8-2021

Receiver-anonymity in rerandomizable RCCA-secure cryptosystems resolved

Yi WANG
National University of Defense Technology

Rongmao CHEN
National University of Defense Technology

Guomin YANG
Singapore Management University, gmyang@smu.edu.sg

Xinyi HUANG

Baosheng WANG

See next page for additional authors

Citation

WANG, Yi; CHEN, Rongmao; YANG, Guomin; HUANG, Xinyi; WANG, Baosheng; and YUNG, Moti. Receiver-anonymity in rerandomizable RCCA-secure cryptosystems resolved. (2021). *Advances in Cryptology: CRYPTO 2021: 41st Annual International Cryptology Conference, Virtual, August 16-20: Proceedings*. 12828, 270-300.

Available at: https://ink.library.smu.edu.sg/sis_research/7411

Author

Yi WANG, Rongmao CHEN, Guomin YANG, Xinyi HUANG, Baosheng WANG, and Moti YUNG

Receiver-Anonymity in Rerandomizable RCCA-Secure Cryptosystems Resolved

Yi Wang[1], Rongmao Chen[1](✉), Guomin Yang[2], Xinyi Huang[3](✉), Baosheng Wang[1], and Moti Yung[4,5]

[1] School of Computer, National University of Defense Technology, Changsha, China
{wangyi14,chromao,bswang}@nudt.edu.cn

[2] Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia
gyang@uow.edu.au

[3] Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China
xyhuang@fjnu.edu.cn

[4] Google LLC, New York, NY, USA

[5] Columbia University, New York, USA
moti@cs.columbia.edu

Abstract. In this work we resolve the open problem raised by Prabhakaran and Rosulek at CRYPTO 2007, and present the *first* anonymous, rerandomizable, Replayable-CCA (RCCA) secure public-key encryption scheme. This solution opens the door to numerous privacy-oriented applications with a highly desired RCCA security level. At the core of our construction is a non-trivial extension of smooth projective hash functions (Cramer and Shoup, EUROCRYPT 2002), and a modular generic framework developed for constructing rerandomizable RCCA-secure encryption schemes with receiver-anonymity. The framework gives an enhanced abstraction of the original Prabhakaran and Rosulek's scheme (which was the first construction of rerandomizable RCCA-secure encryption in the standard model), where the most crucial enhancement is the first realization of the desirable property of receiver-anonymity, essential to privacy settings. It also serves as a conceptually more intuitive and generic understanding of RCCA security, which leads, for example, to new implementations of the notion. Finally, note that (since CCA security is not applicable to the privacy applications motivating our work) the concrete results and the conceptual advancement presented here, seem to substantially expand the power and relevance of the notion of rerandomizable RCCA-secure encryption.

Keywords: RCCA security · Receiver-anonymity · Smooth projective hash function

1 Introduction

RCCA security. Security against adaptive chosen-ciphertext attacks (CCA) is widely considered as a *de facto* security standard for public-key encryption

T. Malkin and C. Peikert (Eds.): CRYPTO 2021, LNCS 12828, pp. 270–300, 2021.
https://doi.org/10.1007/978-3-030-84259-8_10

(PKE). However, it is evidenced that for some practical purposes, a somewhat weaker security notion than CCA security is already sufficient [1,16,25]. To this end, Canetti et al. [5] introduced the notion of Replayable-CCA (RCCA) security, which is essentially the same as CCA security, except that no guarantees are given against adversaries with the capability of malleating a ciphertext into a new one of the same plaintext. Such a relaxation endows PKE with desirable features such as rerandomizable RCCA (Rand-RCCA) security which was proposed by Canetti et al. [5] and later formalized by Groth [14]. This notion turns out to have numerous practical applications, such as: cryptographic reverse firewalls [9,12,18], mixnets [13,20] and controlled-malleable NIZK [11].

Constructing Rand-RCCA-secure PKE has been generally considered a difficult problem, and was posed as an open problem in [5]. The difficulty is mainly due to the fact that RCCA security and rerandomizability are seemingly incompatible in some sense. In particular, the construction has to be almost CCA secure while at the same time has special mathematical structure for realizing rerandomizability. A notable construction was by Prabhakaran and Rosulek [22] at CRYPTO 2007 (hereafter referred to as PR scheme) which is the first perfect Rand-RCCA-secure PKE based on the DDH assumption in the standard model.

Receiver-anonymity in the RCCA setting. In [22], Prabhakaran and Rosulek further defined a new notion called *RCCA receiver-anonymity* which is similar to the notion of *key-privacy* introduced by Bellare et al. in [2] but in the RCCA setting. For an RCCA receiver-anonymous encryption scheme, the generated ciphertext should not tell the adversary any information about the underlying public key. Such a property turns out to be essential in privacy-oriented applications where ciphertext-rerandomizability, adaptive security (i.e., permitting strong adversary who may probe the system with ciphertexts), and receiver-anonymity are required simultaneously.

A typical example—given by Prabhakaran and Rosulek [22]—is the application of rerandomizable encryption in mixnets where receiver-anonymity is indispensable. More precisely, consider an anonymous communication (AC) protocol based on universal mixnet [13] where a set of message relays (called mixnodes or mixes) receive a batch of encrypted messages, rerandomize and randomly permute them, and send them on their way forward. Unfortunately, the requirement of ciphertext-rerandomizability, while enabling unlinkability of multiple ciphertexts in terms of their contents, contradicts the desirable strong CCA security. Thus, as it turned out, only rerandomizable CPA-secure encryption schemes are used in previous universal mixnet-based AC protocols [13]. To strengthen the security to the adaptive one (i.e., allowing an adversary of the network to attempt sending ciphertexts of its own to the network as part of its attack), RCCA security is the alternative as it reconciles the required rerandomizability and adaptive security (this active attacker, in fact, is what most earlier works on anonymity are not protected against due to the encryption being CPA-secure only). However, as pointed out by Prabhakaran and Rosulek, without receiver-anonymity, the attacker might still be able to correlate the ciphertexts for the same recipient (i.e., sender-receiver relationships are not broken by the mixing!). This example application demonstrates that anonymous Rand-RCCA-secure PKE is meaningful to strengthening the security of universal mixnet-based AC protocol on the one hand, and to allowing it to achieve anonymity (breaking completely sender-receiver relationship) at the same time. More broadly, for various other privacy-oriented applications [19,23,24,28], RCCA receiver-anonymity is also desirable for privacy protection while withstanding strong adversary with decryption query capability (see the full version [29] for further motivating applications).

The open problem. Unfortunately, the PR scheme [22] does not achieve receiver-anonymity, and therefore, how to construct an anonymous Rand-RCCA-secure PKE to support the above mentioned applications under strong adversary was left as an explicit open problem by Prabhakaran and Rosulek in [22]:

> *"Adding anonymity brings out the power of rerandomizability and yields a potent cryptographic primitive. We note that our scheme does not achieve this definition of anonymity, and leave it as an interesting open problem."*

Somewhat surprisingly, in spite of further developments in constructing Rand-RCCA encryption throughout many years [6,10,11,13,14,17,22], the above open problem remains unsolved to date. The main technical challenge of achieving RCCA receiver-anonymity arises from the fact that different from the typical CCA game, the decryption oracle in the RCCA game would output "`replay`" if the query decryption result equals to either of the challenge plaintexts. Such a relaxation, in fact, gives the adversary more power and consequently raises the difficulty to achieve receiver-anonymity in the RCCA setting. Specifically, the adversary can guess the underlying public key, re-encrypt the challenge ciphertext and verify its guess via querying the decryption oracle. Thus, to defend against this attack, it is required that the rerandomization of ciphertext should not involve the public key. Such a feature was originally referred to as "universal rerandomization" by Golle et al. [13]. However, achieving receiver-anonymity is more challenging than realizing universal rerandomizability, since there may exist other ways allowing the adversary to rerandomize a ciphertext using the public key. In other words, receiver-anonymity is strictly stronger than universal rerandomizability. An example is the PR scheme which is universally rerandomizable but not receiver-anonymous (see Sect. 2 for the detailed analysis).

Motivated by the aforementioned state of affairs and the requirement of receiver-anonymity for privacy-oriented applications, our main goal in this work is to resolve the above challenging problem of achieving RCCA receiver-anonymity. More specifically, we ask *whether it is possible to achieve receiver-anonymity in the RCCA setting; and if the answer is positive, how to attempt a solution which is as generic as possible.* Our second question is motivated by the fact that a generic paradigm would enable a better understanding of the underlying key ideas and more diversified constructions of anonymous Rand-RCCA-secure encryption in a conceptually clear and modular way. Also, a framework using abstract building blocks enables more concrete instantiations from various assumptions, leading to better security (as will be demonstrated by our additional results below).

Our results. We resolve the Prabhakaran and Rosulek's open problem in this work. We design a modular framework for constructing anonymous Rand-RCCA-secure PKE via an extension of the notion of smooth projective hash functions by Cramer and Shoup [8]. Our contributions can be summarized as follows:

- We formalize a novel extension of smooth projective hash function with various types of rerandomizability (Re-SPHF), and redefine the property of smoothness which is crucial to generally realize Rand-RCCA security with receiver-anonymity;
- We design a framework for constructing anonymous Rand-RCCA-secure PKE from Re-SPHFs, and rigorously prove its RCCA security and receiver-anonymity. These turn out to provide a conceptually intuitive understanding of RCCA security and receiver-anonymity;
- We provide the *first* anonymous Rand-RCCA-Secure PKE scheme from k-linear (k-Lin) assumption, which—putting anonymity aside—also improves the PR scheme with its more general hardness assumption.

Remark. It is worth noting that in [22], Prabhakaran and Rosulek also pointed out the potential of generalizing their scheme by following the Cramer-Shoup paradigm [8] (hereafter referred to as CS-paradigm), but they left such an investigation open as well. In fact, as we will illustrate in this work, our proposed framework can, in fact, be viewed as an abstraction of a modified PR scheme. Thus, while mainly motivated by achieving a solution to the RCCA receiver-anonymity, our work also closes Prabhakaran and Rosulek's second open question of generalization via SPHFs.

2 Technical Overview and Related Work

First, let us explain why the PR scheme does not satisfy receiver-anonymity. As a countermeasure, we introduce a concrete approach to achieving RCCA receiver-anonymity based on the PR scheme. To generalize our proposed approach, following the SPHF-based CS-paradigm [8], we then define an extension of SPHF that could well explain the modified PR scheme and its security. To this end, we successfully design a general framework for anonymous, Rand-RCCA-secure PKE, which can, in turn, be instantiated based on different assumptions.

Why the PR scheme is not receiver-anonymous? We start by reviewing the PR scheme and its core idea leading to the RCCA security. The crucial idea toward achieving this goal is using two "strands" of Cramer-Shoup ciphertexts [8] which can be "uniquely" recombined with each other for rerandomization without changing the underlying plaintext.

Overview of the PR scheme. Let $\mathbb{G}$, $\overline{\mathbb{G}}$ be two cyclic groups of prime orders p, q where $p = 2q+1$ where $\overline{\mathbb{G}}$ is also a subgroup of $\mathbb{Z}_p^*$. Let g and $\overline{g}$ be generators of $\mathbb{G}$ and $\overline{\mathbb{G}}$ respectively, $[\mathbf{a}]$ denotes vector $(g^{a_1}, \cdots, g^{a_n})$ for $\mathbf{a} = (a_1, \cdots, a_n) \in \mathbb{Z}_p^n$, and $[\overline{\mathbf{a}}]$ denotes vector $(\overline{g}^{a_1}, \cdots, \overline{g}^{a_n})$ for $\overline{\mathbf{a}} = (a_1, \cdots, a_n) \in \mathbb{Z}_q^n$. The ciphertext of the $\mathtt{PR}$ scheme is

$$\begin{aligned} \zeta &:= \Big(\underbrace{[u(\mathbf{x}+\mathbf{z})]\,, M \cdot \left[\mathbf{b}^\top \mathbf{x}\right], \ \left[\boldsymbol{\alpha}^\top \mathbf{x}\right]}_{C_1\text{: message-carrying strand}}, \ \ \underbrace{[u\mathbf{y}]\,, \ \left[\mathbf{b}^\top \mathbf{y}\right], \ \left[\boldsymbol{\alpha}^\top \mathbf{y}\right]}_{C_2\text{: rerandomization strand}}, \ \ \varrho\Big) \\ \varrho &:= \Big(\underbrace{[\overline{\mathbf{x}}]\,, \ \ u \cdot \left[\overline{\mathbf{b}}^\top \overline{\mathbf{x}}\right], \ \ \left[\overline{\mathbf{c}}^\top \overline{\mathbf{x}}\right]}_{C_3\text{: mask-carrying strand}}, \ \ \underbrace{[\overline{\mathbf{y}}]\,, \ \ \left[\overline{\mathbf{b}}^\top \overline{\mathbf{y}}\right], \ \ \left[\overline{\mathbf{c}}^\top \overline{\mathbf{y}}\right]}_{C_4\text{: rerandomization strand}}\Big) \end{aligned} \tag{1}$$

where $u \in \overline{\mathbb{G}}$, given fixed $\mathbf{g} \in \mathbb{Z}_p^4$ and $\overline{\mathbf{g}} \in \mathbb{Z}_q^2$, $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{Z}_p^4$ with $\mathbf{x} = x\mathbf{g}$, $\mathbf{y} = y\mathbf{g}$ for $x, y \in \mathbb{Z}_p$ and $\mathbf{z} \neq z\mathbf{g}$ for any $z \in \mathbb{Z}_p$, $\overline{\mathbf{x}}, \overline{\mathbf{y}}, \overline{\mathbf{b}}, \overline{\mathbf{c}} \in \mathbb{Z}_q^2$ with $\overline{\mathbf{x}} = \overline{x\mathbf{g}}$, $\overline{\mathbf{y}} = \overline{y\mathbf{g}}$ for $\overline{x}, \overline{y} \in \mathbb{Z}_q$, $\boldsymbol{\alpha} = \mathbf{c} + \tau \mathbf{d}$, $\tau = \Psi(M)$ and $\Psi : \mathbb{G} \to \mathbb{Z}_p$ is a collision-resistant hash function. ϱ is the ciphertext of random mask u under a malleable (and also rerandomizable) encryption scheme (see Sect. 3.1). At the high level, the strand C_1 carries the message while the strand C_2 is to help rerandomize C_1 without public key. The encrypted mask u shared between C_1 and C_2 disables the adversary to mix together strands from two different ciphertexts (of the same plaintext) to obtain a valid ciphertext. The exponents of strand C_1 are perturbed by an additional vector $\mathbf{z}$ to restrict the manner of recombining the two strands. Consequently, to rerandomize ciphertext ζ, one randomly picks $\upsilon \in \overline{\mathbb{G}}$, $s, t \in \mathbb{Z}_p^*$, $\overline{s}, \overline{t} \in \mathbb{Z}_q^*$ and computes

$$\begin{aligned} &C_1' := \left([\upsilon \cdot u(\mathbf{x}+\mathbf{z}) + s\upsilon \cdot u\mathbf{y}]\,, \ M \cdot \left[\mathbf{b}^\top \mathbf{x}\right] \cdot \left[s\mathbf{b}^\top \mathbf{y}\right], \ \left[\boldsymbol{\alpha}^\top \mathbf{x}\right] \cdot \left[s\boldsymbol{\alpha}^\top \mathbf{y}\right]\right), \\ &C_3' := \left([\overline{\mathbf{x}} + \overline{s} \cdot \overline{\mathbf{y}}]\,, \ \upsilon \cdot u \cdot \left[\overline{\mathbf{b}}^\top \overline{\mathbf{x}}\right] \cdot \left[\overline{s}\overline{\mathbf{b}}^\top \overline{\mathbf{y}}\right], \ \left[\overline{\mathbf{c}}^\top \overline{\mathbf{x}}\right] \cdot \left[\overline{s}\overline{\mathbf{c}}^\top \overline{\mathbf{y}}\right]\right), \\ &C_2' := \left([t\upsilon \cdot u\mathbf{y}]\,, \ \left[t\mathbf{b}^\top \mathbf{y}\right], \ \left[t\boldsymbol{\alpha}^\top \mathbf{y}\right]\right) \text{ and } C_4' := \left(\left[\overline{t} \cdot \overline{\mathbf{y}}\right], \ \left[\overline{t}\overline{\mathbf{b}}^\top \overline{\mathbf{y}}\right], \ \left[\overline{t}\overline{\mathbf{c}}^\top \overline{\mathbf{y}}\right]\right). \end{aligned}$$

Partial rerandomizability breaking the receiver-anonymity. It is shown in [22] that the above is the only valid way for full rerandomization of ciphertext. However, one can note that strands C_3 and C_4 can also be rerandomized with public keys $\left[\overline{\mathbf{b}}^\top \overline{\mathbf{g}}\right]$ and $\left[\overline{\mathbf{c}}^\top \overline{\mathbf{g}}\right]$ as follows.

$$\begin{aligned} C_3' &:= \left([\overline{\mathbf{x}} + \overline{s} \cdot \overline{\mathbf{g}}]\,, \ u \cdot \left[\overline{\mathbf{b}}^\top \overline{\mathbf{x}}\right] \cdot \left[\overline{s}\overline{\mathbf{b}}^\top \overline{\mathbf{g}}\right], \ \left[\overline{\mathbf{c}}^\top \overline{\mathbf{x}}\right] \cdot \left[\overline{s}\overline{\mathbf{c}}^\top \overline{\mathbf{g}}\right]\right), \\ C_4' &:= \left([\overline{\mathbf{y}} + \overline{t} \cdot \overline{\mathbf{g}}]\,, \ \left[\overline{\mathbf{b}}^\top \overline{\mathbf{y}}\right] \cdot \left[\overline{t}\overline{\mathbf{b}}^\top \overline{\mathbf{g}}\right], \ \left[\overline{\mathbf{c}}^\top \overline{\mathbf{y}}\right] \cdot \left[\overline{t}\overline{\mathbf{c}}^\top \overline{\mathbf{g}}\right]\right), \end{aligned}$$

where $\overline{s}, \overline{t} \in \mathbb{Z}_q^*$. We now demonstrate why the $\mathtt{PR}$ scheme is not RCCA receiver-anonymous. Recalling the game of RCCA receiver-anonymity in Fig. 2, the adversary has access to a guarded decryption oracle which on input ζ, first computes $M_0 = \mathsf{Dec}(\mathsf{SK}_0, \zeta)$ and $M_1 = \mathsf{Dec}(\mathsf{SK}_1, \zeta)$, then checks if $M \in \{M_0, M_1\}$. If so, it returns `replay`, otherwise it returns (M_0, M_1). As for the $\mathtt{PR}$ scheme, adversary could obtain a ciphertext ζ_0^* by rerandomizing strands C_3 and C_4 in the challenge ciphertext ζ^* with public key PK_0 in the above way. If $b = 0$, ζ_0^* is a valid ciphertext of M; otherwise, ζ_0^* is invalid. With the response of the guarded decryption oracle, the adversary is able to distinguish these two cases.

