Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems | School of Computing and Information Systems

4-2017

# Related-key secure key encapsulation from extended computational bilinear Diffie–Hellman

Brandon QIN

Shengli LIU

Shifeng SUN

Robert H. DENG
*Singapore Management University*, robertdeng@smu.edu.sg

Dawu GU

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

 Part of the Hardware Systems Commons, and the Software Engineering Commons

## Citation

QIN, Brandon; LIU, Shengli; SUN, Shifeng; DENG, Robert H.; and GU, Dawu. Related-key secure key encapsulation from extended computational bilinear Diffie–Hellman. (2017). *Information Sciences*. 406-407, 1-11.
**Available at:** https://ink.library.smu.edu.sg/sis_research/3678

This Journal Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

**Published in Information Sciences, 2017 April, Volume 406-407, Pages 1-11**
**http://doi.org/10.1016/j.ins.2017.04.018**

# Related-key secure key encapsulation from extended computational bilinear Diffie–Hellman

Baodong Qin[a,b,*], Shengli Liu[c], Shifeng Sun[c], Robert H. Deng[d], Dawu Gu[c]

[a] *National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, PR China*
[b] *State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, PR China*
[c] *Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, PR China*
[d] *School of Information Systems, Singapore Management University, Singapore 178902, Singapore*

A B S T R A C T

As a special type of fault injection attacks, Related-Key Attacks (RKAs) allow an adversary to manipulate a cryptographic key and subsequently observe the outcomes of the cryptographic scheme under these modified keys. In the real life, related-key attacks are already practical enough to be implemented on cryptographic devices. To avoid cryptographic devices suffering from related-key attacks, it is necessary to design a cryptographic scheme that resists against such attacks. This paper proposes an efficient RKA-secure Key Encapsulation Mechanism (KEM), in which the adversary can modify the secret key $sk$ to any value $f(sk)$, as long as, $f$ is a polynomial function of a bounded degree $d$. Especially, the polynomial-RKA security can be reduced to a hard *search* problem, namely $d$-extended computational Bilinear Diffie-Hellman (BDH) problem, in the standard model. Our construction essentially refines the security of Haralambiev et al.'s BDH-based KEM scheme from chosen-ciphertext security to related-key security. The main technique applied in our scheme is the re-computation of the public key in the decryption algorithm so that any (non-trivial) modification to the secret key can be detected.

*Keywords:*
Key-encapsulation mechanism
Related-key attacks
BDH

## 1. Introduction

To characterize security for a cryptographic scheme, we generally set up a security model to formalize the attacks from adversaries, and define security notions for the scheme. Most of security models assume that any adversary has only a black-box access to the algorithms of the scheme. In other words, the adversary has no information about the internal states of the algorithms, hence is not able to modify the secret key. While this assumption might be reasonable for some settings, there still exists a gap between the idealized assumption and practices. For instance, in the context of side-channel attacks on a hardware device, an adversary can obtain partial information about the secret key, by means of timing [34], "cold-boot" [25] and fault injection [9,11], etc. Relate-Key Attacks (RKAs) are just a type of fault injection attacks. By implementing RKAs on a device, an adversary is not only capable of modifying the original secret key, but also able to observe the output of the device under these modified keys. Biham [8] and Knudsen [33] gave the first security analysis for some block-ciphers in the scenario of related-key attacks. At the same time, RKAs also impose great threats on public key

* Corresponding author.
*E-mail addresses:* qinbaodong@foxmail.com (B. Qin), slliu@sjtu.edu.cn (S. Liu), crypto99@sjtu.edu.cn (S. Sun), robertdeng@smu.edu.sg (R.H. Deng), dwgu@sjtu.edu.cn (D. Gu).

cryptosystems. For example, the well-known CRT-based RSA signature scheme [11] can be completely broken, even if only one bit of the signing key is tampered by the adversary. However, it seems difficult to define a formal security model to capture related-key attacks, as it is hard to precisely depict the adversary's tampering capacity. Recently, Bellare and Kohno made an effort on modelling RKAs. They defined a set of functions $\Phi$ to capture the adversary's tampering ability to the secret key. Informally, a $\Phi$-RKA adversary is able to choose any function $\phi \in \Phi$ to modify the secret key $s$ in a device to a new key $s' = \phi(s)$, then observe the input/output of the device under the new key $s'$. In other words, the adversary knows the relationship between the original key and the modified key, though it may not know their real values. For concrete cryptographic primitives, the $\Phi$-RKA adversary may also have other abilities, e.g., access to a decryption oracle in the setting of public-key encryption, or access to a signing oracle in the setting of digital signature. In this paper, we focus on Key-Encapsulation Mechanism (KEM), and review KEM with its RKA security [5,41].

### 1.1. RKA security

Normally, a KEM scheme consists of a system parameter generation algorithm, a public key/secret key generation algorithm, a session key encryption algorithm and a decryption algorithm. The RKA security model assumes that the system parameter and those algorithms (program code) cannot be manipulated by the adversary, as they are fixed beforehand and independent of users. However, the adversary is able to tamper with the secret key (decryption key). The power of an RKA adversary is parameterized by a class $\Phi$ of functions, also called Related-Key Derivation (RKD) functions, which is used by the adversary to modify the secret key $sk^*$. Informally, a KEM scheme is $\Phi$-RKA secure if the challenge ciphertext $C^*$ remains secure even if the adversary obtains the decryption of any ciphertext $C$ under any modified secret key $\phi(sk^*)$ as long as $\phi \in \Phi$, and $(\phi(sk^*), C)$ does not match the pair of challenge secret key and ciphertext $(sk^*, C^*)$.

The main line of research in RKA security is to construct cryptographic primitives that are provably secure against *larger* classes of RKD functions. Many practical constructions have been proposed with respect to linear functions [4,5,18,19,41], affine functions [2,31,32,35], polynomials [2,7,35] and even any invertible functions [21,40]. However, to achieve larger classes of RKD functions, these schemes usually rely on much stronger assumptions, such as the $d$-Extended Decisional Bilinear Diffie–Hellman ($d$-EDBDH) used in [7]. In general, the harder a problem is, the stronger the assumption associated with the problem is, and the adversary will take more time to break it. It is believed that computational assumptions from search problems are a much weaker class of assumptions than the corresponding decisional versions. For example, in bilinear pairing groups, the Decisional Diffie–Hellman (DDH) assumption does not hold anymore, but its computational version, namely Computational Diffie–Hellman (CDH) problem, appears to be hard. So it is prefer to design a scheme based on much weaker assumptions associated with hard search problems. By this observation, recent development on security of PKE against chosen-ciphertext attack (CCA) sees many practical constructions from various search problems, such as factoring [29] and CDH problems [14]. However, there are still very few RKA-secure schemes from hard search problems, especially for large class of RKD functions. In 2012, Wee [41] gave the first RKA-secure PKE scheme from factoring. But the security was only proved for linear RKD functions. It seems that RKA security from search problems is very hard to obtain especially for non-linear RKD functions.

### 1.2. Our contribution

Our main contribution is a non-linear RKA secure KEM scheme from a hard search problem, namely $d$-Extended computational Bilinear Diffie-Hellman ($d$-EBDH), in the standard model. The $d$-EBDH problem states that

$$\text{Given } (g, g^{\alpha}, \boxed{g^{\alpha^2}, \ldots, g^{\alpha^d}}, g^{\beta}, g^{\gamma}), \quad \text{to compute } T = e(g,g)^{\alpha\beta\gamma}.$$

This is a computational version of the $d$-EDBDH problem used by Bellare, Paterson and Thomson for constructing RKA secure IBE and KEM. The $d$-EDBDH problem is hard in the generic group model [39]. Obviously, the $d$-EBDH problem is at least as hard as the $d$-EDBDH problem and might be even harder.

