Chosen Ciphertext Secure Keyed-Homomorphic Public-Key Encryption

Keita Emura[1], Goichiro Hanaoka[2], Go Ohtake[3],
Takahiro Matsuda[2], and Shota Yamada[4]

[1] National Institute of Information and Communications Technology (NICT), Japan,
k-emura@nict.go.jp
[2] Japan Broadcast Corporation, Japan, ohtake.g-fw@nhk.or.jp
[3] National Institute of Advanced Industrial Science and Technology (AIST), Japan,
{t-matsuda, hanaoka-goichiro}@aist.go.jp
[4] The University of Tokyo, Japan, yamada@it.k.u-tokyo.ac.jp

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, a *homomorphic transitional universal hash family*, and present a number of KH-PKE schemes through hash proof systems. We also present a practical construction of KH-PKE from the DDH assumption. For ℓ-bit security, our DDH-based 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 [13, 16, 25]. Furthermore, fully homomorphic encryption (FHE) that allows a homomorphic operation with respect to any circuit,

has recently been proposed by Gentry [15]. 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).

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., [1, 6, 17, 26, 27]) 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, a *homomorphic transitional universal hash family*, which can be constructed from any diverse group system [11], and present a number of KH-PKE schemes through hash proof systems (HPSs) [11].

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.

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 [9], 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 5ℓ-bit longer ciphertext size even if we choose the Kurosawa-Desmedt PKE scheme [23] and the Cramer-Shoup lite PKE scheme [9] that seems the most efficient combination under the DDH assumption. We give the comparison in Section 5.

To sum up, our construction is superior than 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 [11] 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 [22], 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 *homomorphic transitional universal hash family*) can be obtained from any diverse group system [11], and then we propose a generic construction of KH-PKE based on a homomorphic transitional universal HPS. Moreover, as an implication result, KH-PKE is implied by CPA-secure homomorphic PKE (with cyclic ciphertext space) which implies diverse group systems [19].

Instantiations : According to our methodology, we present a number of KH-PKE schemes from various major cryptographic assumptions such as the DDH assumption, the decisional composite residuosity (DCR) assumption, the decisional linear (DLIN) assumption, the decisional bilinear Diffie-Hellman (DBDH) assumption, and the decisional quadratic residuosity (DQR) assumption. This means that it is not difficult to extend all existing HPS to have the homomorphic transitional property, and thus a homomorphic transitional HPS is not a significantly stronger primitive in practice, compared to an ordinary HPS.

In this paper, we present a practical DDH-based KH-PKE scheme. Other KH-PKE schemes based on the DCR assumption and the DQR assumption from the Cramer-Shoup HPSs [11], based on the DLIN assumption from the Shacham HPS [28], and based on the DBDH assumption from the Galindo-Villar HPS [12], 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 [14] 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. [6] 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 [17, 26].) 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 [1, 29], 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 [27], 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 [4].) 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. [8] 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 [24, 3]. 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 [2] 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.

2 Preliminaries

In this section, we review the basic notations and definitions related to HPSs (mostly following [11] but slightly customized for our convenience).

Throughout this paper, PPT denotes *probabilistic polynomial time*. If n is a natural number, then $[n] = \{1, \ldots, n\}$. If D is a probabilistic 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 $\epsilon = \epsilon(\ell)$.

Projective Hash Families : Let X, Π, W, K, and S be finite, non-empty sets, 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 $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ is a *projective hash family* for (X, L) if for all $k \in K$, the action of H_k on the subset L is uniquely determined by $\alpha(k) \in S$.

Let $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ be a projective hash family, and let $\epsilon \geq 0$. 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 $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ is ϵ-*smooth* if the following two distributions are ϵ-close: $\{k \xleftarrow{\$} K;\ x \xleftarrow{\$} X \setminus L\ :\ (\alpha(k), x, H_k(x))\ \}$ and $\{k \xleftarrow{\$} K;\ x \xleftarrow{\$} X \setminus L;\ \pi \xleftarrow{\$} \Pi\ :\ (\alpha(k), x, \pi)\ \}$.

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).

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 $\Lambda[X, L, W, R] \in [I_\ell]$ specifies non-empty sets X, W, and L, a binary relation R defined over $X \times W$, where X, W, and L are non-empty sets such that $L \subset X$, and an $x \in X$ is in the subset L if and only if there exists a "witness" $\omega \in W$ such that $(x, w) \in R$. (If X, L, W, and R 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 $\Lambda[X, L, W, R] \in [I_\ell]$ chosen according to I_ℓ, and (2) the subset sampling algorithm takes as input 1^ℓ and an instance $\Lambda[X, L, W, R] \in [I_\ell]$, and returns $x \in L$ and a witness $\omega \in W$ for x. We say that a subset membership problem $\mathbf{M} = \{I_\ell\}_{\ell \geq 0}$ is hard if the following two distributions are computationally indistinguishable: $\{\Lambda \leftarrow I_\ell; x \xleftarrow{\$} L : (\Lambda, x)\}$ and $\{\Lambda \leftarrow I_\ell; x \xleftarrow{\$} X \setminus L : (\Lambda, x)\}$.

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 [11], we will occasionally introduce an arbitrary finite set E to extend the sets X and L in an instance $\Lambda[X, L, W, R] \in [I_\ell]$ of $\mathbf{M}$ into $X \times E$ and $L \times E$. If E is not required (e.g., for a smooth HPS in our construction), 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 $\Lambda = \Lambda[X, L, W, R]$ of $\mathbf{M}$ with a projective hash family $\mathbf{H} = (H, K, X \times E, L \times E, \Pi, S, \alpha)$, provides the following three 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 $\Lambda[X, L, W, R] \in [I_\ell]$, $\mathbf{H}$ is an ϵ-universal$_1$ (resp. ϵ-universal$_2$, ϵ-smooth) projective hash family.