Our concrete treatment of the PR scheme for RCCA receiver-anonymity. To achieve RCCA receiver-anonymity, we have to disable the rerandomization of strands C_3 and C_4 employing the public key. Note that the rerandomization of strands C_1 and C_2 is restricted by mask u and vector $\mathbf{z}$. If we also apply this technique to C_3 and C_4, extra strands are required to encrypt the mask in C_3 and C_4, which would incur the partial rerandomization of ciphertext employing the public key again. To bypass this problem, we move the masks and additional vectors to the validity checking components of strands. Since the validity checking part contains only one component, an additional component is appended to each strand for perturbation on the validity checking part. Concretely, the ciphertext of our variant is:

$$\zeta := \Big(\underbrace{[\mathbf{x}], M\cdot\left[\mathbf{b}^\top\mathbf{x}\right], \boxed{\left[u\boldsymbol{\alpha}^\top\mathbf{x}^\dagger\right], \left[u\boldsymbol{\beta}^\top\mathbf{x}^\ddagger\right]}}_{C_1:\ \text{message-carrying strand}}, \underbrace{[\mathbf{y}], \left[\mathbf{b}^\top\mathbf{y}\right], \boxed{\left[u\boldsymbol{\alpha}^\top\mathbf{y}\right],\ \left[u\boldsymbol{\beta}^\top\mathbf{y}\right]}}_{C_2:\ \text{rerandomization strand}}, \varrho\Big),$$

$$\varrho := \Big(\underbrace{[\overline{\mathbf{x}}],\ u\cdot\left[\overline{\mathbf{b}}^\top\overline{\mathbf{x}}\right], \boxed{\left[u\overline{\mathbf{c}}^\top\overline{\mathbf{x}}^\dagger\right], \left[u\overline{\mathbf{d}}^\top\overline{\mathbf{x}}^\ddagger\right]}}_{C_3:\ \text{mask-carrying strand}}, \underbrace{[\overline{\mathbf{y}}],\ \left[\overline{\mathbf{b}}^\top\overline{\mathbf{y}}\right], \boxed{\left[u\overline{\mathbf{c}}^\top\overline{\mathbf{y}}\right], \left[u\overline{\mathbf{d}}^\top\overline{\mathbf{y}}\right]}}_{C_4:\ \text{rerandomization strand}}\Big) \tag{2}$$

where $u \in \overline{\mathbb{G}}$, $\mathbf{x}^\dagger = \mathbf{x} + z_1\mathbf{g}$, $\mathbf{x}^\ddagger = \mathbf{x} + z_2\mathbf{g}$ for $z_1, z_2 \in \mathbb{Z}_p^*$ with $z_1 \neq z_2$, $\overline{\mathbf{x}}^\dagger = \overline{\mathbf{x}} + \overline{z}_1\overline{\mathbf{g}}$, $\overline{\mathbf{x}}^\ddagger = \overline{\mathbf{x}} + \overline{z}_2\overline{\mathbf{g}}$ for $\overline{z}_1, \overline{z}_2 \in \mathbb{Z}_q^*$ with $\overline{z}_1 \neq \overline{z}_2$, $\mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f} \in \mathbb{Z}_p^2$, $\boldsymbol{\alpha} = \mathbf{c} + m\mathbf{d}$, $\boldsymbol{\beta} = \mathbf{e} + m\mathbf{f}$, $m = \Psi(M)$ and $\Psi : \mathbb{G} \to \mathbb{Z}_p$ is a collision-resistant hash function. The rerandomization of strands C_1, C_2 is still restricted by mask u and vector (z_1, z_2). As for strands C_3, C_4, their rerandomization can be restricted by mask u and vector $(\overline{z}_1, \overline{z}_2)$, since u is placed on validity checking part.

We stress that the above modifications are carefully conducted to preserve the RCCA security of the encryption scheme. First of all, extra secret keys (e.g., $\mathbf{e}$, $\mathbf{f}$ and $\overline{\mathbf{d}}$) are introduced to compute the additional component in validity checking part such that, given a valid ciphertext ζ, the attacker cannot infer a new validity checking part for particular $[\mathbf{x}]$ or $[\overline{\mathbf{x}}]$ (that cannot be obtained by re-encrypting ζ). Secondly, the usage of mask u in strands C_3, C_4 is safe and sound. Taking component $\left[u\overline{\mathbf{c}}^\top\overline{\mathbf{x}}^\dagger\right]$ as example, it is equivalent to the value of $\left[(u \mod q)\overline{\mathbf{c}}^\top\overline{\mathbf{x}}^\dagger\right]$, as mask u is an integer in $\mathbb{Z}_p^*$. Since the modular operation satisfies the homomorphism property, the re-encryption on strands C_3, C_4 maintains correctness. Note that a component in the validity checking part actually corresponds to two different masks u, u' with $u' = u \mod q$. We remark that this would not affect the RCCA security as long as the size of the modulus q is large enough so that the attacker cannot guess the value of mask u trivially.

Generalization of our approach. Note that the ciphertext structure of our above variant still shares some similarities with that of the PR scheme which is essentially a double "strand" of Cramer-Shoup ciphertext. We turn to explore whether it is possible to generalize our treatment following the CS-paradigm [8].

We start by recalling the CS-paradigm based on SPHF, and then seek to extend the notion of SPHF to interpret our proposed variant and its security.

Recalling Cramer-Shoup paradigm from SPHFs. Smooth Projective Hash Function (SPHF) was originally proposed by Cramer and Shoup [8] for generally constructing practical CCA-secure PKE. Roughly, SPHF is a family of hash functions $\mathcal{H} = (H_{\mathsf{sk}})_{\mathsf{sk}\in\mathcal{K}}$ indexed by $\mathcal{K}$ that map the non-empty element set $\mathcal{X}$ onto the hash value set Π. Each SPHF is associated with an NP-language $\mathcal{L} \subset \mathcal{X}$ where elements in $\mathcal{L}$ are computationally indistinguishable from those in $\mathcal{X}\backslash\mathcal{L}$ (i.e., hard subset membership problem). For any $x \in \mathcal{L}$, $H_{\mathsf{sk}}(x)$ could be efficiently computed using either the hashing key $\mathsf{sk} \in \mathcal{K}$, i.e., $\mathsf{Priv}(\mathsf{sk}, x) = H_{\mathsf{sk}}(x)$ (*private evaluation mode*), or the projection key $\mathsf{pk} = \phi(\mathsf{sk}) \in \mathcal{P}$ with the witness $w \in \mathcal{W}$ to the fact $x \in \mathcal{L}$, i.e., $\mathsf{Pub}(\mathsf{pk}, x, w) = H_{\mathsf{sk}}(x)$ (*public evaluation mode*). The notion of SPHF could be generalized to tag-based SPHF where a tag τ is also taken as an auxiliary input by $H_{(\cdot)}$, Priv and Pub. The $\mathtt{CS}$-paradigm is based on a Smooth_1 $\mathtt{SPHF} = (H_{(\cdot)}, \phi, \mathsf{Priv}, \mathsf{Pub})$ and a Smooth_2 tag-based $\widehat{\mathtt{SPHF}} = (\widehat{H}_{(\cdot)}, \widehat{\phi}, \widehat{\mathsf{Priv}}, \widehat{\mathsf{Pub}})$. The public key is $(\mathsf{pk}, \widehat{\mathsf{pk}}) = (\phi(\mathsf{sk}), \widehat{\phi}(\widehat{\mathsf{sk}}))$ and the ciphertext is

$$\zeta := \Big(x,\ M \cdot \mathsf{Pub}(\mathsf{pk}, x, w),\ \widehat{\mathsf{Pub}}(\widehat{\mathsf{pk}}, x, w, \tau)\Big) = \Big(x,\ M \cdot H_{\mathsf{sk}}(x),\ \widehat{H}_{\widehat{\mathsf{sk}}}(x, \tau)\Big),$$

where $x \in \mathcal{L}$, w is the witness of x, $\tau = \Psi(x,\ M \cdot H_{\mathsf{sk}}(x))$ and Ψ is a collision-resistant hash function. To make our later argument easier to follow, below we first provide an overview of justification of CCA security from SPHF. Consider the challenge ciphertext $\zeta^* = (x^*,\ M_b \cdot \pi^*,\ \widehat{\pi}^*)$ in the CCA security game.

1) Due to the hard subset membership problem, we can replace $x^* \in \mathcal{L}$ in ζ^* with $x^* \in \mathcal{X}\backslash\mathcal{L}$ and compute $\pi^* = \mathsf{Priv}(\mathsf{sk}, x^*)$, $\widehat{\pi}^* = \widehat{\mathsf{Priv}}(\widehat{\mathsf{sk}}, x^*, \tau^*)$.
2) By the Smooth_2 property of tag-based $\widehat{\mathtt{SPHF}}$, any "bad" ciphertext ζ including $x \neq x^* \in \mathcal{X}\backslash\mathcal{L}$ will be rejected by the decryption oracle as $\widehat{\pi} = \widehat{H}_{\widehat{\mathsf{sk}}}(x, \tau)$ is uniformly distributed, even conditioned on $\widehat{\mathsf{pk}}$ and $\widehat{\pi}^*$.
3) By the Smooth_1 property of $\mathtt{SPHF}$, π^* in ζ^* is uniformly distributed and thus ζ^* perfectly hides M_b, which yields the CCA security.

Generalization of our construction via newly extended SPHFs. As the first attempt to generalize our variant, we abstract strands C_1 and C_2 in Eq. (2) using the following SPHFs:

$$\mathtt{SPHF} = (H_{(\cdot)}, \phi, \mathsf{Priv}, \mathsf{Pub}),\ \widehat{\mathtt{SPHF}} = (\widehat{H}_{(\cdot)}, \widehat{\phi}, \widehat{\mathsf{Priv}}, \widehat{\mathsf{Pub}}),\ \widetilde{\mathtt{SPHF}} = (\widetilde{H}_{(\cdot)}, \widetilde{\phi}, \widetilde{\mathsf{Priv}}, \widetilde{\mathsf{Pub}}),$$

based on which C_1 and C_2 in our variant could be written as

$$C_1 := \Big([\mathbf{x}],\ M \cdot H_{\mathsf{sk}}([\mathbf{x}]),\ \boxed{\widehat{H}_{\widehat{\mathsf{sk}}}([\mathbf{x}], \tau)}\Big),\quad C_2 := \Big([\mathbf{y}],\ H_{\mathsf{sk}}([\mathbf{y}]),\ \boxed{\widetilde{H}_{\widetilde{\mathsf{sk}}}([\mathbf{y}], \tau)}\Big),\quad (3)$$

where tag $\tau = (u, m)$, hashing key $\widehat{\mathsf{sk}} = (\mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f})$ and $\widetilde{\mathsf{sk}} = \widehat{\mathsf{sk}}$. Note that these SPHFs are defined on the same set $\mathcal{X} = \big\{[\mathbf{a}]\big|\mathbf{a} \in \mathbb{Z}_p^2\big\}$ with NP-language $\mathcal{L} = \{[r\mathbf{g}]|r \in \mathbb{Z}_p\}$ for $\mathbf{g} \in \mathbb{Z}_p^2$. The rerandomization of C_1 and C_2 is defined as

$$C_1' = \Big([\mathbf{x} + s\mathbf{y}],\ M \cdot \overbrace{\big[\mathbf{b}^\top\mathbf{x}\big] \cdot \big[s\mathbf{b}^\top\mathbf{y}\big]}^{H_{\mathsf{sk}}([\mathbf{x}])\cdot(H_{\mathsf{sk}}([\mathbf{y}]))^s},\ \overbrace{\big[\upsilon u\boldsymbol{\alpha}^\top\mathbf{x}^\dagger\big] \cdot \big[s\upsilon u\boldsymbol{\alpha}^\top\mathbf{y}\big],\ \big[\upsilon u\boldsymbol{\beta}^\top\mathbf{x}^\ddagger\big] \cdot \big[s\upsilon u\boldsymbol{\beta}^\top\mathbf{y}\big]}^{(\widehat{H}_{\widehat{\mathsf{sk}}}([\mathbf{x}],\tau))^\upsilon\cdot(\widetilde{H}_{\widetilde{\mathsf{sk}}}([\mathbf{y}],\tau))^{s\upsilon}}\Big)$$

$$C_2' = \Big([t\mathbf{y}],\ \overbrace{\big[t\mathbf{b}^\top\mathbf{y}\big]}^{(H_{\mathsf{sk}}([\mathbf{y}]))^t},\ \overbrace{\big[t\upsilon \cdot u\boldsymbol{\alpha}^\top\mathbf{y}\big], \big[t\upsilon \cdot u\boldsymbol{\beta}^\top\mathbf{y}\big]}^{(\widetilde{H}_{\widetilde{\mathsf{sk}}}([\mathbf{y}],\tau))^{t\upsilon}}\Big),$$

where $v \leftarrow_{\$} \overline{\mathbb{G}}$, $s, t \leftarrow_{\$} \mathbb{Z}_p^*$. The generalization of strand $C_3(C_4)$ is similar to that of $C_1(C_2)$ and can be denoted by SPHFs defined on the same set $\overline{\mathcal{X}} = \left\{[\overline{\mathbf{a}}] \middle| \overline{\mathbf{a}} \in \mathbb{Z}_q^2\right\}$ with NP-language $\overline{\mathcal{L}} = \{[\overline{r}\overline{\mathbf{g}}] | \overline{r} \in \mathbb{Z}_q\}$ for $\overline{\mathbf{g}} \in \mathbb{Z}_q^2$. The ciphertext rerandomization in our variant could be classified with respect to SPHFs as follows.

- *Self-rerandomization within same* SPHF, e.g.,

$$(H_{\mathsf{sk}}([\mathbf{x}]), H_{\mathsf{sk}}([\mathbf{y}])) \rightsquigarrow H_{\mathsf{sk}}([\mathbf{x}]) \cdot (H_{\mathsf{sk}}([\mathbf{y}]))^s$$

- *Pairwise-rerandomization between different* SPHFs, e.g.,

$$\left(\widehat{H}_{\widehat{\mathsf{sk}}}([\mathbf{x}], \tau), \widetilde{H}_{\widetilde{\mathsf{sk}}}([\mathbf{y}], \tau)\right) \rightsquigarrow \left(\widehat{H}_{\widehat{\mathsf{sk}}}([\mathbf{x}], \tau)\right)^v \cdot \left(\widetilde{H}_{\widetilde{\mathsf{sk}}}([\mathbf{y}], \tau)\right)^{sv}$$

Motivated by these observations, we put forward the notion of *rerandomizable* SPHF (Re-SPHF) which is a regular SPHF augmented with self- and pairwise-rerandomizability. Specifically, based on the typical definition of SPHF, we formalize three extra algorithms namely RandX, RandT and RandH to capture both cases of rerandomization. The correctness of ciphertext in our variant is guaranteed by the *rerandomization correctness* with respect to RandX, RandT and RandH in Re-SPHF, while the perfect rerandomization of ciphertext is captured by the notion of perfect rerandomization in Re-SPHFs.