### 1.3. Overview of our technique

Recall that in the traditional CCA-security, one of the technical difficulties in the security proof is how to simulate the decryption queries without the target secret key. To solve this problem, except for the approach using hash proof system proposed by Cramer and Shoup [16], most of the CCA-secure PKE/KEM schemes e.g., [13,27,28,37], used the "all-but-one" technique, which allows one to set up the system parameter and public key so that one can decrypt all ciphertexts except for one single challenge ciphertext. If CCA security is lifted to RKA security, the problem becomes more challenging since we need to answer decryption queries under not only the target secret key, but also many related keys. To solve this problem, our strategy is the use of a "seed" malleable KEM scheme, together with the "all-but-one" technique. Here, the "seed" is a middle value generated during the decryption. But it can be used to correctly recover the final session key. In addition, validity of a ciphertext is publicly checkable and hence is independent of the secret key. As a result, we can derive a seed from a valid ciphertext under the original key. By the "seed" malleability, we further compute the right seed w.r.t. the modified key and hence the final session key. This strategy works well unless an adversary query decryption with the

| | BHJK-KEM | Variant BHJK-KEM |
|---|---|---|
| Parameter $sp$ | $(g, X, X', Z_1, \ldots, Z_n, R)$ | $(g, X, X', Z_1, \ldots, Z_n, R)$ |
| $(pk, sk)$ | $(g^\alpha, \alpha)$ | $(g^\alpha, \alpha)$ |
| Encap. $(C_0, C_1)$ | $(g^r, (X^t X')^r)$ // $t = \mathsf{TCR}(g^r)$ | $(g^r, (X^t X')^r)$ // $t = \mathsf{TCR}(g^r, \boxed{pk})$ |
| $K = (K_1, \ldots, K_n)$ | $K_i = f_{gl}(\hat{e}(Z_i^r, pk), R)$ | $K_i = f_{gl}(\hat{e}(Z_i^r, pk), R)$ |
| Decap. | $t = \mathsf{TCR}(C_0)$ | $t = \mathsf{TCR}(C_0, \boxed{g^\alpha})$ |
| | $\hat{e}(C_0, X^t X') \stackrel{?}{=} \hat{e}(g, C_1)$ | $\hat{e}(C_0, X^t X') \stackrel{?}{=} \hat{e}(g, C_1)$ |
| $K = (K_1, \ldots, K_n)$ | $K_i = f_{gl}(\hat{e}(C_0^\alpha, Z_i), R)$ | $K_i = f_{gl}(\hat{e}(C_0^\alpha, Z_i), R)$ |

**Fig. 1.** CCA-secure HJKS-KEM and RKA-secure variant of HJKS-KEM.

challenge ciphertext under a key that is distinct from the original secret key. To solve this problem, we use a simple trick of hashing the public key into the ciphertext. The secret key is bound with the public key. If the secret key is modified, so is the public key and hence the ciphertext. This prevents the adversary from reusing the challenge ciphertext.

A candidate for CCA-secure KEM scheme in our construction is a BDH-based KEM scheme proposed by Haralambiev, Jager, Kiltz, and Shoup [26] (denoted by HJKS-KEM). We review it as well as our scheme (a variant HJKS-KEM) in Fig. 1.

In Fig. 1, $f_{gl}(\cdot, R)$ is a Goldreich–Levin hard-core function and TCR is a target collision-resistant hash function. The seed used in our proof is a sequence of values $Z_i^r$. The "seed" malleability is defined as follows: given $Z_i^r$ and $(g, g^\alpha, \ldots, g^{\alpha^d})$, for any polynomial RKD function $\phi_{a_0,a_1,\ldots,a_d}(x) = \sum_{i=0}^{d} a_i x^i$, it is feasible to derive $\hat{e}(g^{\sum_{i=0}^{d} a_i \alpha^i}, Z_i^r)$ which equals the right "seed" $\hat{e}(C_0^{\phi_{a_0,a_1,\ldots,a_d}(\alpha)}, Z_i)$ w.r.t. RKD function $\phi_{a_0,a_1,\ldots,a_d}$. From these right "seeds", it is easy to recover the right encapsulated key $K$. In the proof, to derive the "seeds" $Z_i^r$ without the knowledge of $\alpha$, we employ the "all-but-one" technique, similar to that in the proof of CCA security of HJKS-KEM scheme. Note that, the elements $(g^{\alpha^2}, \ldots, g^{\alpha^d})$ are only used in the proof, and hence they do not amplify the key sizes of the original HJKS-KEM scheme. In addition, the variant of HJKS-KEM scheme is as efficient as the original HJKS-KEM scheme except for adding one more exponent operation in *decryption*.

**Discussion.** Very recently, Cui et al. [20] studied the relations between robustness (discussed in TCC 2010 by Abdalla et al. [1]) and RKA security under public-key encryption. One of their results [20, Theorem 4] shows that a completely robust PKE scheme is also RKA secure with respect to the *restricted* RKD functions. Moreover, a CCA-secure PKE can be made to completely robust via a commitment scheme. The restricted RKD functions should satisfy the following three properties:

- Malleability. Given an RKD function $\phi$ and a decryption key $sk$, there should exist an algorithm which outputs a decryption key sk' under $\phi$ that is distributed identically to the output of $\phi(sk)$.
- Compatibility. Given an RKD function $\phi$, an encryption $pk$, and a decryption $sk$, there should exist a key generation algorithm outputting a key pair $(pk', sk') \neq (pk, sk)$, where $sk' = \phi(sk)$ and the corresponding $pk'$ can be efficiently generated from $\phi$ and $pk$.
- Collision resistance. Given RKD functions $\phi_1$, $\phi_2$, and a decryption key $sk$, $\phi_1(sk)$ should not be equal to $\phi_2(sk)$ if $\phi_1 \neq \phi_2$.

The collision resistance actually requires the RKD functions be claw-free (introduced by Bellare and Cash at CRYPTO 2010 [4]). It seems that the key generation algorithm of the BHJK-KEM scheme meets the above requirements with respect to polynomial RKD functions under the $d$-EBDH assumption. However, the polynomial functions are not claw-free. We believe that the concrete instantiation of Cui et al.'s framework for polynomial functions can remove this restriction of collision resistance. The reason is as follows. For any bounded-degree polynomial over finite field, there exists efficient algorithm to compute all its roots. If $\phi_1(sk) = \phi_2(sk)$, we may find the target decryption key $sk$ from the roots of the polynomial equation $f(x) = \phi_1(x) - \phi_2(x)$=0. Under the $d$-EBDH assumption, the restricted RKD functions may not be larger than $d$-degree polynomials, as the property of compatibility requires the encryption key $pk'$ corresponding to $\phi(sk)$ be publicly computed from the challenge $pk$ (including the system parameter) and the RKD function $\phi$.

### *1.4. Related work*

In 2003, Bellare and Kohno [6] initiated a theoretical investigation of RKA security for PRFs and PRPs. Since then, the study of RKA security has been extended to other primitives [3–5,7,22,24,35,36]. Particularly, in 2011, Bellare, Cash and Miller [5] showed the relations among RKA primitives of PRFs, IBE, PKE, signature, symmetric encryption and weak PRFs. They showed the following results: RKA-PRFs can be used to construct all the other RKA primitives; Φ-RKA secure IBE yields Φ-RKA secure PKE with the BCHK transformation [10]; Φ-RKA signature can be obtained with the IBE-to-Signature transformation of Naor (mentioned in [12]). Though RKA-PRFs play an important role in RKA-secure primitives, it was not until 2014 that the problem to construct non-linear RKA-PRFs was solved by Abdalla, Benhamouda, Passelègue and Paterson [2]. Concretely, they presented a RKA-secure PRFs under the stronger decisional $d$-Diffie–Hellman assumption for a class of polynomial RKD functions of bounded degree $d$. In 2012, Bellare, Paterson and Thomson [7] proposed a framework

for constructing RKA-secure IBE, including the non-linear (polynomials of bounded degree $d$) RKA secure IBE based on the $d$-EDBDH assumption. In the same year, Wee [41] presented a framework for constructing linear-RKA secure PKE schemes from standard assumptions, including factoring and DBDH. In addition, the author introduced the notion of weaker RKA security, in which the adversary is allowed to make decryption queries that are not equal to the challenge ciphertext.

A general approach to RKA security is making use of continuous non-malleable codes [30] or continuous non-malleable key derivation [38]. In these two primitives, the real cryptographic key is encoded into a codeword or is derived from a seed. If the codeword or the seed is modified to another one, the new recovered cryptographic key is either equal to the original key or independent of the original key. Very recently, Chen et al. showed that continuous non-malleable key derivation can be obtained from non-malleable functions [15].

## 2. Preliminaries

### 2.1. Notation

Throughout this paper, we denote by $\kappa$ the security parameter and by $\mathsf{negl}(\kappa)$ a negligible function in $\kappa$, that is, for any positive integer $c$, there exists $N$ such that for all $\kappa > N$, $\mathsf{negl}(\kappa) < 1/\kappa^c$. For a finite set $S$, we use $s \xleftarrow{\$} S$ to represent the operation of sampling $s$ uniformly at random from $S$. We write with $\mathrm{poly}(\kappa)$ an unspecified polynomial value of $\kappa$. "PPT" and "DPT" stand for probabilistic polynomial-time and deterministic polynomial-time respectively. For a positive integer $n$, we denote by $[n]$ the set $\{1, \ldots, n\}$ and by $[n \backslash i]$ the set $\{1, \ldots, i-1, i+1, \ldots, n\}$.

### 2.2. Key-encapsulation mechanism

Bellare, Paterson and Thomson [7, Theorem 7.1] showed that the KEM/DEM paradigm of [17] extends to the RKA setting in a natural way. Therefore this paper will focus on key-encapsulation mechanism (KEM) schemes rather than public-key encryption schemes.