Note that the homomorphic property of the underlying smooth projective hash family is required in our construction, where for all $k \in K$, and $x_1, x_2 \in X$, we have $H_k(x_1) + H_k(x_2) = H_k(x_1 + x_2) \in \Pi$ holds. Then, we call this smooth projective hash family homomorphic smooth projective hash family, and also call a smooth HPS homomorphic smooth HPS if the underlying smooth projective hash family has the homomorphic property.

Diverse Group System and Derived Projective Hash Family : Here, we recall the definition of diverse group systems introduced in [11], 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 $\mathbf{G}$ ([11, Definition 2]): Let $g_1, \ldots, g_d \in L$ be a set of generators of L (i.e., for all $x \in L$, there exist $\omega_1, \ldots, \omega_d \in \mathbb{Z}$ such that $x = \sum_{i=1}^{d} \omega_i g_i$). Set $S = \Pi^d$, and define $\alpha : K \to S$ by $\alpha(k) = (\phi(g_1), \ldots, \phi(g_d))$, where $\phi = H_k$. Note that $\mathbf{H}$ is a 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 [11, Theorem 2].

Lemma 1. *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 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* $\mathcal{KH}$-$\mathcal{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *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 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* $\mathsf{Eval}(sk_h, C_1, C_2) \in \mathcal{C}_{pk,M_1 \odot M_2}$*.*

If an operation $\odot$ is commutative, then the Eval algorithm is also called *commutative*, and we require that the distribution of $\mathsf{Eval}(sk_h, C_1, C_2)$ and that of $\mathsf{Eval}(sk_h, C_2, C_1)$ are identical. We instantiate DDH/DLIN/DBDH-based KH-PKEs with multiplicative homomorphic operations ($\odot := \times$), a DCR-based KH-PKE with additive homomorphic operations ($\odot := +$), and a DQR-based KH-PKE with XOR homomorphic operations ($\odot := \oplus$). Thus, our concrete instantiations are all commutative schemes.

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

Definition 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* $C \neq \perp$ *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* $\perp$ *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 $\perp$ 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 [6]. 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.

4 Generic Construction via Homomorphic Transitional Universal HPS

In this section, we give a generic construction of KH-PKE from an enhanced variant of universal HPS, which we call *homomorphic transitional universal HPS*. A homomorphic transitional universal HPS has, as its internal structure, a family of hash functions which we call *transitional universal projective hash family*.

4.1 Homomorphic Transitional Universal Projective Hash Families

Informally, a projective hash family $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ is said to be a transitional universal projective hash family if an index $k \in K$ for specifying a hash function from the family can be divided into two components as $(k', \widehat{k})$, and even if $\widehat{k}$ is exposed, it still yields the universal$_1$ property.

Definition 4 (Homomorphic Transitional (ϵ, ϵ')-Universal Projective Hash Families). *Let $\mathbf{H} = (H, K, X \times E, L \times E, \Pi, S, \alpha)$ be an ϵ-universal$_2$ hash family. We say that $\mathbf{H}$ is (ϵ, ϵ')-transitional if (1) The function index space K can be divided into two subspaces K_1 and K_2 such that $K = K_1 \times K_2$ (say $\overrightarrow{k} := (k', \widehat{k}) \in K_1 \times K_2$), and (2) Considering the probability space defined by choosing $k' \in K_1$ at random. Then for all $s \in S$, $x \in X \setminus L$, $\widehat{k} \in K_2$ and $\pi \in \Pi$, it holds that $\Pr_{k' \xleftarrow{\$} K_1}[H_{k',\widehat{k}}(x, e) = \pi \wedge \alpha(k', \widehat{k}) = s] \leq \epsilon' \cdot \Pr_{k' \xleftarrow{\$} K_1}[\alpha(k', \widehat{k}) = s]$. Especially, if ϵ and ϵ' are negligible, then $\mathbf{H}$ is called a transitional universal projective hash family. Moreover, if for all $(k', \widehat{k}) \in K_1 \times K_2$ and for all $(x_1, e_1), (x_2, e_2) \in X \times E$, $H_{k',\widehat{k}}(x_1 + x_2, e_1 + e_2)$ can be efficiently computed given $\widehat{k}$, $(x_1, e_1, H_{k',\widehat{k}}(x_1, e_1))$ and $(x_2, e_2, H_{k',\widehat{k}}(x_2, e_2))$, then $\mathbf{H}$ is called a homomorphic transitional universal projective hash family.*

Next, we show that the projective hash family [10, §7.43 Theorem 3] based on a diverse group system, satisfies the homomorphic transitional universal property as it is.

The Cramer-Shoup (CS) Projective Hash Family [10] : Let $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ be a universal$_1$ projective hash family derived from a diverse group system $\mathbf{G} = (\mathcal{H}, X, L, \Pi)$ (see the last paragraph of Section 2), and E be an abelian group. Then the CS projective hash family $\widehat{\mathbf{H}} = (\widehat{H}, \widehat{K} = K^{n+1}, X \times E, L \times E, \widehat{\Pi}, \widehat{S} = S^{n+1}, \widehat{\alpha})$ is constructed as follows: Let $\Gamma : X \times E \to \{0, \ldots, \widetilde{p} - 1\}^n$ be an injective function, where $\widetilde{p}$ is the smallest prime dividing $|X/L|$, and n is sufficiently large enough for Γ to be injective. For $\overrightarrow{k} = (k', k_1, \ldots, k_n) \in K^{n+1}$, $x \in X$, and $e \in E$, $\widehat{H}$ is defined as: $\widehat{H}_{k',\widehat{k}}(x, e) := H_{k'}(x) + \sum_{i=1}^{n} \gamma_i H_{k_i}(x)$, and $\widehat{\alpha}(k', \widehat{k}) = (\alpha(k'), \alpha(k_1), \ldots, \alpha(k_n))$, where $(\gamma_1, \ldots, \gamma_n) = \Gamma(x, e)$. Cramer and Shoup showed that the CS projective hash family $\widehat{\mathbf{H}}$ is $(1/\widetilde{p})$-universal$_2$. Note that since $H_k = \phi \in \mathsf{Hom}(X, \Pi)$, the basic projective hash family H derived from the diverse group system satisfies the homomorphic