Arguments of RCCA security with receiver-anonymity. Analogous to the classification of rerandomization, we redefine two types of smoothness for Re-SPHF as below. Let $\mathsf{CRX}(x^*)$ denote the set of all rerandomization of x^* obtained via RandX, $\mathsf{CRX}(x_1^*, x_2^*)$ denote the set of all rerandomization of x_1^* obtained via RandX with x_2^* and $\mathsf{CRT}(\tau^*)$ denote the set of all rerandomization of τ^* obtained via RandT. Let $\overset{s}{\equiv}$ denote statistical indistinguishability between distributions.
– *Controlled-Self-Rerandomizable Smoothness* ($\mathsf{CSR\text{-}Smooth}$). For any $x^* \in \mathcal{X}$, $\tau^* \in \mathcal{T}$ and $(x, \tau) \in \mathcal{X} \backslash \mathcal{L} \times \mathcal{T}$ with $x \notin \mathsf{CRX}(x^*)$ or $\tau \notin \mathsf{CRT}(\tau^*)$,

$$\Big(\mathsf{pk}, H_{\mathsf{sk}}(x^*, \tau^*), \boxed{H_{\mathsf{sk}}(x, \tau)}\ \Big) \overset{s}{\equiv} \Big(\mathsf{pk}, H_{\mathsf{sk}}(x^*, \tau^*), \boxed{\pi} \leftarrow_{\$} \Pi\Big).$$

– *Controlled-Pairwise-Rerandomizable Smoothness* ($\mathsf{CPR\text{-}Smooth}$). For any x_1^*, $x_2^* \in \mathcal{X}$, $\tau^* \in \mathcal{T}$ and $(x, \tau) \in \mathcal{X} \backslash \mathcal{L} \times \mathcal{T}$ with $x \notin \mathsf{CRX}(x_1^*, x_2^*)$ or $\tau \notin \mathsf{CRT}(\tau^*)$,

$$\Big(\widehat{\mathsf{pk}}, \widehat{H}_{\widehat{\mathsf{sk}}}(x_1^*, \tau^*), \widetilde{H}_{\widetilde{\mathsf{sk}}}(x_2^*, \tau^*), \boxed{\widehat{H}_{\widehat{\mathsf{sk}}}(x, \tau)}\ \Big) \overset{s}{\equiv} \Big(\widehat{\mathsf{pk}}, \widehat{H}_{\widehat{\mathsf{sk}}}(x_1^*, \tau^*), \widetilde{H}_{\widetilde{\mathsf{sk}}}(x_2^*, \tau^*), \boxed{\pi} \leftarrow_{\$} \widehat{\Pi}\Big),$$

where $\widehat{\mathsf{sk}} = \widetilde{\mathsf{sk}}$. Also, we redefine two enhanced Smooth_1 for Re-SPHF as below.
– *Self-Twin 1-Smoothness* ($\mathsf{ST\text{-}Smooth}_1$). For $x_1, x_2 \leftarrow_{\$} \mathcal{X} \backslash \mathcal{L}$ and $\tau \leftarrow_{\$} \mathcal{T}$,

$$\Big(\mathsf{pk}, \boxed{H_{\mathsf{sk}}(x_1, \tau)}, \boxed{H_{\mathsf{sk}}(x_2, \tau)}\ \Big) \overset{s}{\equiv} \Big(\mathsf{pk}, \boxed{\pi_1} \leftarrow_{\$} \Pi, \boxed{\pi_2} \leftarrow_{\$} \Pi\Big).$$

– *Pairwise-Twin 1-Smoothness* ($\mathsf{PT\text{-}Smooth}_1$). For $x_1, x_2 \leftarrow_{\$} \mathcal{X} \backslash \mathcal{L}$ and $\tau \leftarrow_{\$} \mathcal{T}$,

$$\Big(\widehat{\mathsf{pk}}, \boxed{\widehat{H}_{\widehat{\mathsf{sk}}}(x_1, \tau)}, \boxed{\widetilde{H}_{\widetilde{\mathsf{sk}}}(x_2, \tau)}\ \Big) \overset{s}{\equiv} \Big(\widehat{\mathsf{pk}}, \boxed{\pi_1} \leftarrow_{\$} \widehat{\Pi}, \boxed{\pi_2} \leftarrow_{\$} \widetilde{\Pi}\Big).$$

We now show how to realize RCCA security and receiver-anonymity with these new properties. Consider a challenge ciphertext ζ^* with words $[\mathbf{x}^*], [\mathbf{y}^*] \in \mathcal{L}$ and $[\overline{\mathbf{x}}^*], [\overline{\mathbf{y}}^*] \in \overline{\mathcal{L}}$ in the RCCA security game. Similar to the security justification of CS-paradigm, below we provide the arguments to justify the RCCA security of our variant.

1) Due to the hard subset membership problems on $(\mathcal{X}, \mathcal{L})$ and $(\overline{\mathcal{X}}, \overline{\mathcal{L}})$, the challenge ciphertext ζ^* generated by alternative encryption algorithm, where $[\mathbf{x}^*], [\mathbf{y}^*] \in \mathcal{L}$ and $[\overline{\mathbf{x}}^*], [\overline{\mathbf{y}}^*] \in \overline{\mathcal{L}}$ are replaced with non-words (i.e., $[\mathbf{x}^*], [\mathbf{y}^*] \in \mathcal{X} \backslash \mathcal{L}$ and $[\overline{\mathbf{x}}^*], [\overline{\mathbf{y}}^*] \in \overline{\mathcal{X}} \backslash \overline{\mathcal{L}}$) and the corresponding hash values are computed with hashing keys, is computationally indistinguishable from one generated by original encryption algorithm.
2) Note that the Smooth_2 property used for proving the CS-paradigm is not satisfied here as the adversary may construct a valid ciphertext with at least one non-word via rerandomizing ζ^*. Fortunately, the manner to rerandomize ζ^* in our variant is restricted by z_1, z_2, $\overline{z}_1$, $\overline{z}_2$, u and querying such a "valid" rerandomization of ζ^* will not leak information about private key. To the end, a computationally unbounded decryption oracle with public key and challenge ciphertext ζ^* only will reject "bad" ciphertext ζ that includes at least one non-word but is not a "valid" rerandomization of ζ^*, as the corresponding hash values (e.g., $\widetilde{H}_{\widetilde{\mathsf{sk}}}([\mathbf{y}], \tau)$ and $\widehat{H}_{\widehat{\mathsf{sk}}}([\mathbf{x}], \tau)$) in ciphertext ζ are uniformly distributed by properties $\mathsf{CSR\text{-}Smooth}$ and $\mathsf{CPR\text{-}Smooth}$.
3) By properties $\mathsf{ST\text{-}Smooth}_1$ and $\mathsf{PT\text{-}Smooth}_1$, all the hash values in ζ^* are uniformly distributed conditioned on public key, and M_b is perfectly hidden in ζ^*, which yields the RCCA security of our variant.

Note that RCCA security guarantees the privacy of the underlying plaintext, while RCCA receiver-anonymity captures the privacy of the public key. The justification for receiver-anonymity is indeed similar to the above arguments. In particular, the decryption oracle also relies on $\mathsf{CSR\text{-}Smooth}$ and $\mathsf{CPR\text{-}Smooth}$ properties to reject all the "bad" ciphertexts. In the end, the uniform distributions of all the hash values in ζ^* imply the receiver-anonymity in RCCA setting.

Related Work. Here we illustrate several previous constructions of Rand-RCCA-secure PKE and provide an efficiency comparison with our scheme, putting aside the receiver-anonymity. Also, some related SPHFs variants will be given.

Non-anonymous constructions. Groth [14] presented a perfect Rand-RCCA-secure scheme, where the ciphertext can be rerandomized into another one in an unlinkable way, under the generic group model, and the ciphertext size expansion is as large as the bit-length of the plaintext. Phan and Pointcheval [21] then designed an efficient framework of RCCA-secure scheme, while Faonio and Fiore [10] showed that the rerandomizability of its ElGamal-based instantiation in [20] cannot resist any active attacks. Chase et al. [6] introduced a new way to construct perfect Rand-RCCA-secure PKE from a malleable NIZK system, where their construction has public verifiability property. Libert et al. [17] proposed a new construction that improves on Chase et al.'s scheme but still suffers from high computational costs and large ciphertext size (of 62 group elements) due

Table 1. Comparison of Rand-RCCA-secure PKE schemes (k=2). $|\mathsf{PK}|$ and $|CT|$ represent the number of elements in public key and ciphertext, where ℓ denotes the bit-length of plaintext. Here $\mathbb{G}$ and $\overline{\mathbb{G}}$ are standard DDH groups that satisfy certain requirements. $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups in bilinear pairing. Here $E, \overline{E}, E_1, E_2, E_T$ denote the execution time of exponentiation on $\mathbb{G}, \overline{\mathbb{G}}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ and the time cost of pairing is P. "Std" refers to standard model, "GGM" refers to generic group model, and "NPR" refers to non-programmable random oracle model. "Perfect" indicates perfect rerandomizability, "Universal" indicates that ciphertext rerandomization does not require the public key, and "Anonymity" refers to RCCA receiver-anonymity.

PKE	Groth04 [14]	PR07 [22]	LPQ17 [17]	FFHR19 [11]	FF20[10]	Ours (k-Lin)
$\lvert\mathsf{PK}\rvert$	$O(\ell)\mathbb{G}$	$4\overline{\mathbb{G}}+7\mathbb{G}$	$11\mathbb{G}_1+16\mathbb{G}_2$	$7\mathbb{G}_1+7\mathbb{G}_2+2\mathbb{G}_T$	$11\mathbb{G}$	$6\overline{\mathbb{G}}+10\mathbb{G}$
$\lvert CT\rvert$	$O(\ell)\mathbb{G}$	$8\overline{\mathbb{G}}+12\mathbb{G}$	$42\mathbb{G}_1+20\mathbb{G}_2$	$3\mathbb{G}_1+2\mathbb{G}_2+\mathbb{G}_T$	$11\mathbb{G}$	$12\overline{\mathbb{G}}+12\mathbb{G}$
Enc	$O(\ell)E$	$8\overline{E}+14E$	$79E_1+64E_2$	$4E_1+5E_2+3E_T+5\mathsf{P}$	$15E$	$12\overline{E}+16E$
Dec	$O(\ell)E$	$8\overline{E}+24E$	$1E_1+142\mathsf{P}$	$8E_1+4E_2+4\mathsf{P}$	$18E$	$18\overline{E}+18E$
Rerand	$O(\ell)E$	$8\overline{E}+16E$	$48E_1+24E_2$	$6E_1+7E_2+3E_T+9\mathsf{P}$	$11E$	$14\overline{E}+14E$
Model	GGM	Std	Std	Std	NPR	Std
Assumption	DDH	DDH	SXDH	$\mathcal{D}_k$-MDDH	DDH	k-Linear
Perfect	✓	✓	✓	✓	×	✓
Universal	×	✓	×	×	×	✓
Anonymity	×	×	×	×	×	✓

to the adoption of NIZK. Recently, Faonio et al. [11] gave a new construction of perfect Rand-RCCA-secure PKE from $\mathcal{D}_k$-MDDH assumption. The ciphertext in their scheme (when $k=1$) is extremely short and consists of only 6 group elements. In a most recent work, Faonio and Fiore [10] proposed a more efficient Rand-RCCA-secure PKE with only weak rerandomizability, and where security is justified in the random oracle model.

In Table 1, we compare our scheme with previous works, putting aside our exclusive property of receiver-anonymity. Compared with the recent work of Faonio et al. [11], our 2-Lin-based instantiation, although based on special groups which are larger than a regular setting, does not involve any pairing computations.

SPHF variants. Variants of SPHF with new properties have also been proposed in the literature [4,7,11,15,27]. Here we briefly introduce two works that are closely related to our Re-SPHF. Wee [27] built the frameworks for constructing PKE satisfying key-dependent message (KDM) security using SPHF with homomorphic hash function. Faonio et al. [11] presented controlled-malleable smooth-projective hash function (cmSPHF), an extension of malleable smooth-projective hash function (mSPHF) by Chen et al. in [7] with respect to elements and tags. However, the cmSPHF cannot support universal rerandomizability.

3 Preliminaries

Let $n \in \mathbb{N}$ denote the security parameter and $\mathsf{negl}(\cdot)$ denote the negligible function. For $\mathbf{x} = (x_1, \cdots, x_n) \in \mathbb{Z}_p^n$ and $g \in \mathbb{G}$, $[\mathbf{x}]$ denotes vector $(g^{x_1}, \cdots, g^{x_n})$. For set $\mathcal{X}$, $x \leftarrow_{\$} \mathcal{X}$ denotes that x is sampled uniformly from $\mathcal{X}$ at random. For any randomized algorithm $\mathcal{F}$, $y \leftarrow_{\$} \mathcal{F}(x)$ denotes the random output of $\mathcal{F}$.

3.1 Public-Key Encryption (PKE)

A PKE scheme consists of algorithms $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$: $\mathsf{KGen}(1^n)$ takes as input the security parameter 1^n, and outputs the key pair $(\mathsf{PK}, \mathsf{SK})$; The encryption algorithm $\mathsf{Enc}(\mathsf{PK}, M)$ takes as input the public key PK and the plaintext M, and outputs the ciphertext ζ; The decryption algorithm $\mathsf{Dec}(\mathsf{SK}, \zeta)$ takes as input the secret key SK and the ciphertext ζ, and outputs the plaintext M or $\perp$.

A PKE scheme should satisfy *decryption correctness* which captures the fact that, for $(\mathsf{PK}, \mathsf{SK}) \leftarrow_\$ \mathsf{KGen}(1^n)$, for any $M \in \mathcal{M}$ (in valid message space),

$$\Pr\left[\mathsf{Dec}(\mathsf{SK}, \zeta) \neq M : \zeta \leftarrow_\$ \mathsf{Enc}(\mathsf{PK}, M)\right] \leq \mathsf{negl}(n)\,.$$

Below we provide the definitions of rerandomizable PKE. As mentioned above, in this work, we are mainly interested in "universal rerandomization" that does not require the public key, which is crucial to realize receiver-anonymity. Therefore, we mainly follow the definitions given in [22].

Rerandomizable PKE. We say a PKE scheme is (universally) *rerandomizable* if there exists algorithm Rerand that takes as input ciphertext ζ and outputs a new ciphertext ζ'; and for $(\mathsf{PK}, \mathsf{SK}) \leftarrow_\$ \mathsf{KGen}(1^n)$, any (possibly malicious) ciphertext ζ,

$$\Pr\left[\mathsf{Dec}(\mathsf{SK}, \zeta') \neq \mathsf{Dec}(\mathsf{SK}, \zeta) : \zeta' \leftarrow_\$ \mathsf{Rerand}(\zeta)\right] \leq \mathsf{negl}(n)\,.$$

Definition 1 (Perfectly Rerandomizable PKE [11]). *Assume* $\mathtt{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Rerand})$ *is rerandomizable. We say* $\mathtt{PKE}$ *is perfectly rerandomizable if following properties are satisfied.*

- *For* $(\mathsf{PK}, \mathsf{SK}) \leftarrow_\$ \mathsf{KGen}(1^n)$*, any* $M \in \mathcal{M}$ *and any (honestly generated) ciphertext* ζ *in the support of* $\mathsf{Enc}(\mathsf{PK}, M)$*, the distribution of* $\mathsf{Rerand}(\zeta)$ *is identical to that of* $\mathsf{Enc}(\mathsf{PK}, M)$*.*
- *For* $(\mathsf{PK}, \mathsf{SK}) \leftarrow_\$ \mathsf{KGen}(1^n)$ *and any (possibly unbounded) adversary* $\mathcal{A}$*, given* PK*, the probability of* $\mathcal{A}$ *generating a ciphertext* ζ *such that* $\mathsf{Dec}(\mathsf{SK}, \zeta) = M \neq \perp$ *for some* M *and* ζ *is not in the range of* $\mathsf{Enc}(\mathsf{PK}, M)$ *is negligible.*

Coupled with the second property, called the tightness of decryption in both [22] and [11], the first property can be extended to any malicious ciphertext that decrypts successfully.

Malleable PKE. We say a PKE scheme is *malleable* if there exists an algorithm Maul that takes as input a ciphertext ζ and a message M', and outputs a new ciphertext ζ'; and for $(\mathsf{PK}, \mathsf{SK}) \leftarrow_\$ \mathsf{KGen}(1^n)$, any $M, M' \in \mathcal{M}$ and $\zeta \leftarrow_\$ \mathsf{Enc}(\mathsf{PK}, M)$,

$$\Pr\left[\mathsf{Dec}(\mathsf{SK}, \zeta') \neq M \cdot M' : \zeta' \leftarrow_\$ \mathsf{Maul}(\zeta, M')\right] \leq \mathsf{negl}(n)\,.$$

W.l.o.g., we assume that message space $\mathcal{M}$ is a multiplicative group, and let "$\cdot$" denote multiplication operation on $\mathcal{M}$.

Security definitions. We follow the definitions of RCCA security and RCCA receiver-anonymity in [22].

```
$\mathsf{IND\text{-}RCCA}^{\mathcal{A}}_{\mathtt{PKE}}(n)$
    $(\mathsf{PK}, \mathsf{SK}) \leftarrow_\$ \mathsf{KGen}(1^n)$
    $(M_0, M_1) \leftarrow \mathcal{A}^{\mathcal{DO}_{\mathsf{SK}}}(\mathsf{PK})$
    $b \leftarrow_\$ \{0, 1\}$
    $\zeta^* \leftarrow_\$ \mathsf{Enc}(\mathsf{PK}, M_b)$
    $b' \leftarrow \mathcal{A}^{\mathcal{GDO}^{M_0, M_1}_{\mathsf{SK}}}(\mathsf{PK}, \zeta^*)$
    if $b = b'$, return $1$
    else return $0$

$\mathcal{DO}_{\mathsf{SK}}(\zeta)$
    return $\mathsf{Dec}(\mathsf{SK}, \zeta)$

$\mathcal{GDO}^{M_0, M_1}_{\mathsf{SK}}(\zeta)$
    $M := \mathsf{Dec}(\mathsf{SK}, \zeta)$
    if $M \in \{M_0, M_1\}$, return replay
    else return $M$
```

Fig. 1. Definition of IND-RCCA game.

```
$\mathsf{ANON\text{-}RCCA}^{\mathcal{A}}_{\mathtt{PKE}}(n)$
    $(\mathsf{PK}_0, \mathsf{SK}_0) \leftarrow_\$ \mathsf{KGen}(1^n)$
    $(\mathsf{PK}_1, \mathsf{SK}_1) \leftarrow_\$ \mathsf{KGen}(1^n)$
    $M \leftarrow \mathcal{A}^{\mathcal{DO}_{\mathsf{SK}_0, \mathsf{SK}_1}}(\mathsf{PK}_0, \mathsf{PK}_1)$
    $b \leftarrow_\$ \{0, 1\}$
    $\zeta^* \leftarrow_\$ \mathsf{Enc}(\mathsf{PK}_b, M)$
    $b' \leftarrow \mathcal{A}^{\mathcal{GDO}^{M}_{\mathsf{SK}_0, \mathsf{SK}_1}}(\mathsf{PK}_0, \mathsf{PK}_1, \zeta^*)$
    if $b = b'$, return $1$
    else return $0$

$\mathcal{DO}_{\mathsf{SK}_0, \mathsf{SK}_1}(\zeta)$
    return $(\mathsf{Dec}(\mathsf{SK}_0, \zeta), \mathsf{Dec}(\mathsf{SK}_1, \zeta))$

$\mathcal{GDO}^{M}_{\mathsf{SK}_0, \mathsf{SK}_1}(\zeta)$
    $M_0 := \mathsf{Dec}(\mathsf{SK}_0, \zeta)$; $M_1 := \mathsf{Dec}(\mathsf{SK}_1, \zeta)$
    if $M \in \{M_0, M_1\}$, return replay
    else return $(M_0, M_1)$
```

Fig. 2. Definition of ANON-RCCA game.