A key-encapsulation mechanism scheme consists of four (probabilistic) polynomial-time algorithms: $(\mathsf{KEM.Sys}, \mathsf{KEM.Gen}, \mathsf{KEM.Encap}, \mathsf{KEM.Decap})$ that satisfy the following properties: (1) $\mathsf{KEM.Sys}(1^\kappa)$ is a PPT parameter generation algorithm. It takes as input a security parameter $1^\kappa$ and outputs a system parameter $sp$ (which implicitly defines an encapsulated key space $\mathcal{K}$ and a ciphertext space $\mathcal{C}$); (2) $\mathsf{KEM.Gen}(sp)$ is a PPT public key and secret key generation algorithm. It takes as input the system parameter $sp$ and outputs a public key $pk$ and a secret key $sk$; (3) $\mathsf{KEM.Encap}(pk)$ is a PPT encapsulation algorithm. It takes as input the public key $pk$ and outputs a ciphertext $C$ as well as a random symmetric key (session key) $K \in \mathcal{K}$; (4) $\mathsf{KEM.Decap}(sk, C)$ is a DPT decapsulation algorithm. It takes as input a ciphertext $C$ and a secret key $sk$, and outputs a symmetric key $K \in \mathcal{K}$ or the special reject symbol $\perp$, indicating that $C$ is an invalid ciphertext. The consistence requires that for all $\kappa \in \mathbb{N}$, all possible system parameter $sp \leftarrow \mathsf{KEM.Sys}(1^\kappa)$ and public/secret key pair $(pk, sk) \leftarrow \mathsf{KEM.Gen}(sp)$, and all $(C, K) \leftarrow \mathsf{Encap}(pk)$, it always has $\mathsf{KEM.Decap}(sk, C) = K$.

**RKD functions.** Suppose that $S$ is a finite set. Let $\mathcal{F} = \{f : S \to S\}$ denote the set of all efficiently computable functions with the same domain and range $S$. Moreover, their relationships should be efficiently checkable. We say a class of functions $\Phi$ is a related-key derivation (RKD) function class, if it is a subset of $\mathcal{F}$. In this paper, $S$ is just the secret key space and the RKD functions may depend on the system parameters. Hereafter, we write with $\Phi^{\mathrm{lin}}$, $\Phi^{\mathrm{aff}}$ and $\Phi^{\mathrm{poly}}$ the concrete classes of linear, affine and polynomial RKD functions respectively.

Following [7], we define the indistinguishability against adaptive chosen-ciphertext and related-key attacks for RKD function class $\Phi$ (abbreviated as $\Phi$-RKA security) for a KEM scheme.

**Definition 1** ($\Phi$-RKA security)**.** We say that a key-encapsulation mechanism $\mathsf{KEM}=(\mathsf{KEM.Sys}, \mathsf{KEM.Gen}, \mathsf{KEM.Encap}, \mathsf{KEM.Decap})$ is $\Phi$-RKA secure, if for any stateful PPT adversary $\mathcal{A}$, the following advantage

$$\mathsf{Adv}^{\Phi-\mathrm{rka}}_{\mathsf{KEM},\mathcal{A}}(\kappa) := \left| \Pr\left[ b' = b : \begin{array}{l} sp^* \leftarrow \mathsf{KEM.Sys}(1^\kappa) \\ (pk^*, sk^*) \leftarrow \mathsf{KEM.Gen}(sp^*) \\ (C^*, K_0^*) \leftarrow \mathsf{KEM.Encap}(pk^*) \\ K_1^* \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\} \\ b' \leftarrow \mathcal{A}^{\mathcal{O}^{\Phi}_{sk^*}(\cdot,\cdot)}(pk^*, (C^*, K_b^*)) \end{array} \right] - \frac{1}{2} \right|$$

is negligible in $\kappa$, where the RKA oracle $\mathcal{O}^{\Phi}_{sk^*}(\cdot,\cdot)$, on input $(\phi, C) \in \Phi \times \mathcal{C}$, returns $\mathsf{KEM.Decap}(\phi(sk^*), C)$. Naturally, the adversary is forbidden to make queries such that $(\phi(sk^*), C) = (sk^*, C^*)$ once the adversary saw the challenge ciphertext $C^*$.

### 2.3. Target collision-resistant hash function

Let $\mathcal{H} = \{\mathsf{TCR} : \mathcal{R} \to \mathcal{D}\}$ be a family of hash functions from domain $\mathcal{D}$ to range $\mathcal{R}$. For simplicity, we denote by $\mathsf{TCR} \leftarrow \mathcal{H}$ the function sampling algorithm. Actually, this would be a PPT algorithm that takes as input the security parameter and outputs a function index. The target collision-resistant hash functions says that given a hash function $\mathsf{TCR}$ and a random $x \in \mathcal{D}$, it is hard to find $x' \neq x$ such that $\mathsf{TCR}(x') = \mathsf{TCR}(x)$.

**Definition 2** (TCR Hash Function)**.** We say that a hash function family $\mathcal{H} = \{\mathsf{TCR} : \mathcal{R} \to \mathcal{D}\}$ is target collision resistant, if for any PPT algorithm $\mathcal{A}$, the following advantage

$$\mathsf{Adv}^{\mathrm{tcr}}_{\mathcal{H},\mathcal{A}}(\kappa) := \Pr\left[x' \neq x \wedge \mathsf{TCR}(x') = \mathsf{TCR}(x) : \begin{array}{c}\mathsf{TCR} \leftarrow \mathcal{H}, x \xleftarrow{\$} \mathcal{D} \\ x' \leftarrow \mathcal{A}(\mathsf{TCR}, x)\end{array}\right]$$

is negligible in $\kappa$.

### *2.4. Extended bilinear Diffie–Hellman assumption*

A pairing instance generation algorithm $\mathsf{PGen}(1^\kappa)$ is a PPT algorithm that on input $1^\kappa$, outputs a description of bilinear groups $(\hat{e}, \mathbb{G}, \mathbb{G}_T, p)$. This paper considers bilinear groups with prime order $p$ and symmetric bilinear map, i.e., there exists an efficiently compubable map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that $\hat{e}(g^a, g^b) = \hat{e}(g^b, g^a) = \hat{e}(g, g)^{ab}$ for any $a, b \in \mathbb{Z}_p$ and $g \in \mathbb{G}$. We simply denote by $g \xleftarrow{\$} \mathbb{G} \setminus \{1\}$ a random generator sampling algorithm, where 1 stands for the identity element.

For any positive integer $d = \mathrm{poly}(\kappa)$, the $d$-Extended Bilinear Diffie–Hellman ($d$-EBDH) problem over bilinear group $\mathbb{G}$ states that

$$\text{Given}((\hat{e}, \mathbb{G}, \mathbb{G}_T, p), g, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^d}, g^\beta, g^\gamma), \text{ to compute } T = \hat{e}(g, g)^{\alpha\beta\gamma},$$

where $(\hat{e}, \mathbb{G}, \mathbb{G}_T, p) \leftarrow \mathsf{PGen}(1^\kappa)$, $g \xleftarrow{\$} \mathbb{G} \setminus \{1\}$ and $\alpha, \beta, \gamma \xleftarrow{\$} \mathbb{Z}_p$.

Clearly, if $d = 1$, this is just the standard computational Bilinear Diffie–Hellman (BDH) problem.

**Definition 3** ($d$-EBDH Assumption)**.** We say that the $d$-EBDH assumption holds over $\mathbb{G}$, if for any PPT algorithm $\mathcal{A}$, it solves the $d$-EBDH problem with a negligible probability in $\kappa$, where the probability is defined as follows

$$\mathsf{Adv}^{d-\mathrm{ebdh}}_{\mathcal{A},\mathbb{G}}(\kappa) := \Pr\left[T = \hat{e}(g, g)^{\alpha\beta\gamma} : \begin{array}{c}(\hat{e}, \mathbb{G}, \mathbb{G}_T, p) \leftarrow \mathsf{PGen}(1^\kappa) \\ g \xleftarrow{\$} \mathbb{G} \setminus \{1\}, \alpha, \beta, \gamma \xleftarrow{\$} \mathbb{Z}_p \\ T \leftarrow \mathcal{A}(g, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^d}, g^\beta, g^\gamma)\end{array}\right].$$

In the following, we denote by $f_{gl} : \mathbb{G}_T \times \{0,1\}^u \to \{0,1\}^v$ the Goldreich–Levin hard-core function for $d$-EBDH problem with randomness space $\{0,1\}^u$ and range $\{0,1\}^v$, where $u$ and $v$ are suitable positive integers. The Goldreich–Levin theorem [23] gives the following lemma.