property, namely for all $k \in K$, and $x_1, x_2 \in X$, we have $H_k(x_1) + H_k(x_2) = H_k(x_1 + x_2) \in \Pi$. Next, we show that it is in fact a homomorphic transitional universal projective hash family.

Lemma 2. *If an index $\overrightarrow{k} \in K^{n+1}$ is divided into $k' \in K$ and $\widehat{k} = (k_1, \ldots, k_n) \in K^n$, then the CS projective hash family $\widehat{\mathbf{H}}$ is a homomorphic transitional $(1/\widetilde{p}, 1/\widetilde{p})$-universal projective hash family.*

Proof: For $\overrightarrow{k} \in K^{n+1}$, fix $(k_1, \ldots, k_n) \in K^n$, and consider the probability space is defined by choosing $k' \in K$ at random. Then, $\widehat{\mathbf{H}}$ still provides the $(1/\widetilde{p})$-universal$_1$ property, because the projective hash family $\mathbf{H}$ is a $(1/\widetilde{p})$-universal$_1$ and the output of $\widehat{\mathbf{H}}$ is "masked" by the output of $\mathbf{H}$. Furthermore, for all $(x_1, e_1), (x_2, e_2) \in X \times E$, $H_{k',\widehat{k}}(x_1 + x_2, e_1 + e_2)$ can be efficiently computed given $\widehat{k} = (k_1, \ldots, k_n)$, $(x_1, e_1, H_{k',\widehat{k}}(x_1, e_1))$ and $(x_2, e_2, H_{k',\widehat{k}}(x_2, e_2))$ such that (1) compute $\sum_{i=1}^n \gamma_i^{(1)} H_{k_i}(x_1)$ and $\sum_{i=1}^n \gamma_i^{(2)} H_{k_i}(x_2)$, where $(\gamma_1^{(b)}, \ldots, \gamma_n^{(b)}) = \Gamma(x_b, e_b)$ for $b = 1, 2$, and (2) compute $\widehat{H}_{k',\widehat{k}}(x_1 + x_2, e_1 + e_2) \leftarrow \big(\widehat{H}_{k',\widehat{k}}(x_1, e_1) - \sum_{i=1}^n \gamma_i^{(1)} H_{k_i}(x_1)\big) + \big(\widehat{H}_{k',\widehat{k}}(x_2, e_2) - \sum_{i=1}^n \gamma_i^{(2)} H_{k_i}(x_2)\big) + \sum_{i=1}^n \gamma_i H_{k_i}(x_1 + x_2)$, where $(\gamma_1, \ldots, \gamma_n) = \Gamma(x_1 + x_2, e_1 + e_2)$. □

Finally we define the notion of homomorphic transitional universal HPS.

Definition 5 (Homomorphic Transitional Universal HPS). *Let $\mathbf{M} = \{I_\ell\}_{\ell \geq 0}$ be a subset membership problem. We say that a HPS $\mathbf{P}$ for $\mathbf{M}$ is homomorphic transitional (ϵ, ϵ')-universal if for all $\ell > 0$ and for all $\Lambda = \Lambda[X, L, W, R] \in [I_\ell]$, the projective hash family $\mathbf{H}$ that $\mathbf{P}$ associates with Λ is homomorphic transitional (ϵ, ϵ')-universal.*

4.2 Generic Construction of KH-PKE

Here, we give the proposed construction of a KH-PKE scheme based on a homomorphic transitional universal HPS given in the previous subsection, a homomorphic smooth projective HPS, and a universal$_2$ projective HPS. We note that all of the projective hash families used in our construction can be constructed from a diverse group system [11]. Therefore, our proposed construction is fairly generic.

We set $E = \Pi$ (Π is an abelian group, for which we use additive notation) and $\Gamma : X \times \Pi \to \Pi^n$ is an injective function, where n is a natural number which is sufficiently large so that Γ is injective. Let $\mathbf{M} = \{I_\ell\}_{\ell \geq 0}$ be a subset membership problem which specifies an instance description $\Lambda = \Lambda[X, L, W, R] \in [I_\ell]$. We will use the following three kinds of projective hash families $\mathbf{H}$, $\widehat{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$ and corresponding HPS (for $\mathbf{M}$). Using these building blocks, we construct a KH-PKE scheme as in Figure 1.

– $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ is a homomorphic smooth and projective hash family. Let $\mathbf{P} = (\mathsf{HPS.param}, \mathsf{HPS.priv}, \mathsf{HPS.pub})$ be a homomorphic smooth projective HPS for $\mathbf{M}$ which associates the instance Λ with $\mathbf{H}$.