Definition 2 (RCCA Security). *Let* PKE = (KGen, Enc, Dec) *be a PKE scheme. Consider the security game* $\mathsf{IND\text{-}RCCA}^{\mathcal{A}}_{\mathtt{PKE}}(n)$ *in Fig. 1. We say* PKE *is RCCA-secure if for any PPT algorithm* $\mathcal{A}$ *in game* $\mathsf{IND\text{-}RCCA}^{\mathcal{A}}_{\mathtt{PKE}}(n)$,

$$\mathsf{Adv}^{\mathsf{IND\text{-}RCCA}}_{\mathcal{A},\mathsf{PKE}}(n) := \left| \Pr\left[\mathsf{IND\text{-}RCCA}^{\mathcal{A}}_{\mathtt{PKE}}(n) = 1 \right] - \frac{1}{2} \right| \leq \mathsf{negl}(n) .$$

Definition 3 (RCCA Receiver-Anonymity). *Let* PKE = (KGen, Enc, Dec) *be a PKE scheme. Consider the security game* $\mathsf{ANON\text{-}RCCA}^{\mathcal{A}}_{\mathtt{PKE}}(n)$ *in Fig. 2. We say* PKE *is RCCA receiver-anonymous if for any PPT algorithm* $\mathcal{A}$ *in game* $\mathsf{ANON\text{-}RCCA}^{\mathcal{A}}_{\mathtt{PKE}}(n)$,

$$\mathsf{Adv}^{\mathsf{ANON\text{-}RCCA}}_{\mathcal{A},\mathsf{PKE}}(n) := \left| \Pr\left[\mathsf{ANON\text{-}RCCA}^{\mathcal{A}}_{\mathtt{PKE}}(n) = 1 \right] - \frac{1}{2} \right| \leq \mathsf{negl}(n) .$$

3.2 Smooth Projective Hash Function (SPHF)

In this work, we focus on a more general version of smooth projective hash function, called tag-based smooth projective hash function (tag-SPHF) [8]. The regular SPHF can be regarded as a special case of tag-SPHF with empty tag space $\mathcal{T} = \emptyset$. A tag-SPHF is associated with set $\mathcal{X}$, NP-language $\mathcal{L}$ where $\mathcal{L} \subset \mathcal{X}$, and defined by four algorithms $(\mathsf{Setup}, \phi, \mathsf{Priv}, \mathsf{Pub})$ as follows:

- $\mathsf{Setup}(1^n)$ takes as input a security parameter 1^n, and outputs public parameters $\mathsf{pp} = \big(\mathcal{K}, \mathcal{T}, \Pi, H_{(\cdot)}\big)$, where $\mathcal{K}$ is the hashing key space, $\mathcal{T}$ is the tag space, Π is the hash value space, $H_{(\cdot)} : \mathcal{X} \times \mathcal{T} \to \Pi$ is an efficiently computable hash function family indexed by hashing key $\mathsf{sk} \in \mathcal{K}$.
- $\phi(\mathsf{sk})$ derives the projection key pk from the hashing key $\mathsf{sk} \in \mathcal{K}$.
- $\mathsf{Priv}(\mathsf{sk}, x, \tau)$ takes as input an element $x \in \mathcal{X}$, tag $\tau \in \mathcal{T}$ and hashing key sk, and outputs hash value $\pi = H_{\mathsf{sk}}(x, \tau) \in \Pi$.
- $\mathsf{Pub}(\mathsf{pk}, x, w, \tau)$ takes as input a word $x \in \mathcal{L}$ with witness w, tag τ and projection key pk, and outputs hash value $\pi = H_{\mathsf{sk}}(x, \tau) \in \Pi$.

In regular SPHF, both the input of algorithms $\mathsf{Priv}(\mathsf{sk}, x)$ and $\mathsf{Pub}(\mathsf{pk}, x, w)$ do not include tag τ, and the outputted hash value is $\pi = H_{\mathsf{sk}}(x)$.

Definition 4 (Correctness). *For* $\mathsf{pp} \leftarrow_{\$} \mathsf{Setup}(1^n)$, $\mathsf{sk} \leftarrow_{\$} \mathcal{K}$ *and* $\mathsf{pk} = \phi(\mathsf{sk})$, *any* $x \in \mathcal{L}$ *with witness* w *to the fact of* $x \in \mathcal{L}$ *and any* $\tau \in \mathcal{T}$,

$$\Pr\left[\mathsf{Priv}(\mathsf{sk}, x, \tau) \neq \mathsf{Pub}(\mathsf{pk}, x, w, \tau)\right] \leq \mathsf{negl}(n).$$

Assume that $\mathtt{SPHF} = (\mathsf{Setup}, \phi, \mathsf{Priv}, \mathsf{Pub})$ is associated with $\mathcal{X}$, $\mathcal{L}$ and $\mathcal{T}$.

Definition 5 (1-Smoothness). *We say* $\mathtt{SPHF}$ *is* Smooth_1 *if for* $\mathsf{pp} \leftarrow_{\$} \mathsf{Setup}(1^n)$, $\mathsf{sk} \leftarrow_{\$} \mathcal{K}$, $\mathsf{pk} = \phi(\mathsf{sk})$ *and any* $(x, \tau) \in \mathcal{X} \backslash \mathcal{L} \times \mathcal{T}$, *the following two distributions are statistically indistinguishable:*

$$V_1 = \{(\mathsf{pk}, x, \tau, \pi) | \pi = H_{\mathsf{sk}}(x, \tau)\}, \quad V_2 = \{(\mathsf{pk}, x, \tau, \pi') | \pi' \leftarrow_{\$} \Pi\}.$$

For certain tag-SPHFs, the smoothness property may be enhanced as follows.

Definition 6 (2-Smoothness). *We say* $\mathtt{SPHF}$ *is* Smooth_2 *if for* $\mathsf{pp} \leftarrow_{\$} \mathsf{Setup}(1^n)$, $\mathsf{sk} \leftarrow_{\$} \mathcal{K}$, $\mathsf{pk} = \phi(\mathsf{sk})$, *any* $(x^*, \tau^*) \in \mathcal{X} \times \mathcal{T}$ *and any* $(x, \tau) \in \mathcal{X} \backslash \mathcal{L} \times \mathcal{T}$ *with* $(x, \tau) \neq (x^*, \tau^*)$, *the following two distributions are statistically indistinguishable:*

$$V_1 = \{(\mathsf{pk}, x^*, \tau^*, x, \tau, H_{\mathsf{sk}}(x^*, \tau^*), \pi) | \pi = H_{\mathsf{sk}}(x, \tau)\},$$
$$V_2 = \{(\mathsf{pk}, x^*, \tau^*, x, \tau, H_{\mathsf{sk}}(x^*, \tau^*), \pi') | \pi' \leftarrow_{\$} \Pi\}.$$

We assume that it is efficient to sample elements from set $\mathcal{X}$ and $\mathcal{L}$. Below we define the hard subset membership problem (SMP) between $\mathcal{X}$ and $\mathcal{L}$.

Definition 7 (Hard Subset Membership Problem). *We say the subset membership problem is hard on* $(\mathcal{X}, \mathcal{L})$ *if for any PPT adversary* $\mathcal{A}$,

$$|\Pr[\mathcal{A}(x) = 1] - \Pr[\mathcal{A}(x') = 1]| \leq \mathsf{negl}(n),$$

where $x \leftarrow_{\$} \mathcal{L}$ *and* $x' \leftarrow_{\$} \mathcal{X}$.

4 Rerandomizable Tag-SPHF

4.1 Syntax of Rerandomizable Tag-SPHF

We slightly extend the typical SPHF syntax in such a way that the hash function family H is indexed not only by the hashing key $\mathsf{sk} \in \mathcal{K}$ (as the typical case) but also by some (possible) auxiliary information ax, which is fixed as part of the public parameter. For generality and simplicity considerations, hereafter we assume that such information is public and implicitly included in the description of hash function family, and remain to use $H_{(\cdot)}$ instead of $H_{\mathsf{ax},(\cdot)}$. Note that ax is set as "null" for typical SPHFs. We remark that since now the hash function family is not solely indexed by the hash key, for two SPHFs that are even with the same $(\mathcal{X}, \mathcal{L}, \mathcal{K}, \mathcal{T}, \Pi)$, their corresponding hash function families are not necessarily the same due to the possibly different auxiliary index ax.

Definition 8 (Rerandomizable Tag-SPHF (Re-T-SPHF)). *Let I and I' be two tag-SPHFs associated with same sets $\mathcal{X}$ and $\mathcal{L}$, sharing partially the same public parameter $(\mathcal{K}, \mathcal{T}, \Pi)$ but having (possibly) different hash function families $H_{(\cdot)}$ and $H'_{(\cdot)}$. We say I is* ***pairwise-rerandomizable*** *with respect to I' if:*

- *There exist three efficient algorithms as below.*
  - *$I.\mathsf{RandX}(x, x', r_x)$ takes as input elements $x, x' \in \mathcal{X}$ and randomness $r_x \in \mathcal{R}_x$, outputs a new element $x^* \in \mathcal{X}$;*
  - *$I.\mathsf{RandT}(\tau, r_\tau)$ takes as input tag $\tau \in \mathcal{T}$ and randomness $r_\tau \in \mathcal{R}_\tau$, outputs a new tag $\tau^* \in \mathcal{T}$;*
  - *$I.\mathsf{RandH}(\pi, \pi', r_x, r_\tau)$ takes as input hash values $\pi, \pi' \in \Pi$ and randomnesses $r_x \in \mathcal{R}_x, r_\tau \in \mathcal{R}_\tau$, outputs a rerandomized hash value $\pi^* \in \Pi$,*

  where $\mathcal{R}_x$ and $\mathcal{R}_\tau$ are randomness space for element and tag respectively.
- *For $\mathsf{sk} \leftarrow_\$ \mathcal{K}$, any $x, x' \in \mathcal{X}$, any $\tau \in \mathcal{T}$, let $\pi = H_{\mathsf{sk}}(x, \tau)$ and $\pi' = H'_{\mathsf{sk}}(x', \tau)$,*

$$\Pr\left[H_{\mathsf{sk}}(x^*, \tau^*) \neq \pi^* : \begin{array}{c} r_x \leftarrow_\$ \mathcal{R}_x;\ r_\tau \leftarrow_\$ \mathcal{R}_\tau \\ x^* := I.\mathsf{RandX}(x, x', r_x); \\ \tau^* := I.\mathsf{RandT}(\tau, r_\tau) \\ \pi^* := I.\mathsf{RandH}(\pi, \pi', r_x, r_\tau) \end{array} \right] \leq \mathsf{negl}(n).$$

If $I' = I$[1]*, we say that I is* ***self-rerandomizable****. In this case, the input x and x' for algorithm RandX could be the same element. We say that I is linearly rerandomizable if for any $\pi, \pi', \Delta \in \Pi$ (w.l.o.g., considering Π as a multiplicative group), $r_x \leftarrow_\$ \mathcal{R}_x$, $r_\tau \leftarrow_\$ \mathcal{R}_\tau$, $I.\mathsf{RandH}(\pi \cdot \Delta, \pi', r_x, r_\tau) = I.\mathsf{RandH}(\pi, \pi', r_x, r_\tau) \cdot \Delta$.*

Remark (Re-SPHF). For a regular rerandomizable SPHF (hereafter referred to as Re-SPHF) where tag space $\mathcal{T} = \emptyset$, the algorithm RandT is absent and the parameter r_τ in the input of algorithm RandH is explicitly omitted.

[1] That is, $H_{(\cdot)}$ and $H'_{(\cdot)}$ have the same auxiliary index (which could be "null"), and thus are the same (since they work on the same $\mathcal{K}$).

Definition 9 (Perfect Re-T-SPHF). *Assume I is pairwise-rerandomizable with respect to I'. We say that I is* ***perfectly rerandomizable*** *on $\mathcal{T}_s$ with respect to I' if for $\mathsf{sk} \leftarrow_{\$} \mathcal{K}$, any $x, x' \in \mathcal{X}$, any $\tau \in \mathcal{T}_s \subseteq \mathcal{T}$, $r_x \leftarrow_{\$} \mathcal{R}_x$, $r_\tau \leftarrow_{\$} \mathcal{R}_\tau$ and $\pi = H_{\mathsf{sk}}(x, \tau)$, $\pi' = H'_{\mathsf{sk}}(x', \tau)$, the following distributions are identical:*

$$V_1 = \{(x'', \tau'', \pi'') | x'' \leftarrow_{\$} \mathcal{X};\ \tau'' \leftarrow_{\$} \mathcal{T}_s;\ \pi'' = H_{\mathsf{sk}}(x'', \tau'')\},$$
$$V_2 = \left\{ (x^*, \tau^*, \pi^*) \middle| \begin{array}{l} x^* := I.\mathsf{RandX}(x, x', r_x);\ \tau^* := I.\mathsf{RandT}(\tau, r_\tau) \\ \pi^* := I.\mathsf{RandH}(\pi, \pi', r_x, r_\tau) \end{array} \right\}.$$

If $\mathcal{T}_s = \mathcal{T}$, we say I is ***perfectly pairwise-rerandomizable*** *with respect to I'. If $I' = I$, we say I is* ***perfectly self-rerandomizable*** *on $\mathcal{T}_s$.*

4.2 Redefining Smoothness for Re-T-SPHFs

We define the property of smoothness for Re-T-SPHFs as below.

Definition 10 (Controlled-Self-Rerandomizable Smoothness). *Let I be self-rerandomizable. Assume it is associated with sets $\mathcal{X}$ and $\mathcal{L}$, and the public parameter is $(\mathcal{K}, \mathcal{T}, \Pi, H_{(\cdot)})$. Denote $\mathsf{CRX}(x) = \{I.\mathsf{RandX}(x, x, r_x) | r_x \in \mathcal{R}_x\}$ and $\mathsf{CRT}(\tau) = \{I.\mathsf{RandT}(\tau, r_\tau) | r_\tau \in \mathcal{R}_\tau\}$. We say I satisfies* ***controlled-self-rerandomizable smoothness*** (CSR-Smooth) *if for $\mathsf{sk} \leftarrow_{\$} \mathcal{K}$ and $\mathsf{pk} := I.\phi(\mathsf{sk})$, any $(x^*, \tau^*) \in \mathcal{X} \times \mathcal{T}$ and any $(x, \tau) \in \mathcal{X} \backslash \mathcal{L} \times \mathcal{T}$ with $x \notin \mathsf{CRX}(x^*)$ or $\tau \notin \mathsf{CRT}(\tau^*)$, the following two distributions are statistically indistinguishable,*

$$V_1 = \{(\mathsf{pk}, x^*, x, H_{\mathsf{sk}}(x^*, \tau^*), \pi) | \pi = H_{\mathsf{sk}}(x, \tau)\},$$
$$V_2 = \{(\mathsf{pk}, x^*, x, H_{\mathsf{sk}}(x^*, \tau^*), \pi') | \pi' \leftarrow_{\$} \Pi\}.$$

Definition 11 (Controlled-Pairwise-Rerandomizable Smoothness). *Let I be pairwise-rerandomizable with respect to I'. Assume they are associated with sets $\mathcal{X}$ and $\mathcal{L}$, and work on $(\mathcal{K}, \mathcal{T}, \Pi)$. Let $H_{(\cdot)}$ and $H'_{(\cdot)}$ be the hash function family of I and I' respectively. Denote $\mathsf{CRX}(x, x') = \{I.\mathsf{RandX}(x, x', r_x) | r_x \in \mathcal{R}_x\}$ and $\mathsf{CRT}(\tau) = \{I.\mathsf{RandT}(\tau, r_\tau) | r_\tau \in \mathcal{R}_\tau\}$. We say I satisfies* ***controlled-pairwise-rerandomizable smoothness*** (CPR-Smooth) *with respect to I' if for $\mathsf{sk} \leftarrow_{\$} \mathcal{K}$ and $\mathsf{pk} := I.\phi(\mathsf{sk})$, any $(x_1^*, \tau_1^*), (x_2^*, \tau_2^*) \in \mathcal{X} \times \mathcal{T}$ with $\tau_1^* = \tau_2^*$ and any $(x, \tau) \in \mathcal{X} \backslash \mathcal{L} \times \mathcal{T}$ with $x \notin \mathsf{CRX}(x_1^*, x_2^*)$ or $\tau \notin \mathsf{CRT}(\tau_1^*)$, the following two distributions are statistically indistinguishable:*

$$V_1 = \{(\mathsf{pk}, x_1^*, x_2^*, x, H_{\mathsf{sk}}(x_1^*, \tau_1^*), H'_{\mathsf{sk}}(x_2^*, \tau_2^*), \pi) | \pi = H_{\mathsf{sk}}(x, \tau)\},$$
$$V_2 = \{(\mathsf{pk}, x_1^*, x_2^*, x, H_{\mathsf{sk}}(x_1^*, \tau_1^*), H'_{\mathsf{sk}}(x_2^*, \tau_2^*), \pi') | \pi' \leftarrow_{\$} \Pi\}.$$

Definition 12 (Self-Twin 1-Smoothness). *Let I be self-rerandomizable. Assume it is associated with sets $\mathcal{X}$ and $\mathcal{L}$, and the public parameter is $(\mathcal{K}, \mathcal{T}, \Pi, H_{(\cdot)})$. We say I satisfies* ***self-twin 1-smoothness*** (ST-Smooth_1) *if for $\mathsf{sk} \leftarrow_{\$} \mathcal{K}$ and $\mathsf{pk} := I.\phi(\mathsf{sk})$, $x^*, x \leftarrow_{\$} \mathcal{X} \backslash \mathcal{L}$, $\tau \leftarrow_{\$} \mathcal{T}$, the following two distributions are statistically indistinguishable:*

$$V_1 = \{(\mathsf{pk}, x^*, x, \tau, \pi^*, \pi) | \pi^* = H_{\mathsf{sk}}(x^*, \tau), \pi = H_{\mathsf{sk}}(x, \tau)\},$$
$$V_2 = \{(\mathsf{pk}, x^*, x, \tau, \pi'', \pi') | \pi'', \pi' \leftarrow_{\$} \Pi\}.$$