**Lemma 1.** *For $(\hat{e}, \mathbb{G}, \mathbb{G}_T, p) \leftarrow \mathsf{PGen}(1^\kappa)$. Define two distributions*

$$\Delta_{real} = (g, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^d}, g^\beta, g^\gamma, K, R)$$
$$\Delta_{rand} = (g, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^d}, g^\beta, g^\gamma, U_v, R)$$

*where $g$ is a random generator of group $\mathbb{G}$, $\alpha, \beta, \gamma \xleftarrow{\$} \mathbb{Z}_p$, $R \xleftarrow{\$} \{0,1\}^u$, $K = f_{gl}(\hat{e}(g, g)^{\alpha\beta\gamma}, R)$ and $U_v \xleftarrow{\$} \{0,1\}^v$. For any PPT algorithm $\mathcal{A}$, the following advantage*

$$\mathsf{Adv}^{d-\mathrm{ebdh}}_{GL,\mathcal{B}}(\kappa) := |\Pr[\mathcal{A}(\Delta_{real}) = 1] - \Pr[\mathcal{A}(\Delta_{rand}) = 1]|$$

*is negligible in $\kappa$ under the d-EBDH assumption. In other words, if the d-EBDH assumption holds, then no PPT algorithm can distinguish the aforementioned two distributions $\Delta_{real}$ and $\Delta_{rand}$.*

## 3. RKA-secure KEM from the EBDH assumption

### *3.1. The construction*

- $\mathsf{KEM.Sys}(1^\kappa)$: Run $(\hat{e}, \mathbb{G}, \mathbb{G}_T, p) \leftarrow \mathsf{PGen}(1^\kappa)$. Choose $f_{gl} : \mathbb{G}_T \times \{0,1\}^u \to \{0,1\}^v$ and randomness $R \xleftarrow{\$} \{0,1\}^u$ for $f_{gl}$. Choose a function TCR from a target collision-resistant hash function family $\mathcal{H} = \{\mathsf{TCR} : \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_p\}$. Choose a random generator $g \xleftarrow{\$} \mathbb{G} \setminus \{1\}$ and random elements $X, X', Z_1, \ldots, Z_n \xleftarrow{\$} \mathbb{G}$. Return
$$sp = \left((\hat{e}, \mathbb{G}, \mathbb{G}_T, p), f_{gl}, R, \mathsf{TCR}, g, X, X', Z_1, \ldots, Z_n\right).$$
- $\mathsf{KEM.Gen}(sp)$: Given a system parameter $sp$, choose a random exponent $\alpha \xleftarrow{\$} \mathbb{Z}_p$, and compute $Y = g^\alpha$. Set
$$pk = Y \quad \text{and} \quad sk = \alpha$$
and return $(pk, sk)$.
- $\mathsf{KEM.Encap}(pk)$: On input a public key $pk = Y$, pick $r \xleftarrow{\$} \mathbb{Z}_p$ and then compute
$$C_0 = g^r \quad C_1 = (X^t X')^r \quad K = \left(f_{gl}(\hat{e}(Y^r, Z_i), R)\right)_{i \in [n]}$$
where $t = \mathsf{TCR}(C_0, Y)$. Return $((C_0, C_1), K)$.

- $\mathsf{KEM.Decap}(sk, (C_0, C_1))$: On input the secret key $sk = \alpha$ and a ciphertext $(C_0, C_1)$, compute $Y = g^{\alpha}$ and $t = \mathsf{TCR}(C_0, Y)$. If $\hat{e}(C_0, X^t X') \neq \hat{e}(g, C_1)$, then return $\perp$. Otherwise, compute, for each $i \in [n]$,

$$K_i = f_{gl}(\hat{e}(C_0^{\alpha}, Z_i), R)$$

and return $K = (K_1, \ldots, K_n) \in \{0,1\}^{nv}$.

The correctness of the above scheme can be checked directly, we omit it here. Its security is established by the following theorem.

**Our RKD functions:** The related-key derivation function class $\Phi^{\mathrm{poly}(d)}$ used in the following theorem consists of all polynomials with bounded degree $d$ over the finite field $\mathbb{F}_p$, where $p$ is the prime order of group $\mathbb{G}$. Without loss of generality, for any $\phi_{a_0,a_1,\ldots,a_d} \in \Phi^{\mathrm{poly}(d)}$, the evaluation of polynomial $\sum_{i=0}^{d} a_i x^i$ is over the finite field $\mathbb{F}_p$.

**Theorem 1.** *Suppose that $f_{gl}$ is a Goldreich–Levin hard-core function, $\mathcal{H}$ is a target collision-resistant hash function family and the d-Extended Bilinear Diffie-Hellman assumption holds in $\mathbb{G}$. Then, the above scheme is a $\Phi^{\mathrm{poly}(d)}$-RKA secure KEM scheme. Concretely, for any PPT $\Phi^{\mathrm{poly}(d)}$-RKA adversary $\mathcal{A}$ that makes at most $q = \mathrm{poly}(\kappa)$ RKA queries, there exist adversaries $\mathcal{B}'$ and $\mathcal{B}''$ of roughly the same complexity as $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\Phi^{\mathrm{poly}(d)}-\mathrm{rka}}_{\mathsf{KEM},\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\mathrm{tcr}}_{\mathcal{H},\mathcal{B}'}(\kappa) + n \cdot \mathsf{Adv}^{d-\mathrm{ebdh}}_{GL,\mathcal{B}''}(\kappa) + \frac{q}{p}$$

*where $p$ is the order of the underlying bilinear group $\mathbb{G}$.*

### 3.2. Security proof

Before to prove Theorem 1, we briefly discuss the RKA security of the original HJKS-KEM scheme. Though the HJKS-KEM scheme is CCA secure under the BDH assumption, it seems to be hard to prove its RKA security (even under the extended BDH assumption). Consider a (simple) linear RKD function class, the adversary can modify the challenge decryption key $sk = \alpha$ to some related decryption key, such as $sk' = \alpha + 1$. Given a challenge HJKS-KEM ciphertext $(C_0^*, C_1^*)$, we know that $(C_0^*, C_1^*)$ is still a valid ciphertext under the modified key $sk'$ (according to the validity checking of the HJKS-KEM scheme). Hence, submitting $(\phi = x + 1, C := (C_0^*, C_1^*))$ to the RKA decryption oracle, the adversary should obtain a session key $K' = (K_1', K_2', \ldots, K_n')$, where $K_i' = f_{gl}(\hat{e}(C_0^{*\alpha+1}, Z_i), R)$. Note that the real session (challenge) key is $(K_i)_{i\in[n]} = (f_{gl}(\hat{e}(C_0^{*\alpha}, Z_i), R))_{i\in[n]}$. It is not clear whether the adversary can derive some useful information about $K_i$ from the knowledge of $K_i'$. To prevent this related-key attack, $K'$ should be unrelated to the challenge key $K$. However, the underlying hard-core function does not suggest such a property. We do not know how to prove the RKA security of the original HJKS-KEM scheme. Nevertheless, our variant HJKS-KEM scheme proposed in this paper can be formally proven to be RKA secure due to the following analysis.

We proceed through a sequence of games played between a fixed PPT adversary $\mathcal{A}$ and a challenger. Let $S_i$ denote the event that $\mathcal{A}$ succeeds (i.e., $b' = b$ in Definition 1) in $\mathsf{Game}_i$.

$\mathsf{Game}_0$: This is the original RKA game. By definition we have

$$\mathsf{Adv}^{\Phi^{\mathrm{poly}(d)}-\mathrm{rka}}_{\mathsf{KEM},\mathcal{A}}(\kappa) := \left|\Pr[S_0] - \frac{1}{2}\right|.$$

In the following games, we denote by $Y^*$, $(C_0^*, C_1^*)$ and $K_b^*$ the challenge public key, challenge ciphertext and challenge encapsulated key respectively, and denote by $(\phi_{a_0,a_1,\ldots,a_d}, (C_0, C_1))$ the RKA queries issued by $\mathcal{A}$. We write with $Y_\phi$ the modified public key corresponding to the modified secret key $\phi_{a_0,a_1,\ldots,a_d}(\alpha)$, i.e., $Y_\phi = g^{\phi_{a_0,a_1,\ldots,a_d}(\alpha)}$.

$\mathsf{Game}_1$: This game is identical to $\mathsf{Game}_0$, except that the challenger rejects all RKA queries of the form $(\phi_{a_0,a_1,\ldots,a_d}, (C_0, C_1))$ such that $\sum_{i=0}^{d} a_i \alpha^i = \alpha$ and $(C_0, C_1) = (C_0^*, C_1^*)$. Recall that in the original RKA game, the challenger only rejects such queries after the adversary seeing the challenge ciphertext $(C_0^*, C_1^*)$. Since $C_0^*$ is chosen uniformly at random from $\mathbb{G}$, the probability that the adversary submits an RKA query such that $C_0 = C_0^*$ before seeing the challenge ciphertext is bounded by $q/p$ where $q$ is the number of RKA queries issued by $\mathcal{A}$. Since $q = \mathrm{poly}(\kappa)$, we have $q/p$ is negligible in $\kappa$. Then

$$|\Pr[S_1] - \Pr[S_0]| \leq \frac{q}{p}.$$

$\mathsf{Game}_2$: Instead of testing $\phi_{a_0,a_1,\ldots,a_d}(\alpha) = \alpha \pmod p$, the challenger checks whether $Y_\phi = g^{\sum_{i=0}^{d} a_i \alpha^i} = Y^*$. That is, the challenger returns $\perp$, if $\mathcal{A}$ submits an RKA query of the form $(\phi_{a_0,a_1,\ldots,a_d}, (C_0, C_1))$ such that $g^{\sum_{i=0}^{d} a_i \alpha^i} = Y^*$ and $(C_0, C_1) = (C_0^*, C_1^*)$. Note that $g$ is a generator of $\mathbb{G}$ with prime order $p$, and thus $\phi_{a_0,a_1,\ldots,a_d}(\alpha) = \alpha$ holds if and only if $g^{\sum_{i=0}^{d} a_i \alpha^i} = Y^*$. This modification is just conceptional and hence

$$\Pr[S_2] = \Pr[S_1].$$

(The purpose of this game is to check the equivalency between the original secret key $\alpha$ and the modified secret key $\phi(\alpha)$ using the public parameters $(g, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^d})$ rather than $\alpha$.)