$\mathsf{KeyGen}(1^\ell)$:
Pick $\Lambda = \Lambda[X, L, W, R] \leftarrow [I_\ell]$.
$(k, s) \leftarrow \mathsf{HPS.param}(1^\ell, \Lambda)$
$(\overrightarrow{\widehat{k}}, \widehat{s}) \leftarrow \widehat{\mathsf{HPS}}.\mathsf{param}(1^\ell, \Lambda)$
Parse $\overrightarrow{\widehat{k}} \in \widehat{K} = K \times K^n$ as $(k', \widehat{k})$
s.t. $k' \in K$ and
$\widehat{k} := (k_1, \ldots, k_n) \in K^n$
$(\widetilde{k}, \widetilde{s}) \leftarrow \widetilde{\mathsf{HPS}}.\mathsf{param}(1^\ell, \Lambda)$
$pk \leftarrow (s, \widehat{s}, \widetilde{s})$
$sk_d \leftarrow (k, (k', \widehat{k}), \widetilde{k})$; $sk_h \leftarrow (\widehat{k}, \widetilde{k})$
Return (pk, sk_d, sk_h)

$\mathsf{Dec}(sk_d, C)$:
Parse sk_d as $(k, (k', \widehat{k}), \widetilde{k})$
Parse C as $(x, e, \widehat{\pi}, \widetilde{\pi})$
$\widehat{\pi}' \leftarrow \widehat{\mathsf{HPS}}.\mathsf{priv}(1^\ell, \Lambda, (k', \widehat{k}), (x, e))$
$\widetilde{\pi}' \leftarrow \widetilde{\mathsf{HPS}}.\mathsf{priv}(1^\ell, \Lambda, \widetilde{k}, (x, e))$
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 $M \leftarrow e - \pi$

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

$\mathsf{Eval}(sk_h, C_1, C_2)$:
Parse sk_h as $(\widehat{k}, \widetilde{k})$ where $\widehat{k} = (k_1, \ldots, k_n)$
Parse C_b as $(x_b, e_b, \widehat{\pi}_b, \widetilde{\pi}_b)$ for $b = 1, 2$
$\widetilde{\pi}'_b \leftarrow \widetilde{\mathsf{HPS}}.\mathsf{priv}(1^\ell, \Lambda, \widetilde{k}, (x_b, e_b))$ for $b = 1, 2$
If $\widetilde{\pi}_1 \neq \widetilde{\pi}'_1$ or $\widetilde{\pi}_2 \neq \widetilde{\pi}'_2$ then return $\perp$
For $b = 1, 2$ Do:
$(\gamma_1^{(b)}, \ldots, \gamma_n^{(b)}) \leftarrow \Gamma(x_b, e_b)$
$H_{k_i}(x_b) \leftarrow \mathsf{HPS.priv}(1^\ell, \Lambda, k_i, x_b)$ for all $i \in [n]$
$\widehat{\pi}'_b \leftarrow \widehat{\pi}_b - \sum_{i \in [n]} \gamma_i^{(b)} H_{k_i}(x_b)$
End For
$x \leftarrow x_1 + x_2$; $e \leftarrow e_1 + e_2$
$(\gamma_1, \ldots, \gamma_n) \leftarrow \Gamma(x, e)$
$H_{k_i}(x) \leftarrow \mathsf{HPS.priv}(1^\ell, \Lambda, k_i, x)$ for all $i \in [n]$
$\widehat{\pi} \leftarrow \widehat{\pi}'_1 + \widehat{\pi}'_2 + \sum_{i \in [n]} \gamma_i H_{k_i}(x)$
$\widetilde{\pi} \leftarrow \widetilde{\mathsf{HPS}}.\mathsf{priv}(1^\ell, \Lambda, \widetilde{k}, (x, e))$
Return $C \leftarrow (x, e, \widehat{\pi}, \widetilde{\pi})$

Fig. 1. The proposed KH-PKE construction from HPS.

- $\widehat{\mathbf{H}} = (\widehat{H}, \widehat{K} = K \times K^n, X \times \Pi, L \times \Pi, \widehat{\Pi}, \widehat{S} = S^{n+1}, \widehat{\alpha})$ is the CS (homomorphic transitional universal) projective hash family that we showed in the previous subsection (with the index space $\widehat{K}$ is divided into $K_1 = K$ and $K_2 = K^n$). Let $\widehat{\mathbf{P}} = (\widehat{\mathsf{HPS}}.\mathsf{param}, \widehat{\mathsf{HPS}}.\mathsf{priv}, \widehat{\mathsf{HPS}}.\mathsf{pub})$ be a homomorphic transitional universal HPS for $\mathbf{M}$ which associates Λ with $\widehat{\mathbf{H}}$.
- $\widetilde{\mathbf{H}} = (\widetilde{H}, \widetilde{K}, X \times \Pi, L \times \Pi, \widetilde{\Pi}, \widetilde{S}, \widetilde{\alpha})$ is a universal$_2$ projective hash family. Let $\widetilde{\mathbf{P}} = (\widetilde{\mathsf{HPS}}.\mathsf{param}, \widetilde{\mathsf{HPS}}.\mathsf{priv}, \widetilde{\mathsf{HPS}}.\mathsf{pub})$ be a universal$_2$ HPS for $\mathbf{M}$ which associates Λ with $\widetilde{\mathbf{H}}$.

One might think that in the contraction, $\widetilde{\mathbf{H}}$ 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 an 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.