Definition 13 (Pairwise-Twin 1-Smoothness). *Let I be pairwise-rerandomizable with respect to I'. Assume they are associated with sets $\mathcal{X}$ and $\mathcal{L}$, and work on $(\mathcal{K}, \mathcal{T}, \Pi)$. Let $H_{(\cdot)}$ and $H'_{(\cdot)}$ be the hash function family of I and I' respectively. We say I satisfies* ***pairwise-twin 1-smoothness*** $(\mathsf{PT\text{-}Smooth}_1)$ *with respect to I' if for $\mathsf{sk} \leftarrow_\$ \mathcal{K}$ and $\mathsf{pk} := I.\phi(\mathsf{sk})$, $x^*, x \leftarrow_\$ \mathcal{X} \backslash \mathcal{L}$, $\tau \leftarrow_\$ \mathcal{T}$, the following two distributions are statistically indistinguishable:*

$$V_1 = \{(\mathsf{pk}, x^*, x, \tau, \pi^*, \pi) | \pi^* = H_{\mathsf{sk}}(x^*, \tau), \pi = H'_{\mathsf{sk}}(x, \tau)\},$$
$$V_2 = \{(\mathsf{pk}, x^*, x, \tau, \pi'', \pi') | \pi'', \pi' \leftarrow_\$ \Pi\}.$$

5 A General Framework of Rand-RCCA-secure PKE

5.1 Our Generic Construction

The generic construction of the anonymous Rand-RCCA-secure scheme `PKE` = (KGen, Enc, Dec, Rerand) is depicted in Fig. 3 where the sub-scheme `MPKE` = (MKGen, MEnc, MDec, MRerand, Maul) is given in Fig. 4.

```
$\mathsf{KGen}(1^n)$
  $\mathsf{sk}_0 \leftarrow_\$ \mathcal{K}_0$;  $\mathsf{sk}_1 \leftarrow_\$ \mathcal{K}_1$
  $\mathsf{pk}_0 := I_0.\phi(\mathsf{sk}_0)$;  $\mathsf{pk}_1 := I_1.\phi(\mathsf{sk}_1)$
  $\mathsf{sk}_2 := \mathsf{sk}_1$;  $\mathsf{pk}_2 := \mathsf{pk}_1$
  $(\mathsf{mpk}, \mathsf{msk}) \leftarrow_\$ \mathsf{MKGen}(1^n)$
  $\mathsf{SK} := (\mathsf{sk}_0, \mathsf{sk}_1, \mathsf{sk}_2, \mathsf{msk})$
  $\mathsf{PK} := (\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{mpk})$
  return $(\mathsf{PK}, \mathsf{SK})$
```

```
$\mathsf{Enc}(\mathsf{PK}, M \in \Pi_0)$
  $x_1 \leftarrow_\$ \mathcal{L}$ with witness $w_1$
  $x_2 \leftarrow_\$ \mathcal{L}$ with witness $w_2$
  $u \leftarrow_\$ \overline{\Pi}_0$;  $\tau := (u, \psi(M))$
  $e_1 := I_0.\mathsf{Pub}(\mathsf{pk}_0, x_1, w_1) \cdot M$
  $\widehat{\pi}_1 := I_1.\mathsf{Pub}(\mathsf{pk}_1, x_1, w_1, \tau)$
  $\pi_2 := I_0.\mathsf{Pub}(\mathsf{pk}_0, x_2, w_2)$
  $\widetilde{\pi}_2 := I_2.\mathsf{Pub}(\mathsf{pk}_2, x_2, w_2, \tau)$
  $\varrho \leftarrow_\$ \mathsf{MEnc}(\mathsf{mpk}, u)$
  return $\zeta := (x_1, e_1, \widehat{\pi}_1, x_2, \pi_2, \widetilde{\pi}_2, \varrho)$
```

```
$\mathsf{Dec}(\mathsf{SK}, \zeta)$
  $u := \mathsf{MDec}(\mathsf{msk}, \varrho)$;  if $u = \perp$, return $\perp$
  $\pi_1' := I_0.\mathsf{Priv}(\mathsf{sk}_0, x_1)$;  $\pi_2' := I_0.\mathsf{Priv}(\mathsf{sk}_0, x_2)$
  $M := e_1 \cdot \pi_1'^{-1}$;  $\tau := (u, \psi(M))$
  $\widehat{\pi}_1' := I_1.\mathsf{Priv}(\mathsf{sk}_1, x_1, \tau)$
  $\widetilde{\pi}_2' := I_2.\mathsf{Priv}(\mathsf{sk}_2, x_2, \tau)$
  if $(\widehat{\pi}_1', \widetilde{\pi}_2', \pi_2') \neq (\widehat{\pi}_1, \widetilde{\pi}_2, \pi_2)$, return $\perp$
  else return $M$
```

```
$\mathsf{Rerand}(\zeta)$
  $r_1, r_2 \leftarrow_\$ \mathcal{R}_x$;  $r_\tau \leftarrow_\$ \overline{\Pi}_0$
  $x_1' := I_0.\mathsf{RandX}(x_1, x_2, r_1)$
  $x_2' := I_0.\mathsf{RandX}(x_2, x_2, r_2)$
  $e_1' := I_0.\mathsf{RandH}(e_1, \pi_2, r_1)$
  $\widehat{\pi}_1' := I_1.\mathsf{RandH}(\widehat{\pi}_1, \widetilde{\pi}_2, r_1, r_\tau)$
  $\pi_2' := I_0.\mathsf{RandH}(\pi_2, \pi_2, r_2)$
  $\widetilde{\pi}_2' := I_2.\mathsf{RandH}(\widetilde{\pi}_2, \widetilde{\pi}_2, r_2, r_\tau)$
  $\varrho' := \mathsf{MRerand}(\mathsf{Maul}(\varrho, r_\tau))$
  return $\zeta' := (x_1', e_1', \widehat{\pi}_1', x_2', \pi_2', \widetilde{\pi}_2', \varrho')$
```

Fig. 3. Our anonymous Rand-RCCA-secure scheme `PKE`

```
$\mathsf{MKGen}(1^n)$
    $\overline{\mathsf{sk}}_0 \leftarrow_\$ \overline{\mathcal{K}}_0$; $\mathsf{sk}_3 \leftarrow_\$ \mathcal{K}_3$
    $\overline{\mathsf{pk}}_0 := \overline{I}_0.\phi(\overline{\mathsf{sk}}_0)$
    $\mathsf{pk}_3 := I_3.\phi(\mathsf{sk}_3)$
    $\mathsf{sk}_4 := \mathsf{sk}_3$; $\mathsf{pk}_4 := \mathsf{pk}_3$
    $\mathsf{msk} := (\overline{\mathsf{sk}}_0, \mathsf{sk}_3, \mathsf{sk}_4)$
    $\mathsf{mpk} := (\overline{\mathsf{pk}}_0, \mathsf{pk}_3, \mathsf{pk}_4)$
    return $(\mathsf{mpk}, \mathsf{msk})$

$\mathsf{MEnc}(\mathsf{mpk}, u \in \overline{\Pi}_0)$
    $x_3 \leftarrow_\$ \overline{\mathcal{L}}$ with witness $w_3$
    $x_4 \leftarrow_\$ \overline{\mathcal{L}}$ with witness $w_4$
    $\tau := u$
    $e_3 := \overline{I}_0.\mathsf{Pub}(\overline{\mathsf{pk}}_0, x_3, w_3) \cdot u$
    $\widehat{\pi}_3 := I_3.\mathsf{Pub}(\mathsf{pk}_3, x_3, w_3, \tau)$
    $\pi_4 := \overline{I}_0.\mathsf{Pub}(\overline{\mathsf{pk}}_0, x_4, w_4)$
    $\widetilde{\pi}_4 := I_4.\mathsf{Pub}(\mathsf{pk}_4, x_4, w_4, \tau)$
    $\varrho := (x_3, e_3, \widehat{\pi}_3, x_4, \pi_4, \widetilde{\pi}_4)$
    return $\varrho$

$\mathsf{MDec}(\mathsf{msk}, \varrho)$
    $\pi_3' := \overline{I}_0.\mathsf{Priv}(\overline{\mathsf{sk}}_0, x_3)$; $u := e_3 \cdot \pi_3'^{-1}$
    $\pi_4' := \overline{I}_0.\mathsf{Priv}(\overline{\mathsf{sk}}_0, x_4)$
    $\widehat{\pi}_3' := I_3.\mathsf{Priv}(\mathsf{sk}_3, x_3, u)$; $\widetilde{\pi}_4' := I_4.\mathsf{Priv}(\mathsf{sk}_4, x_4, u)$
    if $(\widehat{\pi}_3', \widetilde{\pi}_4', \pi_4') \neq (\widehat{\pi}_3, \widetilde{\pi}_4, \pi_4)$, return $\perp$
    else return $u$

$\mathsf{Maul}(\varrho, r_\tau \in \overline{\Pi}_0)$
    $\widehat{\pi}_3' := I_3.\mathsf{RandH}(\widehat{\pi}_3, \widetilde{\pi}_4, 1_{\overline{\mathcal{R}}_x}, r_\tau)$
    $\widetilde{\pi}_4' := I_4.\mathsf{RandH}(\widetilde{\pi}_4, \widetilde{\pi}_4, 1_{\overline{\mathcal{R}}_x}, r_\tau)$
    return $\varrho' := (x_3, e_3 \cdot r_\tau, \widehat{\pi}_3', x_4, \pi_4, \widetilde{\pi}_4')$

$\mathsf{MRerand}(\varrho)$
    $r_3, r_4 \leftarrow_\$ \overline{\mathcal{R}}_x$
    $x_3' := \overline{I}_0.\mathsf{RandX}(x_3, x_4, r_3)$; $x_4' := \overline{I}_0.\mathsf{RandX}(x_4, x_4, r_4)$
    $e_3' := \overline{I}_0.\mathsf{RandH}(e_3, \pi_4, r_3)$; $\pi_4' := \overline{I}_0.\mathsf{RandH}(\pi_4, \pi_4, r_4)$
    $\widehat{\pi}_3' := I_3.\mathsf{RandH}(\widehat{\pi}_3, \widetilde{\pi}_4, r_3, 1_{\overline{\Pi}_0})$
    $\widetilde{\pi}_4' := I_4.\mathsf{RandH}(\widetilde{\pi}_4, \widetilde{\pi}_4, r_4, 1_{\overline{\Pi}_0})$
    return $\varrho' := (x_3', e_3', \widehat{\pi}_3', x_4', \pi_4', \widetilde{\pi}_4')$
```

Fig. 4. Generic rerandomizable and malleable encryption scheme MPKE

Table 2. Descriptions of Re-(T)-SPHFs in the PKE. The first four rows describe the sets on which subset membership problems are defined, hash value spaces, tag spaces and hashing key spaces respectively. The rest of rows indicate certain algorithms in these Re-(T)-SPHFs are required to be identical.

SPHF	I_0	I_1	I_2	$\overline{I}_0$	I_3	I_4
SMP	$(\mathcal{X}, \mathcal{L})$			$(\overline{\mathcal{X}}, \overline{\mathcal{L}})$		
Hash Value	Π_0	Π_1		$\overline{\Pi}_0$	Π_3	
Tag	–	$\overline{\Pi}_0 \times \mathbb{Z}$		–	$\overline{\Pi}_0$	
Hashing Key	$\mathcal{K}_0$	$\mathcal{K}_1$		$\overline{\mathcal{K}}_0$	$\mathcal{K}_3$	
Alg. ϕ	$I_0.\phi$	$I_1.\phi$		$\overline{I}_0.\phi$	$I_3.\phi$	
Alg. RandX	I_0.RandX			$\overline{I}_0$.RandX		
Alg. RandT	–	I_1.RandT		–	I_3.RandT	

Descriptions of underlying SPHFs. We firstly describe the details of all the building blocks, i.e., the underlying Re-(T)-SPHFs, in Table 2.

For the Rand-RCCA security of the PKE, the underlying subset membership problems must be hard. Besides, we require that both I_0 and $\overline{I}_0$ are perfectly self-rerandomizable and $\mathsf{ST\text{-}Smooth}_1$; and I_1 is perfectly pairwise-rerandomizable on $\overline{\Pi}_0 \times \{s\}$ for any $s \in \mathbb{Z}$, $\mathsf{CPR\text{-}Smooth}$ and $\mathsf{PT\text{-}Smooth}_1$ with respect to I_2; and I_2 is perfectly self-rerandomizable on $\overline{\Pi}_0 \times \{s\}$ for any $s \in \mathbb{Z}$ and $\mathsf{CSR\text{-}Smooth}$; and I_3 is perfectly pairwise-rerandomizable, $\mathsf{CPR\text{-}Smooth}$ and $\mathsf{PT\text{-}Smooth}_1$ with respect to I_4; and I_4 is perfectly self-rerandomizable and $\mathsf{CSR\text{-}Smooth}$.

To ensure the consistency of rerandomization, we require that I_0 and $\overline{I}_0$ are linearly rerandomizable. Let ψ be an injection that maps Π_0 into $\mathbb{Z}$, $\mathcal{T}_1 = \overline{\Pi}_0 \times \mathbb{Z}$ and $\mathcal{T}_3 = \overline{\Pi}_0$. It is required that $I_1.\mathsf{RandT}(\tau, r_\tau) = (r_\tau \cdot u, \psi(M))$ and $I_3.\mathsf{RandT}(\tau', r_\tau) = r_\tau \cdot u$ for any $\tau = (u, \psi(M)) \in \mathcal{T}_1$, any $\tau' = u \in \mathcal{T}_3$ and any $r_\tau \in \overline{\Pi}_0$. In algorithms Maul and $\mathsf{MRerand}$, $1_{\overline{\mathcal{R}}_x}$ and $1_{\overline{\Pi}_0}$ denote the identity elements in groups $\overline{\mathcal{R}}_x$ and $\overline{\Pi}_0$ respectively.

Correctness. Below we analyze the correctness of the `MPKE` and then the `PKE`.

Theorem 1. *For any key pair* $(\mathsf{mpk}, \mathsf{msk})$*, any randomness* $r_\tau \in \overline{\Pi}_0$*, any ciphertext* ϱ *and* $\varrho' = \mathsf{MRerand}(\mathsf{Maul}(\varrho, r_\tau))$ *in the scheme* `MPKE`*, we have*

$$\mathsf{MDec}(\mathsf{msk}, \varrho') = \begin{cases} r_\tau \cdot \mathsf{MDec}(\mathsf{msk}, \varrho), & \mathsf{MDec}(\mathsf{msk}, \varrho) \neq \perp \\ \perp, & \mathsf{MDec}(\mathsf{msk}, \varrho) = \perp \end{cases}.$$

Proof. Let $\varrho = (x_3, e_3, \widehat{\pi}_3, x_4, \pi_4, \widetilde{\pi}_4)$, $\mathsf{msk} = (\overline{\mathsf{sk}}_0, \mathsf{sk}_3, \mathsf{sk}_4)$ and $u = \mathsf{MDec}(\mathsf{msk}, \varrho)$. If $u \neq \perp$, then $e_3 \cdot u^{-1} = \overline{I}_0.\mathsf{Priv}(\overline{\mathsf{sk}}_0, x_3)$ holds and validity checking on ϱ passes. Let $\varrho' = (x_3', e_3', \widehat{\pi}_3', x_4', \pi_4', \widetilde{\pi}_4') = \mathsf{MRerand}(\mathsf{Maul}(\varrho, r_\tau))$. By the requirement on $I_3.\mathsf{RandT}$, the linear rerandomizability of $\overline{I}_0$ and the consistency of rerandomization in $\overline{I}_0$, I_3 and I_4, let $u' = r_\tau \cdot u$, we have $e_3' \cdot u'^{-1} = \overline{I}_0.\mathsf{Priv}(\overline{\mathsf{sk}}_0, x_3')$ and the validity checking on ϱ' also passes. Thus, $\mathsf{MDec}(\mathsf{msk}, \varrho') = r_\tau \cdot u = r \cdot \mathsf{MDec}(\mathsf{msk}, \varrho)$.

If $u = \perp$, then $\pi_4 \neq \overline{I}_0.\mathsf{Priv}(\overline{\mathsf{sk}}_0, x_4)$, $\widehat{\pi}_3 \neq I_3.\mathsf{Priv}(\mathsf{sk}_3, x_3, u)$ or $\widetilde{\pi}_4 \neq I_4.\mathsf{Priv}(\mathsf{sk}_4, x_4, u)$ holds. In this case, the corresponding inequalities also hold in ciphertext ϱ', then $\mathsf{MDec}(\mathsf{msk}, \varrho') = \perp$. ■

Theorem 2. *For any public/private key pair* $(\mathsf{PK}, \mathsf{SK})$*, any ciphertext* ζ *and* $\zeta' = \mathsf{Rerand}(\zeta)$ *in the scheme* `PKE`*, we have* $\mathsf{Dec}(\mathsf{SK}, \zeta) = \mathsf{Dec}(\mathsf{SK}, \zeta')$.

Proof. Let $\zeta = (x_1, e_1, \widehat{\pi}_1, x_2, \pi_2, \widetilde{\pi}_2, \varrho)$ and $\zeta' = (x_1', e_1', \widehat{\pi}_1', x_2', \pi_2', \widetilde{\pi}_2', \varrho')$ be a rerandomized ciphertext of ζ. Let $\mathsf{SK} = (\mathsf{sk}_0, \mathsf{sk}_1, \mathsf{sk}_2, \mathsf{msk})$, $u = \mathsf{MDec}(\mathsf{msk}, \varrho)$, $M = \mathsf{Dec}(\mathsf{SK}, \zeta)$ and $\tau = (u, \psi(M))$.

If $M \neq \perp$, then $u = \mathsf{MDec}(\mathsf{msk}, \varrho) \neq \perp$, $e_1 \cdot M^{-1} = I_0.\mathsf{Priv}(\mathsf{sk}_0, x_1)$ and the validity checking on ζ passes. By the requirement on $I_1.\mathsf{RandT}$, the linear rerandomizability of I_0 and the consistency of rerandomization in I_0, I_1 and I_2, we have $e_1' \cdot M^{-1} = I_0.\mathsf{Priv}(\mathsf{sk}_0, x_1')$ and the validity checking on ϱ' passes. Thus, we have $\mathsf{Dec}(\mathsf{SK}, \zeta') = M$.