$\mathsf{Game}_3$: This game is identical to $\mathsf{Game}_2$, except the following modification to the computation of $Y$. In the public/secret keys generation algorithm, besides the public key $Y^* = g^\alpha$, the challenger also computes $Y_i^* = g^{\alpha^i}$ for $2 \le i \le n$ and keeps them in hand. In the following, we implicitly set $Y_0^* = g$ and $Y_1^* = Y^*$.

To answer $\mathcal{A}$'s query $(\phi_{a_0,a_1,\ldots,a_d}, (C_0, C_1))$, the challenger computes $Y_\phi$ via $\Pi_{i=0}^d (Y_i^*)^{a_i}$ instead of $g^{\sum_{i=0}^d a_i \alpha^i}$. Since $\Pi_{i=0}^d (Y_i^*)^{a_i} = \Pi_{i=0}^d (g^{\alpha^i})^{a_i} = g^{\sum_{i=0}^d a_i \alpha^i}$, we have

$$\Pr[S_3] = \Pr[S_2].$$

$\mathsf{Game}_4$: This game is identical to $\mathsf{Game}_3$, except that the challenger returns $\perp$ and halts, if $(C_0, Y_\phi) = (C_0^*, Y^*)$. Note that, in this case, $t = \mathsf{TCR}(C_0, Y_\phi) = \mathsf{TCR}(C_0^*, Y^*) = t^*$. If $C_1 \neq C_1^*$, then

$$\hat{e}(C_0, X^t X') = \hat{e}(C_0^*, X^{t^*} X') = \hat{e}(g, C_1^*) \neq \hat{e}(g, C_1).$$

Any RKA query such that $(C_0, Y_\phi) = (C_0^*, Y^*)$ but $C_1 \neq C_1^*$ will be an inconsistent ciphertext and rejected by the decryption oracle. However if $C_1 = C_1^*$, it will also be rejected by the definition. Therefore

$$\Pr[S_4] = \Pr[S_3].$$

$\mathsf{Game}_5$: This game is identical to $\mathsf{Game}_4$, except that the challenger returns $\perp$ and halts, if $(C_0, Y_\phi) \neq (C_0^*, Y^*)$ but $t = \mathsf{TCR}(C_0, Y_\phi) = \mathsf{TCR}(C_0^*, Y^*) = t^*$. By the target collision-resistance of $\mathsf{TCR}$, we have the following lemma (which we do after the main proof):

**Lemma 2.** *Let $\mathcal{B}'$ be a PPT adversary attacking on the target collision-resistance of the hash function* $\mathsf{TCR}$*. Then,*

$$|\Pr[S_5] - \Pr[S_4]| \le \mathsf{Adv}^{\mathrm{tcr}}_{\mathsf{TCR},\mathcal{B}'}(\kappa).$$

$\mathsf{Game}_6$: This game is identical to $\mathsf{Game}_5$, except that the challenger samples $K_0^* \xleftarrow{\$} \{0,1\}^{n\nu}$ instead of computing $K_0^* = \left(f_{gl}(\hat{e}(C_0^{*\alpha}, Z_i), R)\right)_{i\in[n]}$. Note that, in this game both $K_0^*$ and $K_1^*$ are chosen uniformly at random and thereby we have $\Pr[S_6] = \frac{1}{2}$.

We claim that

$$|\Pr[S_6] - \Pr[S_5]| \le n \cdot \mathsf{Adv}^{d-\mathrm{ebdh}}_{GL,\mathcal{B}''}(\kappa).$$

We prove the above claim through defining a sequence of hybrid games $H_0, \ldots, H_n$ such that $H_0$ equals $\mathsf{Game}_5$. For $i \in [n]$, $H_i$ is the same as $H_{i-1}$, except that the first $i\nu$ bits of $K_0^*$ are chosen randomly and independently. Clearly, $H_n$ equals $\mathsf{Game}_6$. We prove that hybrid $H_i$ is indistinguishable from hybrid $H_{i-1}$ in the successive games.

Let $E_i$ denote the event that $\mathcal{A}$ succeeds in hybrid game $H_i$. We fix any index $i^* \in [n]$. Suppose that there exists a PPT adversary $\mathcal{A}$ that has a non-negligible advantage to distinguish games $H_{i^*}$ and $H_{i^*-1}$. Then, we show that there exists an algorithm $\mathcal{B}_{i^*}$ distinguishing the distributions $\Delta_{real}$ and $\Delta_{rand}$. Given a challenge tuple $\delta = ((\hat{e}, \mathbb{G}, \mathbb{G}_T, p), g, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^d}, g^\beta, g^\gamma, L, R)$, the algorithm $\mathcal{B}_{i^*}$ proceeds as follows:

**Setting up the system parameter.** $\mathcal{B}_{i^*}$ chooses random exponents $\omega_1, \omega_2 \in \mathbb{Z}_p$, and sets $X = (g^\beta)^{\omega_1}$, $X' = (g^\beta)^{-\omega_1 t^*} g^{\omega_2}$, and $Z_{i^*} = g^\beta$, where $t^* = \mathsf{TCR}(g^\gamma, g^\alpha)$. For $i \in [n \backslash i^*]$, $\mathcal{B}_{i^*}$ picks random exponents $z_i \in \mathbb{Z}_p$ and sets $Z_i = g^{z_i}$. Finally, $\mathcal{B}_{i^*}$ returns the following system parameter to $\mathcal{A}$

$$sp = ((\hat{e}, \mathbb{G}, \mathbb{G}_T, p), f_{gl}, R, \mathsf{TCR}, g, X, X', Z_1, \ldots, Z_n).$$

Clearly, the simulated system parameter has the same distribution as in that the original game. Note that $\mathcal{B}_{i^*}$ knows the discrete logarithms of all $Z_i$'s to the base $g$, except $Z_{i^*}$.

**Setting up the public key.** $\mathcal{B}_{i^*}$ sets $pk := Y^* = g^\alpha$ and implicitly sets $Y_i^* = g^{\alpha^i}$ for $0 \le i \le d$.

**Setting up the challenge ciphertext.** $\mathcal{B}_{i^*}$ sets $C_0^* = g^\gamma$ and $C_1^* = (g^\gamma)^{\omega_2}$. So, $\mathsf{TCR}(C_0^*, Y^*) = \mathsf{TCR}(g^\gamma, g^\alpha) = t^*$. According to the set-up of $X$ and $X'$, it follows that

$$(X^{t^*} X')^\gamma = (((g^\beta)^{\omega_1})^{t^*} \cdot (g^\beta)^{-\omega_1 t^*} g^{\omega_2})^\gamma = (g^\gamma)^{\omega_2}.$$

So, $(C_0^*, C_1^*)$ is a consistent ciphertext and has the same distribution as in $\mathsf{Game}_0$.
Then $\mathcal{B}_{i^*}$ randomly chooses the first $i^* - 1$ keys $K_{0,1}^*, \ldots, K_{0,i^*-1}^*$, sets $K_{0,i^*}^* := L$, and computes $K_{0,j}^* = f_{gl}(\hat{e}(C_0^*, Y^*)^{z_j}, R)$ for $i^* + 1 \le j \le n$. After that, $\mathcal{B}_{i^*}$ lets $K_0^* = (K_{0,1}^*, \ldots, K_{0,n}^*)$, and samples $K_1^* = (K_{1,1}^*, \ldots, K_{1,n}^*) \xleftarrow{\$} \{0,1\}^{n\nu}$ as in the original RKA game. Finally, $\mathcal{B}_{i^*}$ picks a random coin $b \xleftarrow{\$} \{0,1\}$ and returns the challenge ciphertext $(C_0^*, C_1^*)$ together with the key $K_b^*$ to $\mathcal{A}$.

