathbf{M}$ is a trapdoor subset membership problem, Proposition 5.1 implies that the composition (denoted by $\widetilde{\mathbf{P}}$) of f and the CS hash proof system constructed from the diverse group system $\mathbf{G}$ and the internal function Γ is a first-adaptive computationally universal$_2$ hash proof system.

[4] The probability that g_* is not a generator of L is $1 - (1 - 1/p')(1 - 1/q') = 1/p' + 1/q' - 1/(p'q')$, which is negligible (otherwise, the DCR assumption can be trivially broken since $\mathbb{Z}_{N^2}^*$ is not large enough).

[5] The distribution of the g_*^ω and the uniform distribution on L have statistical distance $(\lfloor N/4 \rfloor - p'q')(2/(p'q') - 1/(p'q')) \leq (2p' + 1)(2q' + 1)/(4p'q') - 1 = 1/(2p') + 1/(2q') + 1/(4p'q')$, which is negligible.

```
$\mathsf{KeyGen}(1^\ell)$ :
    $hk \xleftarrow{\$} \mathcal{HK}$; $\mu \xleftarrow{\$} \mathbb{Z}_{N^2}^*$; $g \leftarrow \mu^{2N}$
    $k, \widehat{k}, \widetilde{k}_0, \widetilde{k}_1 \xleftarrow{\$} \{1, \ldots, \lfloor N^2/4 \rfloor\}$
    $s \leftarrow g^k$;   $\widehat{s} \leftarrow g^{\widehat{k}}$;   $\widetilde{s}_0 \leftarrow g^{\widetilde{k}_0}$;   $\widetilde{s}_1 \leftarrow g^{\widetilde{k}_1}$
    $pk \leftarrow (hk, g, s, \widehat{s}, \widetilde{s}_0, \widetilde{s}_1)$
    $sk_d \leftarrow (k, \widehat{k}, \widetilde{k}_0, \widetilde{k}_1)$
    $sk_h \leftarrow (\widetilde{k}_0, \widetilde{k}_1)$
    Return $(pk, sk_d, sk_h)$
```

```
$\mathsf{Dec}(sk_d, C)$ :
    Parse $C$ as $(x, e, \widehat{\pi}, y)$
    $\widehat{\pi}' \leftarrow x^{\widehat{k}}$
    $\gamma' \leftarrow \Gamma_{hk}(x, e, \widehat{\pi})$;   $y' \leftarrow f(x^{\widetilde{k}_0 + \gamma' \widetilde{k}_1})$
    If either $\widehat{\pi} \neq \widehat{\pi}'$ or $y \neq y'$ then
        return $\perp$
    $\pi \leftarrow x^k$;   $\widetilde{M} \leftarrow e \cdot \pi^{-1}$
    Return $M \leftarrow (\widetilde{M} - 1)/N$
```

```
$\mathsf{Enc}(pk, M)$ (for $M \in \mathcal{M} := \mathbb{Z}_N$) :
    $\omega \xleftarrow{\$} \{1, \ldots, \lfloor N/4 \rfloor\}$;   $x \leftarrow g^\omega$
    $\pi \leftarrow s^\omega$;   $e \leftarrow (1 + N)^M \cdot \pi$
    $\widehat{\pi} \leftarrow \widehat{s}^\omega$
    $\gamma \leftarrow \Gamma_{hk}(x, e, \widehat{\pi})$;   $y \leftarrow f((\widetilde{s}_0 \cdot \widetilde{s}_1^\gamma)^\omega)$
    Return $C = (x, e, \widehat{\pi}, y)$
```

```
$\mathsf{Eval}(sk_h, C_1, C_2)$ :
    Parse $C_b$ as $(x_b, e_b, \widehat{\pi}_b, y_b)$ for $b = 1, 2$
    $\gamma_b \leftarrow \Gamma_{hk}(x_b, e_b, \widehat{\pi}_b)$ for $b = 1, 2$
    $y_b' \leftarrow f(x^{\widetilde{k}_0 + \gamma_b \widetilde{k}_1})$ for $b = 1, 2$
    If either $y_1 \neq y_1'$ or $y_2 \neq y_2'$ then
        return $\perp$
    $\omega \xleftarrow{\$} \{1, \ldots, \lfloor N^2/4 \rfloor\}$
    $x \leftarrow x_1 x_2 g^\omega$;   $e \leftarrow e_1 e_2 s^\omega$;   $\widehat{\pi} \leftarrow \widehat{\pi}_1 \widehat{\pi}_2 \widehat{s}^\omega$
    $\gamma \leftarrow \Gamma_{hk}(x, e, \widehat{\pi})$;   $y \leftarrow f(x^{\widetilde{k}_0 + \gamma \widetilde{k}_1})$
    Return $C \leftarrow (x, e, \widehat{\pi}, y)$
```