If $M = \perp$, then $u = \perp$, $\pi_2 \neq I_0.\mathsf{Priv}(\mathsf{sk}_0, x_2)$, $\widehat{\pi}_1 \neq I_1.\mathsf{Priv}(\mathsf{sk}_1, x_1, \tau)$ or $\widetilde{\pi}_2 \neq I_2.\mathsf{Priv}(\mathsf{sk}_2, x_2, \tau)$ holds. In this case, $u' = \perp$, by Theorem 1, or the corresponding inequalities hold in ζ' as well, and then $\mathsf{Dec}(\mathsf{SK}, \zeta') = \perp$. ■

5.2 Security Analysis

Noting that the scheme MPKE is a sub-scheme of PKE, below we will provide the security of PKE as the whole but will not separately give one regarding MPKE.

Theorem 3 (Perfect Rerandomization). *The scheme* PKE *is a perfectly rerandomizable encryption scheme.*

Proof. Given fixed plaintext M, key pair $(\mathsf{PK}, \mathsf{SK})$, the distribution of the ciphertexts of M is determined by x_1, x_2, x_3, x_4 and u. Let ζ^* be a ciphertext in the support of $\mathsf{Enc}(\mathsf{PK}, M)$. Consider random variables $\zeta \leftarrow_{\$} \mathsf{Enc}(\mathsf{PK}, M)$ and $\zeta' \leftarrow_{\$} \mathsf{Rerand}(\zeta^*)$. In ciphertext ζ, u is uniformly sampled from $\overline{\Pi}_0$, while $u' = r_\tau \cdot u^*$ in ζ' is also uniformly distributed on $\overline{\Pi}_0$ as r_τ is randomly picked from $\overline{\Pi}_0$. By the perfect rerandomizability of $\overline{I}_0$, I_3 and I_4, the distribution of ϱ and ϱ' is identical. Since I_0 is perfectly self-rerandomizable, the distribution of (x_1, e_1) (resp. (x_2, π_2)) in ζ is identical to that of (x'_1, e'_1) (resp. (x'_2, π'_2)) in ζ'. The distributions of $(x_1, \widehat{\pi}_1)$ and $(x'_1, \widehat{\pi}'_1)$ are identical by the perfect pairwise-rerandomizability of I_1. Similarly, the distribution of $(x_2, \widetilde{\pi}_2)$ is the same as that of $(x'_2, \widetilde{\pi}'_2)$ by the perfect self-rerandomizability of I_2. The 1-smoothness of all the Re-(T)-SPHFs guarantees that any (possibly unbounded) adversary is unable to generate a malicious ciphertext that is decryptable. Put it all together, the theorem follows. ■

Theorem 4 (RCCA Security). *For any $(\mathcal{X}, \mathcal{L})$ and $(\overline{\mathcal{X}}, \overline{\mathcal{L}})$ where subset membership problems are hard, the proposed* PKE *in Fig. 3 is RCCA-secure.*

Proof. We prove the RCCA security of the scheme PKE by constructing a sequence of games G_0-G_3 and demonstrating the indistinguishability between them.

Game G_0: This is the $\mathsf{IND\text{-}RCCA}$ game. Specifically, challenger generates key pair $(\mathsf{PK}, \mathsf{SK})$ via KGen, and sends PK to adversary $\mathcal{A}$. After querying decryption oracle $\mathcal{DO}_{\mathsf{SK}}$, $\mathcal{A}$ chooses two plaintexts M_0, M_1. Then, challenger randomly picks $b \in \{0, 1\}$ and sends $\zeta^* \leftarrow_{\$} \mathsf{Enc}(\mathsf{PK}, M_b)$ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs b' after querying guarded decryption oracle $\mathcal{GDO}_{\mathsf{SK}}^{M_0, M_1}$.

Let S_i denote the event that $b = b'$ in game G_i, we have $\mathsf{Adv}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{IND\text{-}RCCA}}(n) = |\Pr[S_0] - 1/2|$. Let the challenge ciphertext be $\zeta^* = (x_1^*, e_1^*, \widehat{\pi}_1^*, x_2^*, \pi_2^*, \widetilde{\pi}_2^*, \varrho^*)$ and $\varrho^* = (x_3^*, e_3^*, \widehat{\pi}_3^*, x_4^*, \pi_4^*, \widetilde{\pi}_4^*)$. Below we describe the modifications in G_1-G_3.

Game G_1: This game is the same as G_0 except that challenge ciphertext ζ^* is generated by using secret key. Specifically, for the challenge ciphertext ζ^*, all the hash values are computed using hashing key. By the correctness of Re-(T)-SPHFs, same values would be computed in G_0. The differences between G_0 and G_1 are only syntactical.

We call a ciphertext ζ bad if it is invalid (i.e., $\mathsf{Dec}(\mathsf{SK}, \zeta) = \perp$) or at least one of its elements is non-language (i.e., $x_1 \in \mathcal{X} \backslash \mathcal{L}$, $x_2 \in \mathcal{X} \backslash \mathcal{L}$, $x_3 \in \overline{\mathcal{X}} \backslash \overline{\mathcal{L}}$ or $x_4 \in \overline{\mathcal{X}} \backslash \overline{\mathcal{L}}$) unless it is a rerandomization of the challenge ciphertext. ■

Lemma 1. *In game G_1, the decryption oracle rejects all the bad ciphertexts except with negligible probability.*

```
$\mathsf{AltEnc}(\mathsf{SK}, M \in \mathit{\Pi}_0)$
    $x_1, x_2 \leftarrow_\$ \mathcal{X} \backslash \mathcal{L}$
    $u \leftarrow_\$ \overline{\mathit{\Pi}}_0;\ \tau := (u, \psi(M))$
    $e_1 := I_0.\mathsf{Priv}(\mathsf{sk}_0, x_1) \cdot M$
    $\widehat{\pi}_1 := I_1.\mathsf{Priv}(\mathsf{sk}_1, x_1, \tau)$
    $\pi_2 := I_0.\mathsf{Priv}(\mathsf{sk}_0, x_2)$
    $\widetilde{\pi}_2 := I_2.\mathsf{Priv}(\mathsf{sk}_2, x_2, \tau)$
    $\varrho \leftarrow_\$ \mathsf{AltMEnc}(\mathsf{msk}, u)$
    return $\zeta := (x_1, e_1, \widehat{\pi}_1, x_2, \pi_2, \widetilde{\pi}_2, \varrho)$
```

```
$\mathsf{AltMEnc}(\mathsf{msk}, u \in \overline{\mathit{\Pi}}_0)$
    $x_3, x_4 \leftarrow_\$ \overline{\mathcal{X}} \backslash \overline{\mathcal{L}}$
    $e_3 := \overline{I}_0.\mathsf{Priv}(\overline{\mathsf{sk}}_0, x_3) \cdot u$
    $\widehat{\pi}_3 := I_3.\mathsf{Priv}(\mathsf{sk}_3, x_3, u)$
    $\pi_4 := \overline{I}_0.\mathsf{Priv}(\overline{\mathsf{sk}}_0, x_4)$
    $\widetilde{\pi}_4 := I_4.\mathsf{Priv}(\mathsf{sk}_4, x_4, u)$
    return $\varrho := (x_3, e_3, \widehat{\pi}_3, x_4, \pi_4, \widetilde{\pi}_4)$
```

Fig. 5. Modified encryption algorithms AltEnc and $\mathsf{AltMEnc}$

Proof. First, querying a valid ciphertext ζ with $x_1, x_2 \in \mathcal{L}$ and $x_3, x_4 \in \overline{\mathcal{L}}$ does not reveal more information about the secret key SK.

Consider the first bad ciphertext ζ submitted to the decryption oracle. If at least one of its elements is non-language, by the 1-smoothness of $I_0, I_1, I_2, \overline{I}_0, I_3$ and I_4, the corresponding hash value is uniformly distributed over appropriate domain and the probability that ζ is valid is negligible. If ζ is invalid, the decryption oracle rejects it with probability 1. Meanwhile, the rejection from decryption oracle rules out a negligible faction of secret keys, and the correct secret key is still uniformly distributed among the rest of secret keys in adversary's view. Since the number of query is polynomial, the probability that adversary generates a "valid" bad ciphertext is negligible. ■

Game G_2: This game is the same as G_1 except that challenge ciphertext ζ^* is generated with $x_3^*, x_4^* \leftarrow_\$ \overline{\mathcal{X}} \backslash \overline{\mathcal{L}}$ and $x_1^*, x_2^* \leftarrow_\$ \mathcal{X} \backslash \mathcal{L}$. That is, ζ^* is generated using AltEnc in Fig. 5. By the hardness of SMP on $(\overline{\mathcal{X}}, \overline{\mathcal{L}})$ and $(\mathcal{X}, \mathcal{L})$, games G_1 and G_2 are of computational indistinguishability. Here we omit the details of reduction.

Lemma 2. *In game* G_2*, if the decryption oracles reject all the bad ciphertexts except with negligible probability, then the challenge ciphertext* ζ^* *is distributed independently of plaintext* M_b *and mask* u^**, even given public key* PK.

Proof. Since $x_1^*, x_2^* \in \mathcal{X} \backslash \mathcal{L}$, by the pairwise-twin 1-smoothness of I_1 with respect to I_2, $\widehat{\pi}_1^*$ and $\widetilde{\pi}_2^*$ are uniformly distributed over appropriate domains given $\mathsf{pk}_1(\mathsf{pk}_2)$. Similarly, $\widehat{\pi}_3^*$ and $\widetilde{\pi}_4^*$ are uniformly distributed over appropriate domains given $\mathsf{pk}_3(\mathsf{pk}_4)$ by the pairwise-twin 1-smoothness of I_3 with respect to I_4. By the self-twin 1-smoothness of I_0, both π_1^* and π_2^* are statistically close to random. Similarly, π_3^* and π_4^* are statistically close to random by the self-twin 1-smoothness of $\overline{I}_0$. ■

By Lemma 1, in Phase 1, the decryption oracle rejects all the bad ciphertexts except with negligible probability. Thus, before Phase 2, u^* is uniformly distributed in adversary's view. This is crucial to the proof of Lemma 4.

Game G_3: This game is the same as G_2 except that both decryption oracle $\mathcal{DO}_{\mathsf{SK}}$ (in Phase 1) and guarded decryption oracle $\mathcal{GDO}_{\mathsf{SK}}^{M_0,M_1}$ (in Phase 2) return the output of alternate decryption algorithm AltDec (described below) that uses public keys and challenge ciphertext to decrypt ciphertexts instead of secret keys. We now prove that G_2 and G_3 are statistically indistinguishable. *Note that in this case* AltDec *is allowed to run in unbounded time.* In fact, this is essentially why AltDec is able to answer any decryption query using the public key and the challenge ciphertext only.

For any decryption query $\zeta = (x_1, e_1, \widehat{\pi}_1, x_2, \pi_2, \widetilde{\pi}_2, \varrho)$, we first describe the sub-algorithm $\mathsf{AltMDec}$ which is called by AltDec to decrypt $\varrho = (x_3, e_3, \widehat{\pi}_3, x_4, \pi_4, \widetilde{\pi}_4)$. Let $\varrho^* = (x_3^*, e_3^*, \widehat{\pi}_3^*, x_4^*, \pi_4^*, \widetilde{\pi}_4^*)$ denote the encryption of u^* in challenge ciphertext ζ^*. To decrypt ϱ, $\mathsf{AltMDec}$ performs as below.

(i) Check that $x_3, x_4 \in \overline{\mathcal{L}}$. If not, go to (ii). Otherwise, let w_3, w_4 be the witnesses of x_3, x_4, check that $\pi_4 = \overline{I}_0.\mathsf{Pub}(\overline{\mathsf{pk}}_0, x_4, w_4)$ holds. If not, output $\perp$. Otherwise, compute $u = e_3 \cdot (\overline{I}_0.\mathsf{Pub}(\overline{\mathsf{pk}}_0, x_3, w_3))^{-1}$, and check that $\widehat{\pi}_3 = I_3.\mathsf{Pub}(\mathsf{pk}_3, x_3, w_3, u)$ and $\widetilde{\pi}_4 = I_4.\mathsf{Pub}(\mathsf{pk}_4, x_4, w_4, u)$ hold. If not, output $\perp$. Otherwise, output $(\sigma = u, s = 0)$.

(ii) If $\mathsf{AltMDec}$ is called in Phase 1, output $\perp$. Otherwise, check that there exist $r_3, r_4 \in \overline{\mathcal{R}}_x$ and $r_\tau \in \overline{\Pi}_0$ such that $\varrho = \mathsf{MRerand}(\mathsf{Maul}(\varrho^*, r_\tau))$. If r_3, r_4 or r_τ does not exist, output $\perp$. Otherwise, output $(\sigma = r_\tau, s = 1)$.

The correctness of $\mathsf{AltMDec}$ is proved in Lemma 3.

Lemma 3. *Let* $(\mathsf{mpk}, \mathsf{msk})$ *be a public/secret key pair of the* `MPKE` *and* ϱ^* *be a ciphertext generated using* $\mathsf{AltMEnc}$. *Let* $(\sigma, s) = \mathsf{AltMDec}(\mathsf{mpk}, \varrho^*, \varrho)$, *if* $(\sigma, s) \neq \perp$, *then* $\mathsf{MDec}(\mathsf{msk}, \varrho) = \sigma \cdot \mathsf{MDec}(\mathsf{msk}, \varrho^*)^s$.

Proof. If $s = 0$, ϱ is a fresh encryption of u with $x_3, x_4 \in \overline{\mathcal{L}}$. By the correctness of $\overline{I}_0$, I_3 and I_4, MDec also decrypts ϱ into u. If $s = 1$, ϱ is a derivative ciphertext of ϱ^*. Although ϱ and ϱ^* both are not generated by MEnc, one can verify that $\mathsf{MDec}(\mathsf{msk}, \varrho) = r_\tau \cdot u^* = r_\tau \cdot \mathsf{MDec}(\mathsf{msk}, \varrho^*)$. ■

Now we are ready to describe AltDec. Let $\zeta^* = (x_1^*, e_1^*, \widehat{\pi}_1^*, x_2^*, \pi_2^*, \widetilde{\pi}_2^*, \varrho^*)$ be the challenge ciphertext. AltDec then decrypts $\zeta = (x_1, e_1, \widehat{\pi}_1, x_2, \pi_2, \widetilde{\pi}_2, \varrho)$ with PK and ζ^* as below.

(i) Compute $(\sigma, s) = \mathsf{AltMDec}(\mathsf{mpk}, \varrho^*, \varrho)$. If $\mathsf{AltMDec}$ returns $\perp$, then also return $\perp$.

(ii) If $s = 0$, then $\sigma = u$. Check that there exist message M and witnesses w_1, w_2 such that $x_1, x_2 \in \mathcal{L}$ and

$$\begin{array}{ll} e_1 = I_0.\mathsf{Pub}(\mathsf{pk}_0, x_1, w_1) \cdot M & \pi_2 = I_0.\mathsf{Pub}(\mathsf{pk}_0, x_2, w_2) \\ \widehat{\pi}_1 = I_1.\mathsf{Pub}(\mathsf{pk}_1, x_1, w_1, \tau) & \widetilde{\pi}_2 = I_2.\mathsf{Pub}(\mathsf{pk}_2, x_2, w_2, \tau), \end{array}$$

where $\tau = (u, \psi(M))$. If not, output $\perp$. If $M \notin \{M_0, M_1\}$, output M; otherwise, output `replay`.

(iii) If $s = 1$, then $\sigma = r_\tau$. Check that there exist randomness $r_1, r_2 \in \mathcal{R}_x$ such that following equalities hold.

$$\begin{array}{ll} x_1 = I_0.\mathsf{RandX}(x_1^*, x_2^*, r_1) & x_2 = I_0.\mathsf{RandX}(x_2^*, x_2^*, r_2) \\ e_1 = I_0.\mathsf{RandH}(e_1^*, \pi_2^*, r_1) & \pi_2 = I_0.\mathsf{RandH}(\pi_2^*, \pi_2^*, r_2) \\ \widehat{\pi}_1 = I_1.\mathsf{RandH}(\widehat{\pi}_1^*, \widetilde{\pi}_2^*, r_1, r_\tau) & \widetilde{\pi}_2 = I_2.\mathsf{RandH}(\widetilde{\pi}_2^*, \widetilde{\pi}_2^*, r_2, r_\tau). \end{array}$$

If not, output $\perp$. Otherwise, output `replay`.

Lemma 4. *The output of $\mathcal{DO}_{\mathsf{SK}}$(resp. $\mathcal{GDO}_{\mathsf{SK}}^{M_0,M_1}$) in G_3 agrees with the output of $\mathcal{DO}_{\mathsf{SK}}$(resp. $\mathcal{GDO}_{\mathsf{SK}}^{M_0,M_1}$) in G_2 with overwhelming probability.*

Proof. In the cases where $\mathcal{DO}_{\mathsf{SK}}$(resp. $\mathcal{GDO}_{\mathsf{SK}}^{M_0,M_1}$) in G_3 outputs M, $\mathcal{DO}_{\mathsf{SK}}$(resp. $\mathcal{GDO}_{\mathsf{SK}}^{M_0,M_1}$) in G_2 also outputs M by Lemma 3 and the correctness of decryption. Similarly, when $\mathcal{GDO}_{\mathsf{SK}}^{M_0,M_1}$ in G_3 outputs `replay`, $\mathcal{GDO}_{\mathsf{SK}}^{M_0,M_1}$ in G_2 also outputs `replay` by Lemma 3 and correctness of decryption and rerandomization.