**Handling RKA queries.** Suppose that $(\phi_{a_0,a_1,\ldots,a_d}, (C_0, C_1))$ is an RKA query issued by $\mathcal{A}$ and suppose that $C_0 = g^r$ for some unknown exponent $r \in \mathbb{Z}_p$. $\mathcal{B}_{i^*}$ computes $Y_\phi = \Pi_{i=0}^d (Y_i^*)^{a_i}$. If $(C_0, Y_\phi) = (C_0^*, Y^*)$ or $t = \mathsf{TCR}(C_0, Y_\phi) = t^*$, $\mathcal{B}_{i^*}$ returns $\perp$ and halts. Otherwise, $\mathcal{B}_{i^*}$ tests the consistency of the ciphertext by verifying $\hat{e}(C_0, X^t X') = \hat{e}(g, C_1)$. If the equality does not hold, $\mathcal{B}$ returns $\perp$ and halts. Otherwise, $\mathcal{B}$ sets $K = (K_1, \ldots, K_n)$ as

$$K_i = \begin{cases} f_{gl}(\hat{e}(C_0^{z_i}, \Pi_{i=0}^d (Y_i^*)^{a_i}), R) & \text{if } i \in [n \setminus i^*] \\ f_{gl}(\hat{e}(\bar{X}, \Pi_{i=0}^d (Y_i^*)^{a_i}), R) & \text{if } i = i^* \end{cases}$$

where $\bar{X} = (C_1/C_0^{\omega_2})^{1/(\omega_1(t-t^*))}$ (note that $t \neq t^*$).
Next, we discuss the correctness of all $K_i$.
Recall that $Y_i^* = g^{\alpha^i}$ $(0 \leq i \leq n)$ for some unknown $\alpha \in \mathbb{Z}_p$. For $i \in [n \backslash i^*]$, we have

$$\begin{aligned} K_i &= f_{gl}(\hat{e}(C_0^{z_i}, \Pi_{i=0}^{d}(Y_i^*)^{a_i}), R) \\ &= f_{gl}(\hat{e}(C_0^{z_i}, g^{\sum_{i=0}^{d} a_i \alpha^i}), R) \\ &= f_{gl}(\hat{e}(C_0^{\sum_{i=0}^{d} a_i \alpha^i}, Z_i), R). \end{aligned}$$

For $i = i^*$, since

$$\begin{aligned} \bar{X} &= (C_1/C_0^{\omega_2})^{1/(\omega_1(t-t^*))} \\ &= \left(\frac{(X^t X')^r}{g^{r\omega_2}}\right)^{\frac{1}{\omega_1(t-t^*)}} \\ &= \left(\frac{((g^\beta)^{\omega_1 t}(g^\beta)^{-\omega_1 t^*} g^{\omega_2})^r}{g^{r\omega_2}}\right)^{\frac{1}{\omega_1(t-t^*)}} \\ &= (g^\beta)^r, \end{aligned}$$

we have

$$\begin{aligned} K_{i^*} &= f_{gl}(\hat{e}((g^\beta)^r, \Pi_{i=0}^{d}(Y_i^*)^{a_i}), R) \\ &= f_{gl}(\hat{e}((g^\beta)^r, g^{\sum_{i=0}^{d} a_i \alpha^i}), R) \\ &= f_{gl}(\hat{e}(C_0^{\sum_{i=0}^{d} a_i \alpha^i}, Z_{i^*}), R). \end{aligned}$$

Thus, $\mathcal{B}_{i^*}$ correctly answers all RKA queries issued by $\mathcal{A}$.
**Summary.** As shown in the above discussion, the distributions of the system parameter, the public key and the RKA queries are identical to that in $H_{i^*}$ (and $H_{i^*-1}$). For the challenge key $K_0^*$, if $\delta$ is an instance of distribution $\Delta_{real}$, then we have $L = f_{gl}(\hat{e}(g,g)^{\alpha\beta\gamma}, R)$. Thus the distribution of $K_0^*$ is identical to that in $H_{i^*-1}$. If $\delta$ is an instance of distribution $\Delta_{rand}$, then $L$ is a random $\nu$ bits string and the distribution of $K_0^*$ is identical to that in $H_{i^*}$. Therefore, $\mathcal{B}_{i^*}$ can use $\mathcal{A}$ to distinguish $\delta \in \Delta_{real}$ from $\delta \in \Delta_{rand}$. This holds for all $i^* \in [n]$. Particularly,

$$|\Pr[E_{i^*}] - \Pr[E_{i^*-1}]| \leq \mathsf{Adv}^{d-\mathrm{ebdh}}_{GL,\mathcal{B}_{i^*}}(\kappa).$$

We conclude that

$$\begin{aligned} |\Pr[S_6] - \Pr[S_5]| &= |\Pr[E_n] - \Pr[E_0]| \\ &\leq \sum_{i=1}^{n} |\Pr[E_i] - \Pr[E_{i-1}]| \\ &\leq n \cdot \mathsf{Adv}^{d-\mathrm{ebdh}}_{GL,\mathcal{B}''}(\kappa). \end{aligned}$$

for some PPT adversary $\mathcal{B}''$ attacking on the Goldreich–Levin hard-core function.

Taking all things together, this completes the proof of Theorem 1.

**Proof of Lemma 2.** To prove this lemma, we introduce two hybrid games.

- $\mathsf{Game}_{4.1}$: This game is the same as $\mathsf{Game}_4$, except the following changes: (i) We keep the discrete logarithms of $X$ and $X'$ to the base $g$, i.e., we first sample random $x, x' \in \mathbb{Z}_p$, and then set $X = g^x$ and $X' = g^{x'}$; (ii) We compute the challenge ciphertext $C_1^*$ using $(C_0^*)^{x \cdot t^* + x'}$ instead of $(X^t X')^{r^*}$ where $r^* = \log_g C_0^*$; (iii) We compute the challenge encapsulated key $K_0^*$ using the secret key $sk^* = \alpha$ instead of the witness $r^*$ of $C_0^*$. That is, for each $i$, $K_{0,i}^*$ is computed via $f_{gl}(\hat{e}(C_0^{*\alpha}, Z_i), R)$ rather than $f_{gl}(\hat{e}(Y^{*r^*}, Z_i), R)$. Clearly, all these modifications are just conceptional, and hence

  $$\Pr[S_{4.1}] = \Pr[S_4].$$

- $\mathsf{Game}_{4.2}$: This game is identical to $\mathsf{Game}_{4.2}$, except that the challenger returns $\perp$ and halts, if $(C_0, Y_\phi) \neq (C_0^*, Y^*)$ but $t = \mathsf{TCR}(C_0, Y_\phi) = \mathsf{TCR}(C_0^*, Y^*) = t^*$. We show that

  $$|\Pr[S_{4.2}] - \Pr[S_{4.1}]| \leq \mathsf{Adv}^{\mathrm{tcr}}_{\mathsf{TCR},\mathcal{B}'}(\kappa)$$

  for a PPT adversary $\mathcal{B}'$ attacking on TCR.
  Given a challenge TCR instance $(A^*, B^*) \in \mathbb{G} \times \mathbb{G}$, we construct an efficient algorithm $\mathcal{S}$ to simulate $\mathsf{Game}_{4.1}$. Without lossy of generality, we assume that the group $\mathbb{G}$ comes from some bilinear group $(\hat{e}, \mathbb{G}, \mathbb{G}_T, p)$ generated by $\mathsf{PGen}(1^\kappa)$. The

simulator first chooses random elements $\alpha, x, x' \in \mathbb{Z}_p$, and sets $g = B^{*\frac{1}{\alpha}}$, $sk^* = \alpha$, $X = g^x$ and $X' = g^{x'}$. So, the challenge public key is $pk^* = Y^* = g^\alpha = B^*$. The simulator then generates the other elements of the system parameter as in $\mathsf{Game}_{4.1}$. Next, the simulator sets $C_0^* = A^*$, and computes

$$\begin{cases} C_1^* = (A^*)^{x \cdot t^* + x'}, \text{ where } t^* = \mathsf{TCR}(C_0^*, Y^*) \\ K_{0,i}^* = f_{gl}(\hat{e}(A^{*\alpha}, Z_i), R), \text{ for } i \in [n] \end{cases}.$$

Observe that the simulated challenge ciphertext $(C_0^*, C_1^*)$ and $K_0^*$ have the same distributions as that in $\mathsf{Game}_{4.1}$. For any RKA queries $(\phi, (C_0, C_1))$, the simulator can also answer the adversary's query using $\alpha$ and keep the tuple $(C_0, Y_\phi)$, where $Y_\phi = g^{\phi(\alpha)}$. Finally, the simulator checks whether there exists same tuple $(C_0, Y_\phi)$ satisfying $\mathsf{TCR}(C_0, Y_\phi) = \mathsf{TCR}(C_0^*, Y^*)$ but $(C_0, Y_\phi) \neq (C_0^*, Y^*)$. If so, the simulator outputs such tuple to its own TCR challenger. By the TCR assumption, such event occurs with probability at most $\mathsf{Adv}^{\mathrm{tcr}}_{\mathsf{TCR},\mathcal{B}'}(\kappa)$. Since $\mathsf{Game}_{4.2}$ is identical to $\mathsf{Game}_{4.1}$ unless this event occurs, we have $|\Pr[S_{4.2}] - \Pr[S_{4.1}]| \leq \mathsf{Adv}^{\mathrm{tcr}}_{\mathsf{TCR},\mathcal{B}'}(\kappa)$.