Figure 3: DCR-based instantiation of our KH-PKE scheme (here $N = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$ satisfies that the DCR assumption holds in $\mathbb{Z}_{N^2}^*$; $\{\Gamma = \Gamma_{hk} \colon X^3 \to \{0, 1, \ldots, \widetilde{p} - 1\} \mid hk \in \mathcal{HK}\}$ is a CR hash family where $X = \{g^2 \mid g \in \mathbb{Z}_{N^2}^*\}$ and $\widetilde{p} = \min\{p, q\}$; and $f \colon X \to \mathcal{Y}$ is a smooth function)

Summarizing, all the conditions for Theorem 4.1 with Assumption A are satisfied, therefore the resulting instantiation of our KH-PKE scheme is KH-CCA secure. We write down the instantiation of the KH-PKE scheme in Figure 3. Here we note that, for the choice of secret keys for the hash proof systems, the uniform distribution on $\{1, \ldots, pqp'q'\}$ has negligible statistical distance from the uniform distribution on $\{1, \ldots, \lfloor N^2/4 \rfloor\}$. We note also that, the multiplicative group Π' is isomorphic to the additive group $\mathbb{Z}_N$, with efficiently computable isomorphism $\mathbb{Z}_N \ni M \mapsto (1 + N)^M \bmod N^2 \in \Pi'$ and its efficiently computable inverse $\Pi' \ni \widetilde{M} \mapsto (\widetilde{M} - 1)/N \bmod N \in \mathbb{Z}_N$. In the instantiation here, we switch the plaintext space from Π' to $\mathbb{Z}_N$ via the isomorphism. As in [13], we implicitly assume that the Dec algorithm checks that x, e, and $\widehat{\pi}$ lie in $\mathbb{Z}_{N^2}^*$ and $\widetilde{M} - 1$ is a multiple of N.

Acknowledgement: We thank the anonymous reviewers and the members of Shin-Akarui-Angou-Benkyou-Kai for their helpful comments. This work was supported by JSPS KAKENHI Grant Number 24700009.

References

[1] M. Abe, J. Groth, M. Ohkubo, and M. Tibouchi. Unified, minimal and selectively randomizable structure-preserving signatures. In *TCC*, pages 688–712, 2014.

[2] J. H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *EUROCRYPT*, pages 83–107, 2002.

[3] M. Barbosa and P. Farshim. Delegatable homomorphic encryption with applications to secure outsourcing of computation. In *CT-RSA*, pages 296–312, 2012.

[4] M. Bellare and P. Rogaway. Collision-resistant hashing: Towards making UOWHFs practical. In *CRYPTO*, pages 470–484, 1997.

[5] D. Bernhard, V. Cortier, O. Pereira, B. Smyth, and B. Warinschi. Adapting Helios for provable ballot privacy. In *ESORICS*, pages 335–354, 2011.

[6] D. Boneh, G. Segev, and B. Waters. Targeted malleability: homomorphic encryption for restricted computations. In *ITCS*, pages 350–366, 2012.

[7] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, pages 207–222, 2004.

[8] R. Canetti, H. Krawczyk, and J. B. Nielsen. Relaxing chosen-ciphertext security. In *CRYPTO*, pages 565–582, 2003.

[9] D. Cash, E. Kiltz, and V. Shoup. The twin Diffie-Hellman problem and applications. In *EUROCRYPT*, pages 127–145, 2008.

[10] M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn. Malleable proof systems and applications. In *EUROCRYPT*, pages 281–300, 2012.

[11] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.

[12] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. Cryptology ePrint Archive, Report 2001/085, 2001. `http://eprint.iacr.org/`.

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

[14] Y. Desmedt, R. Gennaro, K. Kurosawa, and V. Shoup. A new and improved paradigm for hybrid encryption secure against chosen-ciphertext attack. *J. Cryptology*, 23(1):91–120, 2010.