We now prove that when $\mathcal{DO}_{\mathsf{SK}}$(resp. $\mathcal{GDO}_{\mathsf{SK}}^{M_0,M_1}$) in G_3 outputs $\perp$ on query ζ, $\mathcal{DO}_{\mathsf{SK}}$(resp. $\mathcal{GDO}_{\mathsf{SK}}^{M_0,M_1}$) in G_2 also would output $\perp$ with overwhelming probability. That is, when AltDec outputs $\perp$, Dec also would output $\perp$ with overwhelming probability. Let $\zeta^* = (x_1^*, e_1^*, \widehat{\pi}_1^*, x_2^*, \pi_2^*, \widetilde{\pi}_2^*, \varrho^*)$ denote the challenge ciphertext where $\varrho^* = (x_3^*, e_3^*, \widehat{\pi}_3^*, x_4^*, \pi_4^*, \widetilde{\pi}_4^*)$ and $\zeta = (x_1, e_1, \widehat{\pi}_1, x_2, \pi_2, \widetilde{\pi}_2, \varrho)$ denote the decryption query input where $\varrho = (x_3, e_3, \widehat{\pi}_3, x_4, \pi_4, \widetilde{\pi}_4)$.

Case 1. If AltDec outputs $\perp$ due to $\mathsf{AltMDec}$ returning $\perp$, there are following possible sub-cases.

- In Phase 1, $x_3 \notin \overline{\mathcal{L}}$ or $x_4 \notin \overline{\mathcal{L}}$. By the 1-smoothness of $\overline{I}_0$, $\pi_3 = e_3 \cdot u^{-1}$ or π_4 is statistically close to random, and thus ζ will be rejected by Dec with overwhelming probability.
- In Phase 2, $r_3, r_4 \in \overline{\mathcal{R}}_x$ or $r_\tau \in \overline{\varPi}_0$ does not exist for $\varrho = \mathsf{MRerand}(\mathsf{Maul}(\varrho^*, r_\tau))$ with x_3 or $x_4 \notin \overline{\mathcal{L}}$. If r_τ does not exist, by the $\mathsf{CPR\text{-}Smooth}$ of I_3 or $\mathsf{CSR\text{-}Smooth}$ of I_4, $\widehat{\pi}_3$ or $\widetilde{\pi}_4$ is close to random, as x_3 or $x_4 \notin \overline{\mathcal{L}}$. If r_3 does not exist and $x_3 \notin \overline{\mathcal{L}}$, $\widehat{\pi}_3$ is close to random by the $\mathsf{CPR\text{-}Smooth}$ of I_3. If r_4 does not exist and $x_4 \notin \overline{\mathcal{L}}$, $\widetilde{\pi}_4$ is close to random by the $\mathsf{CSR\text{-}Smooth}$ of I_4. If r_3 does not exist and $x_3 \in \overline{\mathcal{L}}$, then $x_4 \notin \overline{\mathcal{L}}$. In this case, we assume that there exists r_4 such that $x_4 = \overline{I}_0.\mathsf{RandX}(x_4^*, x_4^*, r_4)$. Since u^* is uniformly sampled from $\overline{\varPi}_0$ at random, the underlying u of $\widetilde{\pi}_4$ equals to $r_\tau \cdot u^*$ which is uniformly distributed over $\overline{\varPi}_0$. Then, $\widehat{\pi}_3$ is close to random, as u is uniformly distributed and $\widehat{\pi}_3$ is independent of $\widehat{\pi}_3^*$. Similarly, we can prove that $\widetilde{\pi}_4$ is close to random when r_4 does not exist, r_3 exists, $x_4 \in \overline{\mathcal{L}}$ and $x_3 \notin \overline{\mathcal{L}}$.
- In both Phase 1 and 2, $\pi_4 \neq \overline{I}_0.\mathsf{Pub}(\overline{\mathsf{pk}}_0, x_4, w_4)$, $\widehat{\pi}_3 \neq I_3.\mathsf{Pub}(\mathsf{pk}_3, x_3, w_3, u)$ or $\widetilde{\pi}_4 \neq I_4.\mathsf{Pub}(\mathsf{pk}_4, x_4, w_4, u)$ holds. Obviously, MDec would reject ϱ and Dec would reject ζ.

Case 2. Suppose that $(\sigma, s) = \mathsf{AltMDec}(\mathsf{mpk}, \varrho^*, \varrho)$ and $(\sigma, s) \neq \perp$. There are following sub-cases where AltDec outputs $\perp$.

- In Phase 1, $(\sigma, s) = (u, 0)$ and x_1 or $x_2 \notin \mathcal{L}$. By the 1-smoothness of I_0, I_1 and I_2, π_2, $\widehat{\pi}_1$ or $\widetilde{\pi}_2$ is statistically close to random. Suppose $x_1, x_2 \in \mathcal{L}$ and $\mathsf{pk}_0, \mathsf{pk}_1(\mathsf{pk}_2)$ are fixed. If any equation in decryption rule (ii) of AltDec does not hold for any $M \in \Pi_0$, ζ would be rejected due to the validity checking.
- In Phase 2, $(\sigma, s) = (u, 0)$ and x_1 or $x_2 \notin \mathcal{L}$. If $x_1 = I_0.\mathsf{RandX}(x_1^*, x_2^*, r_1)$ or $x_2 = I_0.\mathsf{RandX}(x_2^*, x_2^*, r_2)$, the underlying tag $\tau = (u, \psi(M))$ of $\widehat{\pi}_1$ or $\widetilde{\pi}_2$ which is derived from $\widehat{\pi}_1^*$ and $\widetilde{\pi}_2^*$ via $I_1.\mathsf{RandH}$ or $I_2.\mathsf{RandH}$ would be related to $\tau^* = (u^*, \psi(M^*))$ where u^* is uniformly distributed over $\overline{\Pi}_0$. However, $s = 0$ indicates that the value of u is fixed and $u = \sigma$. Thus, the validity checking on ζ would fail. Otherwise, $x_1 \neq I_0.\mathsf{RandX}(x_1^*, x_2^*, r_1)$ and $x_2 \neq I_0.\mathsf{RandX}(x_2^*, x_2^*, r_2)$. Given fixed pk_1, $\widehat{\pi}_1^*$ and $\widetilde{\pi}_2^*$, the value of $\widehat{\pi}_1$ is statistically close to random as I_1 is $\mathsf{CPR\text{-}Smooth}$.
- In Phase 1 and 2, $(\sigma, s) = (u, 0)$ and $x_1, x_2 \in \mathcal{L}$. If equations in rule (ii) of AltDec do not hold simultaneously for any $M \in \Pi_0$, the validity checking on ζ in Dec would fail.
- In Phase 2, $(\sigma, s) = (r_\tau, 1)$, and there do not exist $r_1, r_2 \in \mathcal{R}_x$ such that equations in decryption rule (iii) of AltDec hold at the same time. If $x_1 \neq I_0.\mathsf{RandX}(x_1^*, x_2^*, r_1)$ for any $r_1 \in \mathcal{R}_x$ or $\tau \neq I_0.\mathsf{RandT}(\tau^*, r_\tau)$, due to the fact that I_1 is $\mathsf{CPR\text{-}Smooth}$, $\widehat{\pi}_1$ is statistically indistinguishable from random hash value given fixed pk_1, $\widehat{\pi}_1^*$ and $\widetilde{\pi}_2^*$. Similarly, if $x_2 \neq I_0.\mathsf{RandX}(x_2^*, x_2^*, r_2)$ for any $r_2 \in \mathcal{R}_x$ or $\tau \neq I_0.\mathsf{RandT}(\tau^*, r_\tau)$, due to the fact that I_2 is $\mathsf{CSR\text{-}Smooth}$, $\widetilde{\pi}_2$ is statistically close to random hash value given fixed pk_2 and $\widetilde{\pi}_2^*$. Suppose that $x_1 = I_0.\mathsf{RandX}(x_1^*, x_2^*, r_1)$, $x_2 = I_0.\mathsf{RandX}(x_1^*, x_2^*, r_2)$ and $\tau = I_0.\mathsf{RandT}(\tau^*, r_\tau)$. If equations in rule (iii) of AltDec do not hold simultaneously, the validity checking on ζ in Dec would fail.

In conclusion, The output of $\mathcal{DO}_{\mathsf{SK}}$(resp. $\mathcal{GDO}_{\mathsf{SK}}^{M_0,M_1}$) in G_3 is the same as that in G_2 in every case with overwhelming probability. ∎

Lemma 5. $\Pr[S_3] = 1/2$.

Proof. Note that $\mathsf{AltMDec}$ and AltDec do not use secret key to perform decryption. The decryption oracle responses in game G_3 do not provide extra information about secret key besides public key and challenge ciphertext ζ^* generated using AltEnc. Lemma 2 shows that ζ^* is distributed independently of bit b, from which the lemma follows. ∎

Putting it all together, the theorem follows. ∎

Theorem 5 (RCCA Receiver-Anonymity). *For any* $(\mathcal{X}, \mathcal{L})$ *and* $(\overline{\mathcal{X}}, \overline{\mathcal{L}})$ *where subset membership problems are hard, the proposed* $\mathtt{PKE}$ *in Fig. 3 is RCCA receiver-anonymous.*

Proof. We prove the receiver-anonymity of $\mathtt{PKE}$ by constructing a sequence of games G_0-G_3 and demonstrating the indistinguishability between them.

Game G_0: This is the $\mathsf{ANON\text{-}RCCA}$ game. Specifically, challenger generates two key pairs $(\mathsf{PK}_0, \mathsf{SK}_0)$ and $(\mathsf{PK}_1, \mathsf{SK}_1)$ via KGen, and sends $(\mathsf{PK}_0, \mathsf{PK}_1)$ to adversary $\mathcal{A}$. After querying decryption oracle $\mathcal{DO}_{\mathsf{SK}_0,\mathsf{SK}_1}$, $\mathcal{A}$ chooses a plaintext M. Then, challenger randomly picks $b \in \{0,1\}$ and sends $\zeta^* \leftarrow_\$ \mathsf{Enc}(\mathsf{PK}_b, M)$ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs b' after querying guarded decryption oracle $\mathcal{GDO}^M_{\mathsf{SK}_0,\mathsf{SK}_1}$.

Let S_i denote the event that $b = b'$ in game G_i, we have $\mathsf{Adv}^{\mathsf{ANON\text{-}RCCA}}_{\mathcal{A},\mathsf{PKE}}(n) = |\Pr[S_0] - 1/2|$.

Game G_1: This game is the same as G_0 except that challenge ciphertext ζ^* is generated by using secret key SK_b. According to the analysis in Theorem 4, game G_1 is identical to G_0 by the correctness of SPHFs.

Game G_2: This game is the same as G_1 except that challenge ciphertext ζ^* is generated with $x_3^*, x_4^* \leftarrow_\$ \overline{\mathcal{X}} \backslash \overline{\mathcal{L}}$ and $x_1^*, x_2^* \leftarrow_\$ \mathcal{X} \backslash \mathcal{L}$. That is, ζ^* is generated using AltEnc in Fig. 5. By the hardness of SMP on $(\overline{\mathcal{X}}, \overline{\mathcal{L}})$ and $(\mathcal{X}, \mathcal{L})$, games G_1 and G_2 are of computational indistinguishability.

Game G_3: This game is the same as G_2 except that both decryption oracle $\mathcal{DO}_{\mathsf{SK}_0,\mathsf{SK}_1}$ (in Phase 1) and guarded decryption oracle $\mathcal{GDO}^M_{\mathsf{SK}_0,\mathsf{SK}_1}$ (in Phase 2) work as follows. First, it runs alternative decryption algorithm AltDec^*, which is the same as AltDec in Theorem 4 except that it outputs `replay` when decryption result equals to M, with PK_0 and PK_1 respectively. If AltDec^* outputs `replay`, it returns `replay`, otherwise, it returns the results of running AltDec^*. By Lemma 4, the output of $\mathcal{DO}_{\mathsf{SK}_0,\mathsf{SK}_1}$ ($\mathcal{GDO}^M_{\mathsf{SK}_0,\mathsf{SK}_1}$) in G_3 agrees with the output of $\mathcal{DO}_{\mathsf{SK}_0,\mathsf{SK}_1}$ ($\mathcal{GDO}^M_{\mathsf{SK}_0,\mathsf{SK}_1}$) in G_2 with overwhelming probability. Thus, games G_2 and G_3 are statistically indistinguishable.

Note that AltDec^* does not use secret key to perform decryption. The decryption oracle responses in game G_3 do not provide extra information about secret key SK_b besides public keys $\mathsf{PK}_0, \mathsf{PK}_1$ and challenge ciphertext ζ^* generated using AltEnc. By Lemma 2, ζ^* is distributed independently of PK_b. Thus, we have $\Pr[S_3] = 1/2$, from which the theorem follows. ■

6 Instantiations

In this section, we show how to instantiate our framework from the k-Lin assumption. More generally, it could be constructed from graded rings [3] and we provide the details in the full version [29].

6.1 Regular SPHF from k-Lin Assumption

Let $\mathbb{G}$ be a cyclic group with prime order p. The k-Lin assumption says that $[\mathbf{r}^\top \mathbf{g}_{k+1}]$ is pseudorandom given $[\mathbf{g}^\top]$, $[g_{k+1}]$, $[\mathbf{r}^\top \mathbf{G}]$ where $\mathbf{r}, \mathbf{g} \leftarrow_\$ \mathbb{Z}_p^k$, $g_{k+1} \leftarrow_\$ \mathbb{Z}_p$ and $\mathbf{G} = \mathsf{diag}(\mathbf{g}^\top) \in \mathbb{Z}_p^{k \times k}$, $\mathbf{g}_{k+1} = (g_{k+1}, \cdots, g_{k+1})^\top \in \mathbb{Z}_p^k$.

Let element set $\mathcal{X} = \{[\mathbf{x}^\top] \,|\, \mathbf{x} \in \mathbb{Z}_p^{k+1}\}$ and $\mathcal{L} = \{[\mathbf{w}^\top \mathbf{P}] \,|\, \mathbf{w} \in \mathbb{Z}_p^k\}$ where $\mathbf{P} = (\mathbf{G}\ \mathbf{g}_{k+1}) \in \mathbb{Z}_p^{k \times (k+1)}$. Below is a regular SPHF from k-Lin assumption.

- $\mathsf{Setup}(1^n)$. Let $\mathcal{K} = \mathbb{Z}_p^{k+1}$, $\Pi = \mathbb{G}$ and $\mathcal{T} = \emptyset$. Since the tag space is empty, $H_{(\cdot)} : \mathcal{X} \to \mathbb{G}$ is an efficient hash function family indexed by $\mathsf{sk} \in \mathbb{Z}_p^{k+1}$.
- $\phi(\mathsf{sk})$. For $\mathsf{sk} = \mathbf{a} \in \mathbb{Z}_p^{k+1}$, outputs $\mathsf{pk} = [\mathbf{Pa}] \in \mathbb{G}^k$.

- $\mathsf{Priv}(\mathsf{sk}, x)$. For $\mathsf{sk} = \mathbf{a} \in \mathbb{Z}_p^{k+1}$ and $x = \left[\mathbf{x}^\top\right] \in \mathcal{X}$, outputs $\pi = \left[\mathbf{x}^\top \mathbf{a}\right] \in \mathbb{G}$.
- $\mathsf{Pub}(\mathsf{pk}, x, w)$. For $\mathsf{pk} = [\mathbf{Pa}] \in \mathbb{G}^k$ and $x = \left[\mathbf{w}^\top \mathbf{P}\right] \in \mathcal{L}$ with witness $\mathbf{w} \in \mathbb{Z}_p^k$, outputs $\pi = \left[\mathbf{w}^\top (\mathbf{Pa})\right] \in \mathbb{G}$.

Since $\left[\mathbf{w}^\top (\mathbf{Pa})\right] = \left[(\mathbf{w}^\top \mathbf{P})\mathbf{a}\right]$, the correctness of SPHF holds. For any $x \notin \mathcal{L}$ and $\mathsf{pk} = [\mathbf{Pa}]$, vector $\mathbf{x}^\top$ is not in the linear span of $\mathbf{P}$, then hash value $H_{\mathsf{sk}}(x) = \left[\mathbf{x}^\top \mathbf{a}\right]$ is independent from $\mathsf{pk} = [\mathbf{Pa}]$. This guarantees the 1-smoothness.

6.2 Instantiating the Underlying Re-(T)-SPHFs of Our Framework

(1) **Construction of I_0 and $\overline{I}_0$.** The algorithms ($I_0.\mathsf{Setup}, I_0.\phi, I_0.\mathsf{Priv}, I_0.\mathsf{Pub}$) are the same as those of regular SPHF from k-LIN assumption, and thus the 1-smoothness of I_0 is obvious. Below we provide the remaining algorithms, i.e., $I_0.\mathsf{RandX}$ and $I_0.\mathsf{RandH}$.

- $I_0.\mathsf{RandX}(x, x', r_x)$. For $x = \left[\mathbf{x}^\top\right], x' = \left[\mathbf{x}'^\top\right] \in \mathcal{X}$ and $r_x \in \mathbb{Z}_p$, outputs $x^* = \left[\mathbf{x}^\top + r_x \mathbf{x}'^\top\right]$.
- $I_0.\mathsf{RandH}(\pi, \pi', r_x)$. For $\pi = \left[\mathbf{x}^\top \mathbf{a}\right], \pi' = \left[\mathbf{x}'^\top \mathbf{a}\right] \in \mathbb{G}$ and $r_x \in \mathbb{Z}_p$, outputs $\pi^* = \pi \cdot (\pi')^{r_x} = \left[\mathbf{x}^\top \mathbf{a} + r_x \mathbf{x}'^\top \mathbf{a}\right]$.

Since $\pi^* = \left[(\mathbf{x}^\top + r_x \mathbf{x}'^\top)\mathbf{a}\right] = I_0.\mathsf{Priv}(\mathsf{sk}, I_0.\mathsf{RandX}(x, x', r_x))$, the correctness of rerandomization holds. For any $\pi, \pi', \Delta \in \mathbb{G}$ and any $r_x \in \mathbb{Z}_p$, we have $I_0.\mathsf{RandH}(\pi \cdot \Delta, \pi', r_x) = (\pi \cdot \Delta) \cdot (\pi')^{r_x} = (\pi \cdot (\pi')^{r_x}) \cdot \Delta = I_0.\mathsf{RandH}(\pi, \pi', r_x) \cdot \Delta$ and I_0 is linearly rerandomizable. Due to lack of space, the proofs of following theorems appear in the full version [29].