Recall that $\mathsf{Game}_5$ is identical to $\mathsf{Game}_{4.2}$, except that the challenge ciphertext is again computed using the witness $r^*$ of $C_0^*$. So, the difference between these two games is conceptional. This finishes the proof of Lemma 2. □

### 3.3. Reducing pairings in encapsulation

One drawback of the above scheme is that the encapsulation algorithm requires too many pairings. In general, computing the pairing is much slower than an exponentiation. To alleviate this problem, we introduce a simple way to reduce the number of pairings in the encryption algorithm. The idea is to precompute the pairings $\hat{Z}_i = \hat{e}(Y, Z_i)$ for $i \in [n]$, and add these values into the public key. Concretely, the variant scheme $\mathsf{KEM}' = (\mathsf{KEM}'.\mathsf{Sys}, \mathsf{KEM}'.\mathsf{Gen}, \mathsf{KEM}'.\mathsf{Encap}, \mathsf{KEM}'.\mathsf{Decap})$ is the same as our original KEM scheme described in Section 3.1, except the following two differences in $\mathsf{KEM}'.\mathsf{Sys}$ and $\mathsf{KEM}'.\mathsf{Encap}$:

1. In $\mathsf{KEM}'.\mathsf{Sys}$, the system parameter is modified to
$$pk = (Y, \hat{Z}_1, \ldots, \hat{Z}_n),$$
where $Y = g^\alpha$ for some secret key $sk = \alpha$, and $\hat{Z}_i = \hat{e}(Y, Z_i)$.
2. On input a public key $pk = (Y, \hat{Z}_1, \ldots, \hat{Z}_n)$, the encapsulation algorithm $\mathsf{KEM}'.\mathsf{Encap}$ picks a random $r \in \mathbb{Z}_p$ and then computes
$$C_0 = g^r \quad C_1 = (X^t X')^r \quad K = \left(f_{gl}(\hat{Z}_i^r, R)\right)_{i \in [n]}$$
where $t = \mathsf{TCR}(C_0, Y)$. Return $((C_0, C_1), K)$.

We claim that the RKA-security of the variant KEM scheme is the same as that of the original KEM scheme, i.e., we have the following theorem.

**Theorem 2.** *For any PPT $\Phi^{\mathrm{poly}(d)}$-RKA adversary $\mathcal{A}$ that makes at most $q = \mathrm{poly}(\kappa)$ RKA queries, there exist adversaries $\mathcal{B}'$ and $\mathcal{B}''$ of roughly the same complexity as $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\Phi^{\mathrm{poly}(d)}-\mathrm{rka}}_{\mathsf{KEM}',\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\mathrm{tcr}}_{\mathcal{H},\mathcal{B}'}(\kappa) + n \cdot \mathsf{Adv}^{d-\mathrm{ebdh}}_{GL,\mathcal{B}''}(\kappa) + \frac{q}{p}$$

*where $p$ is the order of the underlying bilinear group $\mathbb{G}$.*

The proof of the above theorem is almost identical to that of Theorem 1, except that we need to simulate the elements $\hat{Z}_i$ of the challenge public key $pk$ in $\mathsf{Game}_6$. Observe that the simulator can compute $\hat{Z}_i$ from the values $Y^*$ and $Z_i$ using pairing $\hat{e}(Y^*, Z_i)$, where $Y^*$ comes from the challenge $d$-EBDH instance and $Z_i$ are chosen by the simulator itself. In this way, the challenge public key $pk = (Y^*, \hat{Z}_1, \ldots, \hat{Z}_n)$ can be simulated by the simulator. We omit the formal security proof.

## 4. Comparison

In this section, we compare our result with previous known RKA-secure public key primitives, including public-key encryption (PKE), identity-based encryption (IBE) and key-encapsulation mechanism (KEM), in terms of RKD function classes and assumptions. Note that, the later two primitives imply RKA-secure PKE schemes in a modular and efficient way. According to the framework proposed by Bellare, Cash and Miller [5], RKA-secure pseudo-random functions (PRFs) can be used to construct other primitives for the same set of RKD functions. We omit the comparison with the results obtained from the RKA-PRFs, as the best result from RKA-PRFs does not surpass the best result from RKA-PKE or RKA-KEM.

It has been shown in Table 1 that our result achieves non-linear RKA security from search problem while the others rely on decisional assumptions with the exception of Wee's factoring-based construction and Cui et al.'s work. Note that Cui et al.'s method requires extra operations of computing a commitment and verifying the commitment, while our method only involves an extra operation of computing one exponentiation in the decryption. Our polynomial RKA security is obtained from a slightly less standard search problem, i.e., the $d$-extended BDH problem. This limitation exists in Cui et al.'s

**Table 1**
Summary of existing RKA-secure PKE, IBE and KEM schemes. The KEM schemes also rely on a TCR function. We do not write it explicitly in the figure, as TCR function can be constructed under discrete logarithm assumption, which is weaker than the other assumptions. In the table, "HR", "QR" and "DCR" stand for "Higher Residuosity assumption", "Quadratic Residuosity assumption" and "Decisional Composite Residuosity assumption" respectively.

| Scheme | Primitive | Assumption | RKD functions | Search problem |
|---|---|---|---|---|
| [41] | PKE | DBDH | Linear | ✗ |
| [7] | IBE | DBDH | Affine | ✗ |
| [7] | IBE | $d$-EDBDH | Polynomial | ✗ |
| [7] | KEM | DBDH | Affine | ✗ |
| [32] | PKE | DDH/HR | Affine | ✗ |
| [31] | PKE | DCR/QR | Affine | ✗ |
| [21] | KEM | DBDH | Invertible | ✗ |
| [21] | IBE | DBDH | Invertible | ✗ |
| [41] | PKE | Factoring | Linear | ✓ |
| [20][a] | PKE | $d$-EBDH | Restricted functions | ✓ |
| Sections 3.1 and 3.3 | KEM | BDH | Affine | ✓ |
| Sections 3.1 and 3.3 | KEM | $d$-EBDH | Polynomial | ✓ |

[a] Obtained via applying their framework to the HJKS-KEM scheme under the $d$-EBDH assumption.

**Table 2**
Efficiency comparison.

| Scheme | Param. $\#\mathbb{G}$ | pk $[\#\mathbb{G}, \#\mathbb{G}_T]$ | sk $\#\mathbb{Z}_p$ | CT $\#\mathbb{G}$ | Encap. [#Exp., #Pairing] | Decap. [#Exp., #Pairing] | RKA |
|---|---|---|---|---|---|---|---|
| [26] | $n+1$ | $[1, 0]$ | 1 | 2 | $[n+3, n]$ | $[n+1, n+2]$ | unknow |
| Scheme 3.1 | $n+1$ | $[1, 0]$ | 1 | 2 | $[n+3, n]$ | $[n+2, n+2]$ | Yes |
| Scheme 3.3 | $n+1$ | $[1, n]$ | 1 | 2 | $[n+3, 0]$ | $[n+2, n+2]$ | Yes |

method too. We leave it as a further work to construct RKA-secure primitives from standard search problems, such as BDH and factoring, for polynomial and even invertible RKD functions.

In Table 2, we also compare our schemes with the original CCA-secure HJKS-KEM scheme, in terms of key and ciphertext sizes, encapsulation and decapsulation computations. For the system parameter size, we do not consider the size of the group description, TCR functions and the hard-core function. In encapsulation and decapsulation, we only consider the dominating computations, including exponentiation (Exp.) and pairing (Pairing), omitting the computations of TCR function and some constant number of multiplications. Table 2 shows that, to achieve RKA security, our two schemes only add an extra exponentiation in the decapsulation of the original HJKS-KEM scheme. Moreover, the second scheme even eliminates the $n$ pairings in the encapsulation with the price of a pre-computation of $n$ pairings in the public key.

## 5. Conclusion

In this paper, we presented an efficient key-encapsulation mechanism that is secure against polynomial related-key attacks in the standard model. Different from previous polynomial-RKA secure encryption schemes, the construction is based on a hard search problem, namely, $d$-extended BDH, rather than decisional problems. It is also the first affine-RKA secure encryption scheme under the BDH assumption. However, like in previous RKA-secure primitives, to protect against polynomial RKD functions, we rely on a less standard assumption. It remains an open problem to construct RKA-secure primitives, such as KEM/PKE for larger classes of RKD functions from standard search assumptions.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China [grant numbers 61502400, 61672346, 61373153, 61303230], by the Foundation of Sichuan Educational Committee [grant number 16ZB0140] and by the Natural Science Foundation of Southwest University of Science and Technology [grant number 16zx7107].