[15] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[16] K. Emura, G. Hanaoka, G. Ohtake, T. Matsuda, and S. Yamada. Chosen ciphertext secure keyed-homomorphic public-key encryption. In *Public Key Cryptography*, pages 32–50, 2013.

[17] D. Galindo and J. L. Villar. An instantiation of the Cramer-Shoup encryption paradigm using bilinear map groups. Workshop on Mathematical Problems and Techniques in Cryptology, 2005.

[18] C. Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.

[19] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

[20] S. Goldwasser and S. Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *STOC*, pages 365–377, 1982.

[21] J. Groth. Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In *TCC*, pages 152–170, 2004.

[22] G. Hanaoka and K. Kurosawa. Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. In *ASIACRYPT*, pages 308–325, 2008.

[23] B. Hemenway and R. Ostrovsky. On homomorphic encryption and chosen-ciphertext security. In *Public Key Cryptography*, pages 52–65, 2012.

[24] D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In *CRYPTO*, pages 553–571, 2007.

[25] C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In *ASIACRYPT (1)*, pages 1–20, 2013.

[26] E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, pages 581–600, 2006.

[27] E. Kiltz. Chosen-ciphertext secure key-encapsulation based on gap hashed Diffie-Hellman. In *PKC*, pages 282–297, 2007.

[28] E. Kiltz, K. Pietrzak, M. Stam, and M. Yung. A new randomness extraction paradigm for hybrid encryption. In *EUROCRYPT*, pages 590–609, 2009.

[29] K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO*, pages 426–442, 2004.

[30] B. Libert, T. Peters, M. Joye, and M. Yung. Linearly homomorphic structure-preserving signatures and their applications. In *CRYPTO (2)*, pages 289–307, 2013.

[31] B. Libert, T. Peters, M. Joye, and M. Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive nizk proofs and CCA2-secure encryption from homomorphic signatures. In *EUROCRYPT*, pages 514–532, 2014.

[32] J. Loftus, A. May, N. P. Smart, and F. Vercauteren. On CCA-secure somewhat homomorphic encryption. In *Selected Areas in Cryptography*, pages 55–72, 2011.

[33] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238, 1999.

[34] K. G. Paterson, J. C. N. Schuldt, M. Stam, and S. Thomson. On the joint security of encryption and signature, revisited. In *ASIACRYPT*, pages 161–178, 2011. The full version is available at `http://eprint.iacr.org/2011/486`.

[35] M. Prabhakaran and M. Rosulek. Rerandomizable RCCA encryption. In *CRYPTO*, pages 517–534, 2007.

[36] M. Prabhakaran and M. Rosulek. Homomorphic encryption with CCA security. In *ICALP (2)*, pages 667–678, 2008.

[37] H. Shacham. A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074, 2007. `http://eprint.iacr.org/`.

[38] V. Shoup. A proposal for an ISO standard for public key encryption. Cryptology ePrint Archive, Report 2001/112, 2001. `http://eprint.iacr.org/`.

A Smoothness of Cryptographic Functions

In this section, we show that natural cryptographic functions, a one-way function (OWF), an always second-preimage resistant (aSec secure) hash function [34], and a key derivation function (KDF) [14], are smooth in the sense of Definition 5.3.

Interestingly, although the amount of smoothness, Smth_f, is always negligible, its "tightness" is different depending on whether the function f is secure against uniform adversaries or against *non-uniform* adversaries.[6] More specifically, for each cryptographic function f considered here, we show that the smoothness of f is (essentially) upperbounded by the square root of the advantage of some (uniform) PPT adversary $\mathcal{A}$ attacking the security of the function f. We also show that the smoothness of f is (essentially) upperbounded by the advantage of some non-uniform PPT adversary A_{nu}. These results suggest that if we can assume the security of these cryptographic functions against non-uniform adversaries, then the output length can be as small as ℓ-bit for ℓ-bit security, because the smoothness of the functions are "tightly" upperbounded by the advantage of "non-uniform" adversaries attacking the security of the cryptographic functions Furthermore,

[6] Recall that a non-uniform algorithm is an algorithm that takes as an advice string (which is dependent only on the input length) as an additional input. The class of non-uniform PPT algorithms is equivalent to the class of polynomial-sized circuit families.

even if this "non-uniform" security assumption is not justified (and instead only security against uniform adversaries is assumed), the output length the function can still be as small as at most 2ℓ-bit, because the main term that contribute to the smoothness is the square root of the advantage of an adversary attacking the security of the cryptographic functions (against uniform PPT adversaries).