To see the correctness for the Eval algorithm, suppose that Eval receives correctly 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. Let $M = M_1 + M_2$. Then, by recall-

ing the homomorphic and transitional properties, the following holds: $\widehat{\pi}'_b = \widehat{\pi}_b - \sum_{i=1}^{n} \gamma_i^{(b)} H_{k_i}(x_b) = H_{k'}(x_b)$ for $b = 1, 2$, $e_1 + e_2 = (M_1 + M_2) + (H_k(x_1) + H_k(x_2)) = (M_1 + M_2) + H_k(x_1 + x_2) = (M_1 + M_2) + H_k(x)$, $\widehat{\pi} = \widehat{\pi}'_1 + \widehat{\pi}'_2 + \sum_{i=1}^{n} \gamma_i H_{k_i}(x) = H_{k'}(x_1) + H_{k'}(x_2) + \sum_{i=1}^{n} \gamma_i H_{k_i}(x) = H_{k'}(x_1 + x_2) + \sum_{i=1}^{n} \gamma_i H_{k_i}(x) = H_{k'}(x) + \sum_{i=1}^{n} \gamma_i H_{k_i}(x) = \widehat{H}_{k',\widehat{k}}(x, e)$, which means that $C = (x, e, \widehat{\pi}, \widetilde{\pi})$ is a valid ciphertext of $M := M_1 + M_2$.

Since all of the projective hash families used in our construction can be constructed from a diverse group system, from the result of [19] (where CPA-secure homomorphic PKE (with cyclic ciphertext space) implies diverse group systems), the following corollary is given.

Corollary 1. *KH-PKE is implied by CPA-secure homomorphic PKE with cyclic ciphertext space.*

The proof of the following theorem is given in the Appendix.

Theorem 1. *Our construction is KH-CCA-secure if* $\mathbf{M}$ *is a hard subset membership problem,* $\mathbf{P}$ *is a homomorphic smooth projective HPS for* $\mathbf{M}$*,* $\widehat{\mathbf{P}}$ *is a homomorphic transitional universal HPS for* $\mathbf{M}$*, and* $\widetilde{\mathbf{P}}$ *is a universal*$_2$ *HPS for* $\mathbf{M}$*.*

5 Practical KH-PKE Construction from DDH

In this section, we present an efficient DDH-based KH-PKE construction. This scheme is not a mere combination of the generic construction of KH-PKE in Section 4 and the transitional HPS from DDH (which will appear in the full version), but introduces additional techniques for enhancing efficiency. Remarkably, efficiency of our scheme is only slightly lower than the Cramer-Shoup encryption in spite of its complicated functionality. In particular, ciphertext length of our scheme is only ℓ-bit larger than that of the Cramer-Shoup scheme, where ℓ is the security parameter. For example, 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).

5.1 Techniques for Improving Efficiency

Before going into the concrete construction of our DDH-based KH-PKE scheme, we briefly explain two additional techniques for enhancing efficiency which are not mentioned in the previous sections. Both these techniques employ target collision resistant (TCR) hash functions [10], and can also be applicable to other various (standard) PKE schemes.

The first technique is just the same as the popular method for transforming hash-free variant of the Cramer-Shoup scheme into the TCR-based one (i.e., the standard Cramer-Shoup scheme). Due to it, the size of the public key is significantly reduced.

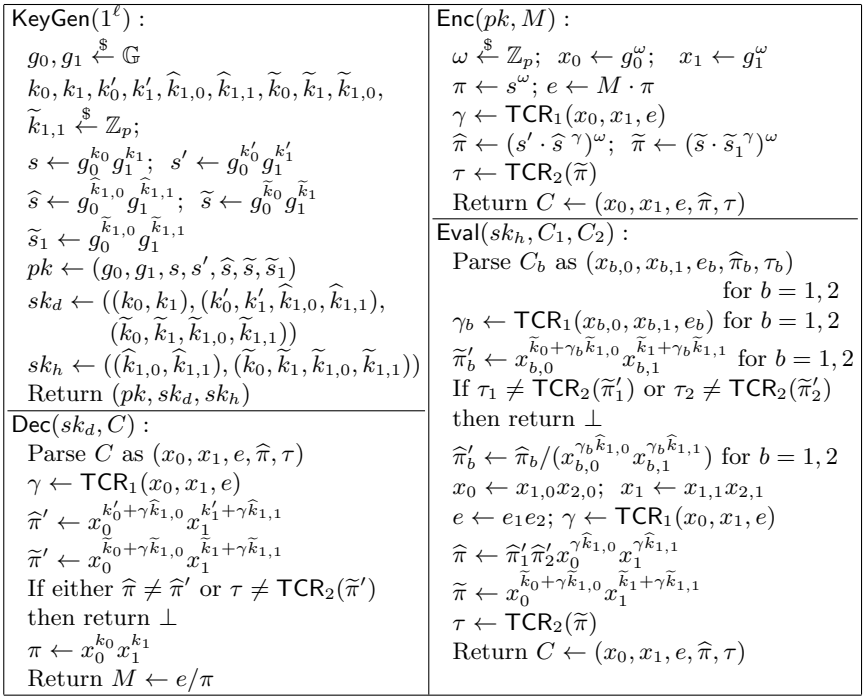

Fig. 2. Our DDH-based KH-PKE Scheme.

The second technique is to compress the redundant part of the ciphertext by using a TCR hash function. Interestingly, our security proof still works even if one of ciphertext components (specifically, a component for validity checking upon the homomorphic operation) is hashed to be a smaller value. 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 literatures. 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 [23] 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 [7], Hanaoka-Kurosawa [18], and Kiltz schemes [21]. 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. [5, 18, 20].

5.2 Practical KH-PKE from DDH

Here, we give a description of our KH-PKE instantiation (using our technique of reducing the ciphertext size). First, we define the DDH assumption as follows.

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, where $|C| - |M|$ denotes ciphertext overhead, $|\mathbb{G}|$ denotes the size of the underlying group element $\mathbb{G}$, and exp denotes exponentiation. We count 1 multi-exp equals as 1.2 regular exp, and the size of MAC and the hashed value of TCR as $0.5|\mathbb{G}|$.