## References

[1] M. Abdalla, M. Bellare, G. Neven, Robust encryption, in: D. Micciancio (Ed.), TCC 2010, Lecture Notes in Computer Science, vol. 5978, Springer, 2010, pp. 480–497.
[2] M. Abdalla, F. Benhamouda, A. Passelègue, K.G. Paterson, Related-key security for pseudorandom functions beyond the linear barrier, in: J.A. Garay, R. Gennaro (Eds.), CRYPTO 2014, Part I, LNCS, vol. 8616, Springer, 2014, pp. 77–94.
[3] B. Applebaum, D. Harnik, Y. Ishai, Semantic security under related-key attacks and applications, in: Innovations in Computer Science - ICS 2010, Tsinghua University Press, 2011, pp. 45–60.
[4] M. Bellare, D. Cash, Pseudorandom functions and permutations provably secure against related-key attacks, in: T. Rabin (Ed.), CRYPTO 2010, LNCS, vol. 6223, Springer, 2010, pp. 666–684.
[5] M. Bellare, D. Cash, R. Miller, Cryptography secure against related-key attacks and tampering, in: D.H. Lee, X. Wang (Eds.), ASIACRYPT 2011, LNCS, vol. 7073, Springer, 2011, pp. 486–503.

[6] M. Bellare, T. Kohno, A theoretical treatment of related-key attacks: Rka-prps, rka-prfs, and applications, in: E. Biham (Ed.), EUROCRYPT 2003, LNCS, vol. 2656, Springer, 2003, pp. 491–506.
[7] M. Bellare, K.G. Paterson, S. Thomson, RKA security beyond the linear barrier: ibe, encryption and signatures, in: X. Wang, K. Sako (Eds.), ASIACRYPT 2012, LNCS, vol. 7658, Springer, 2012, pp. 331–348.
[8] E. Biham, New types of cryptoanalytic attacks using related keys (extended abstract), in: T. Helleseth (Ed.), EUROCRYPT 1993, LNCS, vol. 765, Springer, 1993, pp. 398–409.
[9] E. Biham, A. Shamir, Differential fault analysis of secret key cryptosystems, in: J.B.S. K. (Ed.), CRYPTO 1997, LNCS, vol. 1294, Springer, 1997, pp. 513–525.
[10] D. Boneh, R. Canetti, S. Halevi, J. Katz, Chosen-ciphertext security from identity-based encryption, SIAM J. Comput. 36 (5) (2007) 1301–1328.
[11] D. Boneh, R.A. DeMillo, R.J. Lipton, On the importance of checking cryptographic protocols for faults (extended abstract), in: W. Fumy (Ed.), EUROCRYPT 1997, LNCS, vol. 1233, Springer, 1997, pp. 37–51.
[12] D. Boneh, M.K. Franklin, Identity-based encryption from the weil pairing, SIAM J. Comput. 32 (3) (2003) 586–615.
[13] R. Canetti, S. Halevi, J. Katz, J. Camenisch, Chosen-ciphertext security from identity-based encryption, in: C. Cachin (Ed.), EUROCRYPT 2004, LNCS, vol. 3027, Springer, 2004, pp. 207–222.
[14] D. Cash, E. Kiltz, V. Shoup, The twin diffie-hellman problem and applications, J. Cryptol. 22 (4) (2009) 470–504.
[15] Y. Chen, B. Qin, J. Zhang, Y. Deng, S.S.M. Chow, Non-malleable functions and their applications, in: C. Cheng, K. Chung, G. Persiano, B. Yang (Eds.), PKC 2016, Part II, LNCS, vol. 9615, Springer, 2016, pp. 386–416.
[16] R. Cramer, V. Shoup, Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption, in: L.R. Knudsen (Ed.), EUROCRYPT 2002, LNCS, vol. 2332, Springer, 2002, pp. 45–64.
[17] R. Cramer, V. Shoup, Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack, SIAM J. Comput. 33 (1) (2004) 167–226.
[18] H. Cui, Y. Mu, M.H. Au, Z.M. Mao, Public-key encryption resilient to linear related-key attacks, in: T.A. Zia, A.Y. Zomaya, V. Varadharajan (Eds.), SecureComm 2013, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 127, Springer, 2013, pp. 182–196.
[19] H. Cui, Y. Mu, M.H. Au, Public-key encryption resilient against linear related-key attacks revisited, in: TrustCom 2014, IEEE Computer Society, 2014, pp. 268–275.
[20] H. Cui, Y. Mu, M.H. Au, Relations between robustness and RKA security under public-key encryption, Theor. Comput. Sci. 628 (2016) 78–91.
[21] E. Fujisaki, K. Xagawa, Efficient rka-secure KEM and IBE schemes against invertible functions, in: K.E. Lauter, F. Rodríguez-Henríquez (Eds.), LATINCRYPT 2015, Lecture Notes in Computer Science, vol. 9230, Springer, 2015, pp. 3–20.
[22] D. Goldenberg, M. Liskov, On related-secret pseudorandomness, in: D. Micciancio (Ed.), TCC 2010, LNCS, vol. 5978, Springer, 2010, pp. 255–272.
[23] O. Goldreich, L.A. Levin, A hard-core predicate for all one-way functions, in: D.S. Johnson (Ed.), STOC 1989, ACM, 1989, pp. 25–32.
[24] V. Goyal, A. O'Neill, V. Rao, Y. Ishai, Correlated-input secure hash functions, in: TCC 2011, in: LNCS, vol. 6597, Springer, 2011, pp. 182–200.
[25] J.A. Halderman, S.D. Schoen, N. Heninger, W. Clarkson, W. Paul, J.A. Calandrino, A.J. Feldman, J. Appelbaum, E.W. Felten, Lest we remember: cold boot attacks on encryption keys, in: P.C. van Oorschot (Ed.), Proceedings of the 17th USENIX Security Symposium, San Jose, CA, USA. USENIX Association, 2008, 2008, pp. 45–60.
[26] K. Haralambiev, T. Jager, E. Kiltz, V. Shoup, Simple and efficient public-key encryption from computational diffie-hellman in the standard model, in: P.Q. Nguyen, D. Pointcheval (Eds.), PKC 2010, LNCS, vol. 6056, Springer, 2010, pp. 1–18.
[27] D. Hofheinz, E. Kiltz, Secure hybrid encryption from weakened key encapsulation, in: A. Menezes (Ed.), CRYPTO 2007, LNCS, vol. 4622, Springer, 2007, pp. 553–571.
[28] D. Hofheinz, E. Kiltz, Practical chosen ciphertext secure encryption from factoring, in: A. Joux (Ed.), EUROCRYPT 2009, LNCS, vol. 5479, Springer, 2009, pp. 313–332.
[29] D. Hofheinz, E. Kiltz, V. Shoup, Practical chosen ciphertext secure encryption from factoring, J. Cryptol. 26 (1) (2013) 102–118.
[30] Z. Jafargholi, D. Wichs, Tamper detection and continuous non-malleable codes, in: Y. Dodis, J.B. Nielsen (Eds.), TCC 2015, Part I, LNCS, vol. 9014, Springer, 2015, pp. 451–480.
[31] D. Jia, B. Li, X. Lu, Q. Mei, Related key secure PKE from hash proof systems, in: M. Yoshida, K. Mouri (Eds.), IWSEC 2014, Lecture Notes in Computer Science, vol. 8639, Springer, 2014, pp. 250–265.
[32] D. Jia, X. Lu, B. Li, Q. Mei, RKA secure PKE based on the DDH and HR assumptions, in: W. Susilo, R. Reyhanitabar (Eds.), ProvSec 2013, LNCS, vol. 8209, Springer, 2013, pp. 271–287.
[33] L.R. Knudsen, Cryptanalysis of LOKI91, in: J. Seberry, Y. Zheng (Eds.), AUSCRYPT 1992, LNCS, vol. 718, Springer, 1992, pp. 196–208.
[34] P.C. Kocher, Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems, in: N. Koblitz (Ed.), CRYPTO 1996, LNCS, vol. 1109, Springer, 1996, pp. 104–113.
[35] K. Lewi, H.W. Montgomery, A. Raghunathan, Improved constructions of PRFs secure against related-key attacks, in: I. Boureanu, P. Owesarski, S. Vaudenay (Eds.), ACNS 2014, LNCS, vol. 8479, Springer, 2014, pp. 44–61.
[36] S. Lucks, Ciphers secure against related-key attacks, in: B.K. Roy, W. Meier (Eds.), FSE 2004, LNCS, vol. 3017, Springer, 2004, pp. 359–370.
[37] C. Peikert, B. Waters, Lossy trapdoor functions and their applications, in: C. Dwork (Ed.), STOC 2008, ACM, 2008, pp. 187–196.
[38] B. Qin, S. Liu, T.H. Yuen, R.H. Deng, K. Chen, Continuous non-malleable key derivation and its application to related-key security, in: J. Katz (Ed.), PKC 2015, LNCS, volume 9020, Springer, 2015, pp. 557–578.
[39] V. Shoup, Lower bounds for discrete logarithms and related problems, in: W. Fumy (Ed.), EUROCRYPT 1997, LNCS, vol. 1233, Springer, 1997, pp. 256–266.
[40] S. Sun, J.K. Liu, Y. Yu, B. Qin, D. Gu, Rka-secure public key encryptions against efficiently invertible functions, Comput. J. 59 (11) (2016) 1637–1658.
[41] H. Wee, Public key encryption against related key attacks, in: M. Fischlin, J. Buchmann, M. Manulis (Eds.), PKC 2012, LNCS, vol. 7293, Springer, 2012, pp. 262–279.