In practice, for example, (an appropriate modification of) cryptographic hash functions such as SHA-series, can be assumed to be the cryptographic functions (secure against non-uniform adversaries) considered here.

Some Notation: To show the smoothness of each cryptographic function, it is useful to introduce the following notation. Let $f : \mathcal{X}_\ell \to \{0,1\}^\ell$ be a function. For each $\ell \in \mathbb{N}$, let $y_\ell^{\max} \in \{0,1\}^\ell$ be the lexicographically smallest string[7] such that $\Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[f(x) = y_\ell^{\max}] \geq \Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[f(x) = y]$ holds for any $y \in \{0,1\}^\ell$. Then, by definition, we have $\mathsf{Smth}_f = \max_{y \in \{0,1\}^\ell} \Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[f(x) = y] = \Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[f(x) = y_\ell^{\max}]$. Next, for each $\ell \in \mathbb{N}$, we define $x_\ell^{\max} \in \mathcal{X}_\ell$ to be the lexicographically smallest string in the set $\{x \in \mathcal{X}_\ell | f(x) = y_\ell^{\max}\}$. Note that $y_\ell^{\max} \in \{0,1\}^\ell$ and $x_\ell^{\max} \in \mathcal{X}_\ell$ are uniquely determined for each $\ell \in \mathbb{N}$.

For the function f, it is also useful to note the following properties about the probability of "collision" for random inputs:

$$\Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[f(x) = y_\ell^{\max}] = \mathsf{Smth}_f \qquad \text{and} \qquad \Pr_{x,x' \xleftarrow{\$} \mathcal{X}_\ell}[f(x) = f(x')] \geq (\mathsf{Smth}_f)^2, \tag{1}$$

where the former is by definition, and the latter is obtained as follows:

$$\begin{aligned}\Pr_{x,x' \xleftarrow{\$} \mathcal{X}_\ell}[f(x) = f(x')] &\geq \Pr_{x,x' \xleftarrow{\$} \mathcal{X}_\ell}[f(x) = y_\ell^{\max} \wedge f(x) = y_\ell^{\max}] \\ &= (\Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[f(x) = y_\ell^{\max}])^2 = (\mathsf{Smth}_f)^2\end{aligned}$$

A.1 One-Way Function

Definition A.1 (One-Way Function (OWF))**.** *Let $f : \mathcal{X}_\ell \to \{0,1\}^\ell$ be a function, where $n = n(\ell) := \log_2 |\mathcal{X}_\ell| \in \omega(\log_2 \ell)$. We say that f is a one-way function (OWF) if (1) f is efficiently computable in terms of the security parameter ℓ (and thus n is some polynomial of ℓ), (2) we can efficiently sample an element uniformly at random from the domain $\mathcal{X}_\ell$, and (3) $Adv_{\mathcal{A}}^{\mathsf{OWF}}(\ell) := \Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[x' \leftarrow \mathcal{A}(1^\ell, f(x)) : f(x') = f(x)]$ is negligible for any PPT algorithm $\mathcal{A}$.*

Furthermore, we say that f is a OWF against non-uniform adversaries if the condition (3) is replaced with "$Adv_{\mathcal{A}_{\mathrm{nu}}}^{\mathsf{OWF}}(\ell)$ is negligible for any non-uniform PPT algorithms $\mathcal{A}_{\mathrm{nu}}$."

Lemma A.1. *If f is a OWF as defined in Definition A.1, then f is smooth. Specifically, there exists a PPT algorithm $\mathcal{A}$ such that*

$$\mathsf{Smth}_f \leq \sqrt{Adv_{\mathcal{A}}^{\mathsf{OWF}}(\ell)}.$$

Furthermore, there exists a non-uniform PPT algorithm $\mathcal{A}_{\mathrm{nu}}$ such that

$$\mathsf{Smth}_f = Adv_{\mathcal{A}_{\mathrm{nu}}}^{\mathsf{OWF}}(\ell).$$

Proof. We first show the existence of the uniform PPT adversary $\mathcal{A}$ against the one-wayness of f. Consider the algorithm $\mathcal{A}$ that takes 1^ℓ and $y = f(x)$ (where $x \in \mathcal{X}_\ell$ is chosen uniformly at random) as input, picks $x' \in \mathcal{X}_\ell$ uniformly at random, and terminates with output this x'. Note that $\mathcal{A}$ is a (uniform) PPT algorithm, and its one-wayness advantage is as follows:

$$Adv_{\mathcal{A}}^{\mathsf{OWF}}(\ell) = \Pr_{x,x' \leftarrow \mathcal{X}_\ell}[f(x) = f(x')] \geq (\mathsf{Smth}_f)^2$$