	$\|C\| - \|M\|$	Cost (Enc)	Cost (Dec)	KH property
CS [9]	$3\|\mathbb{G}\|$	4.2 exp	2.4 exp	No
KD [23]	$2.5\|\mathbb{G}\|$	3.2 exp	1.2 exp	No
KD+CS-lite Double Enc	$5.5\|\mathbb{G}\|$	7.2 exp	3.6 exp	No?
Our DDH-based KH-PKE	$3.5\|\mathbb{G}\|$	5.4 exp	3.6 exp	Yes

Definition 6 (The Decisional Diffie-Hellman (DDH) Assumption). *Let $\mathbb{G}$ be a group with prime order p. We say that the DDH assumption holds in $\mathbb{G}$ if the advantage $Adv^{DDH}_{\mathbb{G},\mathcal{A}}(1^\ell) := |\Pr[\mathcal{A}(g_0, g_1, g_0^r, g_1^r) = 0] - \Pr[\mathcal{A}(g_0, g_1, g_0^r, g_1^{r'}) = 0]|$ is negligible for any PPT algorithm $\mathcal{A}$, where g_0 and g_1 are randomly chosen from $\mathbb{G}$, and r and r' are randomly chosen from $\mathbb{Z}_p$.*

Our DDH-Based KH-PKE Scheme : Let $\mathsf{TCR}_1 : \mathbb{G}\times\mathbb{G}\times\mathbb{G} \to \mathbb{Z}_p$ and $\mathsf{TCR}_2 : \mathbb{G} \to \{0,1\}^{\log p/2}$ be TCR hash functions. We give our DDH-based KH-PKE scheme in Figure 2. Here, we explain the usage of $sk_h = ((\widehat{k}_{1,0}, \widehat{k}_{1,1}), (\widetilde{k}_0, \widetilde{k}_1, \widetilde{k}_{1,0}, \widetilde{k}_{1,1}))$. $\widehat{\pi}'_1 = \widehat{\pi}_1/(x_{1,0}^{\gamma_1\widehat{k}_{1,0}} x_{1,1}^{\gamma_1\widehat{k}_{1,1}}) = x_{1,0}^{k'_0} x_{1,1}^{k'_1}$ and $\widehat{\pi}'_2 = \widehat{\pi}_2/(x_{2,0}^{\gamma_2\widehat{k}_{1,0}} x_{2,1}^{\gamma_2\widehat{k}_{1,1}}) = x_{2,0}^{k'_0} x_{2,1}^{k'_1}$ hold using $(\widehat{k}_{1,0}, \widehat{k}_{1,1})$. So, $\widehat{\pi} \leftarrow \widehat{\pi}'_1\widehat{\pi}'_2 x_0^{\gamma\widehat{k}_{1,0}} x_1^{\gamma\widehat{k}_{1,1}} = x_0^{k'_0+\gamma\widehat{k}_{1,0}} x_1^{k'_1+\gamma\widehat{k}_{1,1}}$ holds. Therefore, the Eval algorithm works. The other keys $(\widetilde{k}_0, \widetilde{k}_1, \widetilde{k}_{1,0}, \widetilde{k}_{1,1})$ (and TCR_2) are used for computing $\widetilde{\pi}'_1$ (resp. $\widetilde{\pi}'_2$) to check the validity of C_1 (resp. C_2).

The following theorem can be proved in the same way as Theorem 1.

Theorem 2. *The proposed DDH-based KH-PKE scheme is KH-CCA-secure if the DDH assumption holds, and TCR_1 and TCR_2 are TCR hash functions.*

In Table 1, we give an efficiency comparison of our DDH-based KH-PKE scheme with the CS PKE [9], the KD PKE [23], and the naive construction (See Section 1). We note that these three schemes do not yield keyed-homomorphic property and/or 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 both ciphertext overhead and computational costs. Especially, ciphertext overhead of our scheme is only ℓ-bit longer than that of the CS scheme for ℓ-bit security. 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.

Acknolwdgement : We thank anonymous reviewers and the members of Shin-Akarui-Angou-Benkyou-Kai for their helpful comments. The 4th and 5th authors are supported by a JSPS Fellowship for Young Scientists. This work was supported by JSPS KAKENHI Grant Number 24700009.

References

1. J. H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *EUROCRYPT*, pages 83–107, 2002.
2. M. Barbosa and P. Farshim. Delegatable homomorphic encryption with applications to secure outsourcing of computation. In *CT-RSA*, pages 296–312, 2012.
3. D. Bernhard, V. Cortier, O. Pereira, B. Smyth, and B. Warinschi. Adapting Helios for provable ballot privacy. In *ESORICS*, pages 335–354, 2011.
4. D. Boneh, G. Segev, and B. Waters. Targeted malleability: homomorphic encryption for restricted computations. In *ICTS*, pages 350–366, 2012.
5. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, pages 207–222, 2004.
6. R. Canetti, H. Krawczyk, and J. B. Nielsen. Relaxing chosen-ciphertext security. In *CRYPTO*, pages 565–582, 2003.
7. D. Cash, E. Kiltz, and V. Shoup. The twin Diffie-Hellman problem and applications. In *EUROCRYPT*, pages 127–145, 2008.
8. M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn. Malleable proof systems and applications. In *EUROCRYPT*, pages 281–300, 2012.
9. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.
10. 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/.
11. 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.
12. 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.
13. T. E. Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.
14. C. Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.
15. C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.
16. S. Goldwasser and S. Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *STOC*, pages 365–377, 1982.
17. J. Groth. Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In *TCC*, pages 152–170, 2004.
18. G. Hanaoka and K. Kurosawa. Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. In *ASIACRYPT*, pages 308–325, 2008.
19. B. Hemenway and R. Ostrovsky. On homomorphic encryption and chosen-ciphertext security. In *Public Key Cryptography*, pages 52–65, 2012.

20. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, pages 581–600, 2006.
21. E. Kiltz. Chosen-ciphertext secure key-encapsulation based on gap hashed Diffie-Hellman. In *PKC*, pages 282–297, 2007.
22. E. Kiltz, K. Pietrzak, M. Stam, and M. Yung. A new randomness extraction paradigm for hybrid encryption. In *EUROCRYPT*, pages 590–609, 2009.
23. K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO*, pages 426–442, 2004.
24. 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.
25. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238, 1999.
26. M. Prabhakaran and M. Rosulek. Rerandomizable RCCA encryption. In *CRYPTO*, pages 517–534, 2007.
27. M. Prabhakaran and M. Rosulek. Homomorphic encryption with CCA security. In *ICALP (2)*, pages 667–678, 2008.
28. 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/.
29. V. Shoup. A proposal for an ISO standard for public key encryption. Cryptology ePrint Archive, Report 2001/112, 2001. http://eprint.iacr.org/.

Appendix: Proof of Theorem 1

Proof. Let $\mathcal{A}$ be an adversary who breaks KH-CCA security. Then, we construct an algorithm $\mathcal{B}$ that can break the hardness of $\mathbf{M}$. To later calculate the concrete advantage of $\mathcal{A}$, let $\epsilon(\ell)$, $\widehat{\epsilon}(\ell)$, and $\widetilde{\epsilon}(\ell)$ be negligible functions such that $\mathbf{P}$ be $\epsilon(\ell)$-smooth, and $\widehat{\mathbf{P}}$ be homomorphic transitional $(\widehat{\epsilon}(\ell), \widehat{\epsilon}(\ell))$-universal, and $\widetilde{\mathbf{P}}$ be $\widetilde{\epsilon}(\ell)$-universal$_2$.

We describe how $\mathcal{B}$ simulates the KH-CCA experiment for $\mathcal{A}$. First, $\mathcal{B}$ takes as input 1^ℓ along with $\Lambda[X, L, W, R] \in [I_\ell]$ and $x^* \in X$. $\mathcal{B}$ runs $(pk, sk_d, sk_h) \leftarrow \mathsf{KeyGen}(1^\ell)$ as usual using the given value of Λ, where $pk = (s, \widehat{s}, \widetilde{s})$, $sk_d = (k, \overrightarrow{k}, \widetilde{k}) = (k, (k', \widehat{k}), \widetilde{k})$, and $sk_h = (\widehat{k}, \widetilde{k})$. $\mathcal{B}$ sends pk to $\mathcal{A}$.

In find stage, $\mathcal{B}$ answers for each query as follows: For a decryption query C, $\mathcal{B}$ runs $\mathsf{Dec}(sk_d, C)$ as usual using sk_d, and returns the result of the decryption algorithm. For an evaluation query (C_1, C_2), $\mathcal{B}$ runs $\mathsf{Eval}(sk_h, C_1, C_2)$ as usual using sk_h, and returns the result of the evaluation algorithm. For the reveal homomorphic key query, $\mathcal{B}$ returns $sk_h = (\widehat{k}, \widetilde{k})$. In the challenge phase, $\mathcal{A}$ sends (M_0^*, M_1^*) to $\mathcal{B}$. $\mathcal{B}$ chooses $\beta \xleftarrow{\$} \{0, 1\}$, and computes $\pi^* \leftarrow H_k(x^*)$ using the private evaluation algorithm, $e^* = \pi^* + M_\beta^*$, and $\widehat{\pi}^* \leftarrow \widehat{H}_{k', \widehat{k}}(x^*, e^*)$ and $\widetilde{\pi}^* \leftarrow \widetilde{H}_{\widetilde{k}}(x^*, e^*)$ using the private evaluation algorithm, and sends $C^* = (x^*, e^*, \widehat{\pi}^*, \widetilde{\pi}^*)$ to $\mathcal{A}$. In addition, $\mathcal{B}$ sets a ciphertext dictionary $\mathcal{D}$ such that $\mathcal{D} = \{C^*\}$. In guess stage, $\mathcal{B}$ answers for each query as follows: For a decryption query C, if $C \in \mathcal{D}$, then return $\perp$. Otherwise, $\mathcal{B}$ runs $\mathsf{Dec}(sk_d, C)$ as usual using sk_d, and returns the result of the decryption algorithm. For an evaluation query (C_1, C_2), $\mathcal{B}$ runs $\mathsf{Eval}(sk_h, C_1, C_2)$ as usual using sk_h, and returns the result of the

evaluation algorithm. If either $C_1 \in \mathcal{D}$ or $C_2 \in \mathcal{D}$, then $\mathcal{B}$ updates $\mathcal{D} \leftarrow \mathcal{D} \cup \{C_3\}$, where $C_3 = \mathsf{Eval}(sk_h, C_1, C_2)$. Note that if $C_3 = \perp$, then $\mathcal{B}$ does not update the dictionary $\mathcal{D}$. For the reveal homomorphic key query, $\mathcal{B}$ returns $sk_h = (\widehat{k}, \widetilde{k})$. Finally, $\mathcal{A}$ outputs a guessing bit β'. $\mathcal{B}$ outputs 1 if $\beta = \beta'$, and 0 otherwise.