[7] In general, there could be multiple strings $y \in \{0,1\}^\ell$ that maximize the probability $\Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[f(x) = y]$. Choosing the lexicographically smallest one is to canonically specify one of such strings.

where in the last step we use the inequation (1). Therefore, we have $\mathsf{Smth}_f \leq \sqrt{Adv_{\mathcal{A}}^{\mathsf{OWF}}(\ell)}$, as required.

We next show the existence of the non-uniform adversary $\mathcal{A}_{\mathrm{nu}}$ against the one-wayness of f. Consider the non-uniform PPT algorithm $\mathcal{A}_{\mathrm{nu}}$ that has $x_\ell^{\max}$ as an advice (i.e. $x_\ell^{\max}$ is hard-wired inside $\mathcal{A}_{\mathrm{nu}}$ for each security parameter $\ell \in \mathbb{N}$), takes 1^ℓ and $y = f(x)$ as input (where $x \in \mathcal{X}_\ell$ is chosen uniformly at random), and terminates with output the string $x_\ell^{\max}$. Clearly $\mathcal{A}_{\mathrm{nu}}$ is PPT, and its one-wayness advantage is:

$$Adv_{\mathcal{A}_{\mathrm{nu}}}^{\mathsf{OWF}}(\ell) = \Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[f(x) = f(x_\ell^{\max})] = \Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[f(x) = y_\ell^{\max}] = \mathsf{Smth}_f,$$

as required.

This completes the proof of Lemma A.1. □

A.2 Always Second-Preimage Resistant Hash Functions

Definition A.2 (Always Second-Preimage Resistant (aSec) Hash Functions [34])**.** *Let* $\mathsf{H} : \mathcal{X}_\ell \to \{0,1\}^\ell$ *be a function, where* $n = n(\ell) := \log_2 |\mathcal{X}_\ell| \in \omega(\log_2 \ell)$*. We say that* H *is an* always second-preimage resistant *(aSec secure) hash function if (1)* H *is efficiently computable in terms of the security parameter* ℓ *(and thus* n *is some polynomial of* ℓ*), (2) we can efficiently sample an element uniformly at random from the domain* $\mathcal{X}_\ell$*, (3)* $Adv_{\mathcal{A}}^{\mathsf{aSec}}(\ell) := \Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[x' \leftarrow \mathcal{A}(1^\ell, x) : \mathsf{H}(x') = \mathsf{H}(x) \land x' \neq x]$ *is negligible for any PPT algorithm* $\mathcal{A}$*.*

Furthermore, we say that H *is an aSec secure hash function against non-uniform adversaries if the condition (3) is replaced with "*$Adv_{\mathcal{A}}^{\mathsf{aSec}}(\ell)$ *is negligible for any non-uniform PPT algorithm."*

We remark that an aSec secure hash function is (close to but) different from the notion of universal one way hash function (UOWHF) [4]. UOWHF is a family of hash functions (or a keyed hash function), and in the security experiment, an adversary is allowed to choose the first message x for which the adversary has to find a collision, but is required to find a colliding input x' under a randomly chosen key hk.

Lemma A.2. *If* H *is an aSec secure hash function as defined in Definition A.2, then* H *is smooth. Specifically, there exists a PPT algorithm* $\mathcal{A}$ *such that*

$$\mathsf{Smth}_{\mathsf{H}} \leq \sqrt{Adv_{\mathcal{A}}^{\mathsf{aSec}}(\ell) + |\mathcal{X}_\ell|^{-1}}.$$

Furthermore, there exists a non-uniform PPT algorithm $\mathcal{A}_{\mathrm{nu}}$ *such that*

$$\mathsf{Smth}_{\mathsf{H}} = Adv_{\mathcal{A}_{\mathrm{nu}}}^{\mathsf{aSec}}(\ell) + |\mathcal{X}_\ell|^{-1}.$$

Proof. The proof proceeds very similarly to that of Lemma A.1. First, we show the existence of the uniform PPT adversary $\mathcal{A}$ against the aSec security of H. Consider the algorithm $\mathcal{A}$ that takes 1^ℓ and x (for a uniformly chosen value $x \in \mathcal{X}_\ell$) as input, picks $x' \in \mathcal{X}_\ell$ uniformly at random, and terminates with output this x'. Note that $\mathcal{A}$ is trivially a (uniform) PPT algorithm, and its advantage against aSec security of $\mathcal{H}$ is as follows:

$$\begin{aligned} Adv_{\mathcal{A}}^{\mathsf{aSec}}(\ell) &= \Pr_{x,x' \xleftarrow{\$} \mathcal{X}_\ell}[\mathsf{H}(x) = \mathsf{H}(x') \land x \neq x'] \\ &= \Pr_{x,x' \xleftarrow{\$} \mathcal{X}_\ell}[\mathsf{H}(x) = \mathsf{H}(x')] - \Pr_{x,x' \xleftarrow{\$} \mathcal{X}_\ell}[x = x'] \\ &\geq (\mathsf{Smth}_{\mathsf{H}})^2 - |\mathcal{X}_\ell|^{-1} \end{aligned}$$

Therefore, we have $\mathsf{Smth}_{\mathsf{H}} \leq \sqrt{Adv_{\mathcal{A}}^{\mathsf{aSec}}(\ell) + |\mathcal{X}_\ell|^{-1}}$, as required.

We next show the existence of the non-uniform adversary $\mathcal{A}_{\mathrm{nu}}$ against the aSec security of H. Consider the non-uniform PPT algorithm $\mathcal{A}_{\mathrm{nu}}$ that has $x_\ell^{\max} \in \mathcal{X}_\ell$ as an advice (i.e. $x_\ell^{\max}$ is hard-wired inside $\mathcal{A}_{\mathrm{nu}}$

for each security parameter $\ell \in \mathbb{N}$), takes 1^ℓ and x as input (where x is chosen uniformly at random), and terminates with output the string $x_\ell^{\max}$. Clearly $\mathcal{A}_{\rm nu}$ is PPT, and its advantage is:

$$\begin{aligned}
Adv^{\mathsf{aSec}}_{\mathcal{A}_{\rm nu}}(\ell) &= \Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[\mathsf{H}(x) = \mathsf{H}(x_\ell^{\max}) \land x \neq x_\ell^{\max}] \\
&= \Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[\mathsf{H}(x) = y_\ell^{\max}] - \Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[x = x_\ell^{\max}] \\
&= \mathsf{Smth}_{\mathsf{H}} - |\mathcal{X}_\ell|^{-1},
\end{aligned}$$

Therefore, we have $\mathsf{Smth}_{\mathsf{H}} = Adv^{\mathsf{aSec}}_{\mathcal{A}_{\rm nu}}(\ell) + |\mathcal{X}_\ell|^{-1}$, as required.

This completes the proof of Lemma A.2. □

A.3 Key Derivation Function

Definition A.3 (Key Derivation Function (KDF) [14]). *Let* $\mathsf{KDF} : \mathcal{X}_\ell \to \{0,1\}^\ell$ *be a function, where* $n = n(\ell) := \log_2 |\mathcal{X}_\ell| \in \omega(\log_2 \ell)$. *We say that* KDF *is a secure key derivation function (KDF) if (1)* KDF *is efficiently computable in terms of the security parameter* ℓ *(and thus* n *is some polynomial of* ℓ*), (2) We can efficiently sample an element uniformly at random from the domain* $\mathcal{X}_\ell$*, and (3)* $Adv^{\mathsf{KDF}}_{\mathcal{A}}(\ell) := |\Pr_{x \xleftarrow{\$} \Delta}[\mathcal{A}(1^\ell, \mathsf{KDF}(x)) = 1] - \Pr_{y \xleftarrow{\$} \{0,1\}^\ell}[\mathcal{A}(1^\ell, y) = 1]|$ *is negligible for any PPT algorithm* $\mathcal{A}$.