Next, we define two experiments according to whether $x^* \in L$ or $x^* \in X \setminus L$. In the first experiment, $\mathcal{B}$ is given (Λ, x^*), where $\Lambda[X, L, W, R] \in [I_\ell]$ and $x^* \in L$. Let T'_ℓ be the event that $\mathcal{B}$ outputs 1 in this experiment. In the second experiment, $\mathcal{B}$ is given (Λ, x^*), where $\Lambda[X, L, W, R] \in [I_\ell]$ and $x^* \in X \setminus L$. Let T_ℓ be the event that $\mathcal{B}$ outputs 1 in this experiment. By definition of the subset membership problem, the advantage of $\mathcal{B}$ is defined as $\mathsf{AdvDist}(\ell) := |\Pr[T_\ell] - \Pr[T'_\ell]|$. Let $Q_{dec}(\ell)$ be the number of decryption queries and $Q_{eval}(\ell)$ be the number of evaluation queries. In the case of $x^* \in L$, the simulation of the KH-CCA game for the adversary $\mathcal{A}$ is perfect. Thus, we get $\left|\Pr[T'_\ell] - \frac{1}{2}\right| \geq Adv^{KH\text{-}CCA}_{\mathsf{KH\text{-}PKE},\mathcal{A}}(\ell)$ In the case of $x^* \in X \setminus L$, we define the sequences of games. We denote $T^{(0)}_\ell$, $T^{(1)}_\ell$, and $T^{(2)}_\ell$ as the event that $\mathcal{B}$ outputs 1 in the game 0, 1, and 2, respectively.

Game 0: The same as the KH-CCA simulation.

Game 1: Recall that in Game 0, the decryption oracle (and the evaluation oracle also) rejects a query $(x, e, \widehat{\pi}, \widetilde{\pi})$ if either $\widehat{H}_{k',\widehat{k}}(x, e) \neq \widehat{\pi}$ or $\widetilde{H}_{\widetilde{k}}(x, e) \neq \widetilde{\pi}$. In this game, we make these oracles reject a query that contains a ciphertext $(x, e, \widehat{\pi}, \widetilde{\pi})$ satisfying $x \notin L$. Let F_2 be the event that these oracles reject a query $(x, e, \widehat{\pi}, \widetilde{\pi})$ with $x \notin L$, but either $\widehat{H}_{k',\widehat{k}}(x, e) = \widehat{\pi}$ or $\widetilde{H}_{\widetilde{k}}(x, e) = \widetilde{\pi}$ holds. In the find phase, $\widehat{\alpha}(k', \widehat{k}) = \widehat{s}$ and $\widetilde{\alpha}(\widetilde{k}) = \widetilde{s}$ are fixed. Then, the probability that $\widehat{H}_{k',\widehat{k}}(x, e) = \widehat{\pi}$ is at most $\widehat{\epsilon}(\ell)$, since $\widehat{\mathbf{H}}$ is a $\widehat{\epsilon}$-universal$_2$ (or $\widehat{\epsilon}$-universal$_1$ projective, if $\mathcal{A}$ has been an insider via the RevHK oracle) hash family, and the probability that $\widetilde{H}_{\widetilde{k}}(x, e) = \widetilde{\pi}$ is at most $\widetilde{\epsilon}(\ell)$, since $\widetilde{\mathbf{H}}$ is a $\widetilde{\epsilon}$-universal$_2$ hash family. In the challenge phase, $\widehat{\pi}^* = \widehat{H}_{k',\widehat{k}}(x^*, e^*)$ and $\widetilde{\pi}^* = \widetilde{H}_{\widetilde{k}}(x^*, e^*)$ are fixed. After this, in the guess stage, the probability that $\widehat{H}_{k',\widehat{k}}(x, e) = \widehat{\pi}$ is at most $\widehat{\epsilon}(\ell)$, since $\widehat{\mathbf{H}}$ is a $\widehat{\epsilon}$-universal$_2$. Note that if $\mathcal{A}$ has been an insider, then $\mathcal{A}$ does not issue the decryption query. In addition, the probability that $\widetilde{H}_{\widetilde{k}}(x, e) = \widetilde{\pi}$ is at most $\widetilde{\epsilon}(\ell)$, since $\widetilde{\mathbf{H}}$ is a $\widetilde{\epsilon}$-universal$_2$. To sum up, we get $\Pr[F_2] \leq Q_{dec}(\ell)(\widehat{\epsilon}(\ell) + \widetilde{\epsilon}(\ell)) + 2Q_{eval}(\ell)\widetilde{\epsilon}(\ell)$. The term $2Q_{eval}(\ell)$ is derived from the fact that an evaluation query contains two ciphertexts. In addition, from the fact that Game 0 and Game 1 are identical if the event F_2 does not occur, we get $\left|\Pr[T^{(1)}_\ell] - \Pr[T^{(0)}_\ell]\right| \leq \Pr[F_2] \leq Q_{dec}(\ell)(\widehat{\epsilon}(\ell) + \widetilde{\epsilon}(\ell)) + 2Q_{eval}(\ell)\widetilde{\epsilon}(\ell)$.

Game 2: In this game, $\mathcal{B}$ chooses $\pi^* \xleftarrow{\$} \Pi$ (instead of computing $\pi^* = H_k(x^*)$) and computes $e^* = \pi^* + M^*_\beta$. Since $\mathbf{H}$ is an $\epsilon(\ell)$-smooth projective hash family and β is hidden by π^*, we get $\left|\Pr[T^{(2)}_\ell] - \Pr[T^{(1)}_\ell]\right| \leq \epsilon(\ell)$ and $\Pr[T^{(2)}_\ell] = \frac{1}{2}$. By combining the inequalities, we get $Adv^{KH\text{-}CCA}_{\mathsf{KH\text{-}PKE},\mathcal{A}}(\ell) \leq \mathsf{AdvDist}(\ell) + Q_{dec}(\ell)\widetilde{\epsilon}(\ell) + (Q_{dec}(\ell) + 2Q_{eval}(\ell))\widetilde{\epsilon}(\ell) + \epsilon(\ell)$, which is negligible. □