Furthermore, we say that KDF *is a secure KDF against non-uniform adversaries if* $Adv^{\mathsf{KDF}}_{\mathcal{A}_{\rm nu}}(\ell)$ *is negligible for any non-uniform algorithm* $\mathcal{A}_{\rm nu}$.

Lemma A.3. *If* KDF *be a secure key derivation function as defined in Definition A.3, then* KDF *is smooth. Specifically, there exists a uniform PPT algorithm* $\mathcal{A}$ *such that*

$$\mathsf{Smth}_{\mathsf{KDF}} \leq \sqrt{Adv^{\mathsf{KDF}}_{\mathcal{A}}(\ell) + 2^{-\ell}}.$$

Furthermore, there exists a non-uniform PPT algorithm $\mathcal{A}_{nu}$ *such that*

$$\mathsf{Smth}_{\mathsf{KDF}} = Adv^{\mathsf{KDF}}_{\mathcal{A}_{\rm nu}}(\ell) + 2^{-\ell}.$$

Proof. We first show the existence of the uniform PPT adversary $\mathcal{A}$ against the security of KDF. Consider the algorithm $\mathcal{A}$ that takes 1^ℓ and $y \in \{0,1\}^\ell$ as input, picks $x' \in \mathcal{X}_\ell$ uniformly at random, and returns 1 if $G(x') = y$ or returns 0 otherwise. Note that $\mathcal{A}$ is clearly PPT, and its advantage is as follows:

$$\begin{aligned}
Adv^{\mathsf{KDF}}_{\mathcal{A}}(\ell) &= |\Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[\mathcal{A}(1^\ell, \mathsf{KDF}(x)) = 1] - \Pr_{y \xleftarrow{\$} \{0,1\}^\ell}[\mathcal{A}(1^\ell, y) = 1]| \\
&= |\Pr_{x, x' \xleftarrow{\$} \mathcal{X}_\ell}[\mathsf{KDF}(x) = \mathsf{KDF}(x')] - \Pr_{y \xleftarrow{\$} \{0,1\}^\ell, x' \xleftarrow{\$} \mathcal{X}_\ell}[y = \mathsf{KDF}(x')]| \\
&\geq (\mathsf{Smth}_{\mathsf{KDF}})^2 - 2^{-\ell},
\end{aligned}$$

where in the last inequality we use the inequality (1) and the fact that y is chosen uniformly at random from $\{0,1\}^\ell$. Therefore, we have $\mathsf{Smth}_{\mathsf{KDF}} \leq \sqrt{Adv^{\mathsf{KDF}}_{\mathcal{A}}(\ell) + 2^{-\ell}}$, as required.

Next, we show the existence of the non-uniform PPT adversary $\mathcal{A}_{\rm nu}$ against the security of KDF. Consider the algorithm $\mathcal{A}_{\rm nu}$ that has $y_\ell^{\max} \in \{0,1\}^\ell$ as an advice (i.e. $y_\ell^{\max}$ is hard-wired inside $\mathcal{A}_{\rm nu}$ for each $\ell \in \mathbb{N}$), takes 1^ℓ and $y \in \{0,1\}^\ell$ as input, and returns 1 if $y = y_\ell^{\max}$ or returns 0 otherwise. Note that $\mathcal{A}_{\rm nu}$ is clearly PPT, and its advantage is as follows:

$$\begin{aligned}
Adv^{\mathsf{KDF}}_{\mathcal{A}_{\rm nu}}(\ell) &= |\Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[\mathsf{A}_{\rm nu}(1^\ell, \mathsf{KDF}(x)) = 1] - \Pr_{y \xleftarrow{\$} \{0,1\}^\ell}[\mathsf{A}_{\rm nu}(1^\ell, y) = 1]| \\
&= |\Pr_{x \xleftarrow{\$} \mathcal{X}_\ell}[\mathsf{KDF}(x) = y_\ell^{\max}] - \Pr_{y \xleftarrow{\$} \{0,1\}^\ell}[y = y_\ell^{\max}]| \\
&= \mathsf{Smth}_{\mathsf{KDF}} - 2^{-\ell}.
\end{aligned}$$

Therefore, we have $\mathsf{Smth}_{\mathsf{KDF}} = Adv^{\mathsf{KDF}}_{\mathcal{A}_{\mathrm{nu}}}(\ell) + 2^{-\ell}$, as required.
This completes the proof of Lemma A.3. □