Theorem 6. *I_0 is perfectly self-rerandomizable.*

Theorem 7. *I_0 is $\mathsf{ST\text{-}Smooth}_1$ when $k \geq 2$.*

The construction of $\overline{I}_0$ is exactly the same as I_0. In concrete scheme, it is associated with $\overline{\mathcal{X}}$ and NP-language $\overline{\mathcal{L}}$ that are defined over $\overline{\mathbb{G}}^{k+1}$ where $\overline{\mathbb{G}}$ is a cyclic group with prime order q and a subgroup of $\mathbb{Z}_p^*$. Specifically, $\overline{\mathcal{X}} = \left\{\left[\overline{\mathbf{x}}^\top\right] \middle| \overline{\mathbf{x}} \in \mathbb{Z}_q^{k+1}\right\}$, and $\overline{\mathcal{L}} = \left\{\left[\mathbf{w}^\top \overline{\mathbf{P}}\right] \middle| \mathbf{w} \in \mathbb{Z}_q^k\right\}$ where $\overline{\mathbf{P}} = (\overline{\mathbf{G}}\ \overline{\mathbf{g}}_{k+1}) \in \mathbb{Z}_q^{k \times (k+1)}$, $\overline{\mathbf{G}} = \mathsf{diag}(\overline{\mathbf{g}}^\top) \in \mathbb{Z}_q^{k \times k}$, $\overline{\mathbf{g}}_{k+1} = (\overline{g}_{k+1}, \cdots, \overline{g}_{k+1})^\top \in \mathbb{Z}_q^k$, $\overline{\mathbf{g}} \leftarrow_\$ \mathbb{Z}_q^k$, $\overline{g}_{k+1} \leftarrow_\$ \mathbb{Z}_q$.

(2) **Construction of I_1 and I_2.** We first describe the framework of I_1 as below.

- $I_1.\mathsf{Setup}(1^n)$. Let $\mathcal{K}_1 = (\mathbb{Z}_p^{k+1})^4$, $\Pi_1 = \mathbb{G}^2$, $\mathcal{T}_1 = \overline{\mathbb{G}} \times \mathbb{Z}_p^*$. Pick $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \leftarrow_\$ \mathbb{Z}_p^k$ with $\boldsymbol{\lambda}_1 \neq \boldsymbol{\lambda}_2$, $\mathsf{ax} = (\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2)$. $\widehat{H}_{(\cdot)} : \mathcal{X} \times \mathcal{T}_1 \to \mathbb{G}^2$ is indexed by $\mathsf{sk}_1 \in \mathcal{K}_1$ and ax.
- $I_1.\phi(\mathsf{sk}_1)$. For $\mathsf{sk}_1 = (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}) \in (\mathbb{Z}_p^{k+1})^4$, outputs
$$\mathsf{pk}_1 = ([\mathbf{Pb}], [\mathbf{Pc}], [\mathbf{Pd}], [\mathbf{Pe}]).$$
- $I_1.\mathsf{Priv}(\mathsf{sk}_1, x, \tau)$. For $\mathsf{sk}_1 = (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e})$, $x = \left[\mathbf{x}^\top\right]$ and $\tau = (\tau_0, \tau_1)$, outputs hash value $\pi = \widehat{H}_{\mathsf{sk}_1}(x, \tau) = (\pi_1, \pi_2) =$
$$\left(\left[(\mathbf{x}^\top + \boldsymbol{\lambda}_1^\top \mathbf{P})(\tau_0(\mathbf{b} + \tau_1 \mathbf{c}))\right], \left[(\mathbf{x}^\top + \boldsymbol{\lambda}_2^\top \mathbf{P})(\tau_0(\mathbf{d} + \tau_1 \mathbf{e}))\right]\right).$$

- I_1.$\mathsf{Pub}(\mathsf{pk}_1, x, w, \tau)$. For $\mathsf{pk}_1 = ([\mathbf{Pb}], [\mathbf{Pc}], [\mathbf{Pd}], [\mathbf{Pe}])$, $x = \left[\mathbf{w}^\top \mathbf{P}\right]$ with witness $\mathbf{w}$ and $\tau = (\tau_0, \tau_1)$, outputs $\pi = \widehat{H}_{\mathsf{sk}_1}(x, \tau) = (\pi_1, \pi_2) =$

$$\left(\left[(\mathbf{w}^\top + \boldsymbol{\lambda}_1^\top)(\tau_0(\mathbf{Pb} + \tau_1 \mathbf{Pc}))\right], \left[(\mathbf{w}^\top + \boldsymbol{\lambda}_2^\top)(\tau_0(\mathbf{Pd} + \tau_1 \mathbf{Pe}))\right]\right).$$

- I_1.$\mathsf{RandX}(x, x', r_x)$. For $x = \left[\mathbf{x}^\top\right]$, $x' = \left[\mathbf{x}'^\top\right]$ and $r_x \in \mathbb{Z}_p$, outputs $x^* = \left[\mathbf{x}^\top + r_x \mathbf{x}'^\top\right]$.
- I_1.$\mathsf{RandT}(\tau, r_\tau)$. For $\tau = (\tau_0, \tau_1)$ and $r_\tau \in \mathbb{Z}_p$, outputs $\tau^* = (r_\tau \cdot \tau_0, \tau_1)$.
- I_1.$\mathsf{RandH}(\pi, \pi', r_x, r_\tau)$. For $\pi = (\pi_1, \pi_2), \pi' = (\pi_1', \pi_2')$, $r_x \in \mathbb{Z}_p$ and $r_\tau \in \mathbb{Z}_p$, outputs $\pi^* = ((\pi_1 \cdot (\pi_1')^{r_x})^{r_\tau},\ (\pi_2 \cdot (\pi_2')^{r_x})^{r_\tau})$.

As for I_2, its algorithms $I_2.\phi, I_2.\mathsf{RandX}, I_2.\mathsf{RandT}$ and $I_2.\mathsf{RandH}$ are the same as $I_1.\phi$, $I_1.\mathsf{RandX}$, $I_1.\mathsf{RandT}$ and $I_1.\mathsf{RandH}$. Besides, $I_2.\mathsf{Setup}$ is the same as $I_1.\mathsf{Setup}$ except that ax is null and the hash function family is $\widetilde{H}_{(\cdot)} : \mathcal{X} \times \mathcal{T}_2 \to \mathbb{G}^2$ where $\mathcal{T}_2 = \mathcal{T}_1$. $I_2.\mathsf{Priv}$ and $I_2.\mathsf{Pub}$ are equivalent to $I_1.\mathsf{Priv}$ and $I_1.\mathsf{Pub}$ with $\boldsymbol{\lambda}_1 = \boldsymbol{\lambda}_2 = \mathbf{0}$.

- I_2.$\mathsf{Priv}(\mathsf{sk}_2, x, \tau)$. For $\mathsf{sk}_2 = (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}) \in (\mathbb{Z}_p^{k+1})^4$, $x = \left[\mathbf{x}^\top\right]$ and $\tau = (\tau_0, \tau_1)$, outputs hash value $\pi = \widetilde{H}_{\mathsf{sk}_2}(x, \tau) = (\pi_1, \pi_2) =$

$$\left(\left[\mathbf{x}^\top(\tau_0(\mathbf{b} + \tau_1 \mathbf{c}))\right], \left[\mathbf{x}^\top(\tau_0(\mathbf{d} + \tau_1 \mathbf{e}))\right]\right).$$

- I_2.$\mathsf{Pub}(\mathsf{pk}_2, x, w, \tau)$. For $\mathsf{pk}_2 = ([\mathbf{Pb}], [\mathbf{Pc}], [\mathbf{Pd}], [\mathbf{Pe}])$, $x = \left[\mathbf{w}^\top \mathbf{P}\right]$ with witness $\mathbf{w}$ and $\tau = (\tau_0, \tau_1)$, outputs $\pi = \widetilde{H}_{\mathsf{sk}_2}(x, \tau) = (\pi_1, \pi_2) =$

$$\left(\left[\mathbf{w}^\top(\tau_0(\mathbf{Pb} + \tau_1 \mathbf{Pc}))\right], \left[\mathbf{w}^\top(\tau_0(\mathbf{Pd} + \tau_1 \mathbf{Pe}))\right]\right).$$

One can verify the correctness of I_1 and I_2 easily. For any $x \notin \mathcal{L}$, any $\tau \in \mathcal{T}_1$ and $\mathsf{pk}_1 = ([\mathbf{Pb}], [\mathbf{Pc}], [\mathbf{Pd}], [\mathbf{Pe}])$, vector $\mathbf{x}^\top$ is not in the linear span of $\mathbf{P}$, then $\left(\left[(\mathbf{x}^\top + \boldsymbol{\lambda}_1^\top \mathbf{P})(\tau_0(\mathbf{b} + \tau_1 \mathbf{c}))\right], \left[(\mathbf{x}^\top + \boldsymbol{\lambda}_2^\top \mathbf{P})(\tau_0(\mathbf{d} + \tau_1 \mathbf{e}))\right]\right)$ is independent of pk_1, from which the 1-smoothness property holds for both I_1 and I_2. As for the correctness of rerandomization, we consider $\pi = \widehat{H}_{\mathsf{sk}_1}(x, \tau)$ and $\pi' = \widetilde{H}_{\mathsf{sk}_2}(x', \tau)$ as I_1 is rerandomizable with respect to I_2. For $r_x, r_\tau \in \mathbb{Z}_p$, one can verify that rerandomized hash value $\pi^* = I_1.\mathsf{RandH}(\pi, \pi', r_x, r_\tau) = I_1.\mathsf{Priv}(\mathsf{sk}_1, x^*, \tau^*)$ where $x^* = I_1.\mathsf{RandX}(x, x', r_x)$ and $\tau^* = I_1.\mathsf{RandT}(\tau, r_\tau)$. This also holds for $\pi = \widetilde{H}_{\mathsf{sk}_2}(x, \tau)$ and $\pi' = \widetilde{H}_{\mathsf{sk}_2}(x', \tau)$. The proofs of following theorems are provided in the full version [29].

Theorem 8. *Let $\mathcal{T}_1(s) = \overline{\mathbb{G}} \times \{s\} \subseteq \mathcal{T}_1$ with $s \in \mathbb{Z}_p^*$. I_1 is perfectly pairwise-rerandomizable on $\mathcal{T}_1(s)$ with respect to I_2 for any $s \in \mathbb{Z}_p^*$.*

Theorem 9. *Let $\mathcal{T}_2(s) = \overline{\mathbb{G}} \times \{s\} \subseteq \mathcal{T}_2$ with $s \in \mathbb{Z}_p^*$. I_2 is perfectly self-rerandomizable on $\mathcal{T}_2(s)$ for any $s \in \mathbb{Z}_p^*$.*

Theorem 10. *I_1 is* $\mathsf{PT\text{-}Smooth}_1$ *with respect to I_2 when $k \geq 2$.*

Theorem 11. *I_1 is* $\mathsf{CPR\text{-}Smooth}$ *with respect to I_2.*

```
$\mathsf{KGen}(1^n)$
  $\mathbf{a} \leftarrow\$ \mathbb{Z}_p^{k+1};\ \mathbf{A} := [\mathbf{Pa}];\ (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}) \leftarrow\$ (\mathbb{Z}_p^{k+1})^4$
  $(\mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}) := ([\mathbf{Pb}], [\mathbf{Pc}], [\mathbf{Pd}], [\mathbf{Pe}])$
  $(\mathsf{mpk}, \mathsf{msk}) \leftarrow\$$ $\mathsf{MKGen}(1^n)$
      $\mathsf{msk} := (\overline{\mathbf{a}}, \overline{\mathbf{b}}, \overline{\mathbf{c}}) \leftarrow\$ \mathbb{Z}_q^{k+1}$
      $\overline{\mathbf{A}} := [\overline{\mathbf{P}}\overline{\mathbf{a}}]$
      $(\overline{\mathbf{B}}, \overline{\mathbf{C}}) := ([\overline{\mathbf{P}}\overline{\mathbf{b}}], [\overline{\mathbf{P}}\overline{\mathbf{c}}])$
      $\mathsf{mpk} := (\overline{\mathbf{A}}, \overline{\mathbf{B}}, \overline{\mathbf{C}})$
      return $(\mathsf{mpk}, \mathsf{msk})$
  $\mathsf{PK} := (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathsf{mpk})$
  $\mathsf{SK} := (\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathsf{msk})$
  return $(\mathsf{PK}, \mathsf{SK})$

$\mathsf{Enc}(\mathsf{PK}, M \in \mathbb{G})$
  $[\mathbf{x}_1^\top], [\mathbf{x}_2^\top] \leftarrow\$ \mathcal{L}$ with witness $\mathbf{w}_1, \mathbf{w}_2$
  $u \leftarrow\$ \overline{\mathbb{G}};\ m := \psi(M)$
  $\varrho \leftarrow\$$ $\mathsf{MEnc}(\mathsf{mpk}, u \in \overline{\mathbb{G}})$
      $[\mathbf{x}_3^\top], [\mathbf{x}_4^\top] \leftarrow\$ \overline{\mathcal{L}}$ with witness $\mathbf{w}_3, \mathbf{w}_4$
      $e_3 := u \cdot [\mathbf{w}_3^\top \overline{\mathbf{P}}\overline{\mathbf{a}}];\ \pi_4 := [\mathbf{w}_4^\top \overline{\mathbf{P}}\overline{\mathbf{a}}]$
      $\widehat{\pi}_{31} := \left[(\mathbf{w}_3^\top + \overline{\boldsymbol{\lambda}}_1^\top) u \overline{\mathbf{P}}\overline{\mathbf{b}}\right]$
      $\widehat{\pi}_{32} := \left[(\mathbf{w}_3^\top + \overline{\boldsymbol{\lambda}}_2^\top) u \overline{\mathbf{P}}\overline{\mathbf{c}}\right]$
      $\widetilde{\pi}_{41} := \left[\mathbf{w}_4^\top u \overline{\mathbf{P}}\overline{\mathbf{b}}\right];\ \widetilde{\pi}_{42} := \left[\mathbf{w}_4^\top u \overline{\mathbf{P}}\overline{\mathbf{c}}\right]$
      $\widehat{\boldsymbol{\pi}}_3 := (\widehat{\pi}_{31}, \widehat{\pi}_{32});\ \widetilde{\boldsymbol{\pi}}_4 := (\widetilde{\pi}_{41}, \widetilde{\pi}_{42})$
      return $\varrho := ([\mathbf{x}_3^\top], e_3, \widehat{\boldsymbol{\pi}}_3, [\mathbf{x}_4^\top], \pi_4, \widetilde{\boldsymbol{\pi}}_4)$
  $e_1 := M \cdot [\mathbf{w}_1^\top \mathbf{Pa}];\ \ \pi_2 := [\mathbf{w}_2^\top \mathbf{Pa}]$
  $\widehat{\pi}_{11} := \left[(\mathbf{w}_1^\top + \boldsymbol{\lambda}_1^\top)(u(\mathbf{Pb} + m\mathbf{Pc}))\right]$
  $\widehat{\pi}_{12} := \left[(\mathbf{w}_1^\top + \boldsymbol{\lambda}_2^\top)(u(\mathbf{Pd} + m\mathbf{Pe}))\right]$
  $\widetilde{\pi}_{21} := \left[\mathbf{w}_2^\top (u(\mathbf{Pb} + m\mathbf{Pc}))\right]$
  $\widetilde{\pi}_{22} := \left[\mathbf{w}_2^\top (u(\mathbf{Pd} + m\mathbf{Pe}))\right]$
  $\widehat{\boldsymbol{\pi}}_1 := (\widehat{\pi}_{11}, \widehat{\pi}_{12});\ \ \widetilde{\boldsymbol{\pi}}_2 := (\widetilde{\pi}_{21}, \widetilde{\pi}_{22})$
  return $\zeta := ([\mathbf{x}_1^\top], e_1, \widehat{\boldsymbol{\pi}}_1, [\mathbf{x}_2^\top], \pi_2, \widetilde{\boldsymbol{\pi}}_2, \varrho)$

$\mathsf{Dec}(\mathsf{SK}, \zeta)$
  $M := e_1 \cdot [\mathbf{x}_1^\top \mathbf{a}]^{-1};\ m := \psi(M)$
  $u :=$ $\mathsf{MDec}(\mathsf{msk}, \varrho)$
      $u := e_3 \cdot [\mathbf{x}_3^\top \overline{\mathbf{a}}]^{-1}$
      $\widehat{\pi}'_{31} := \left[(\mathbf{x}_3^\top + \overline{\boldsymbol{\lambda}}_1^\top \overline{\mathbf{P}}) u \overline{\mathbf{b}}\right]$
      $\widehat{\pi}'_{32} := \left[(\mathbf{x}_3^\top + \overline{\boldsymbol{\lambda}}_2^\top \overline{\mathbf{P}}) u \overline{\mathbf{c}}\right];\ \pi'_4 := [\mathbf{x}_4^\top \overline{\mathbf{a}}]$
      $\widetilde{\pi}'_{41} := \left[\mathbf{x}_4^\top u \overline{\mathbf{b}}\right];\ \widetilde{\pi}'_{42} := \left[\mathbf{x}_4^\top u \overline{\mathbf{c}}\right]$
      $\widehat{\boldsymbol{\pi}}'_3 := (\widehat{\pi}'_{31}, \widehat{\pi}'_{32});\ \widetilde{\boldsymbol{\pi}}'_4 := (\widetilde{\pi}'_{41}, \widetilde{\pi}'_{42})$
      if $(\widehat{\boldsymbol{\pi}}'_3, \widetilde{\boldsymbol{\pi}}'_4, \pi'_4) \neq (\widehat{\boldsymbol{\pi}}_3, \widetilde{\boldsymbol{\pi}}_4, \pi_4)$, return $\perp$
      else return $u$
  if $u = \perp$, return $\perp$
  $\widehat{\pi}'_{11} := \left[(\mathbf{x}_1^\top + \boldsymbol{\lambda}_1^\top \mathbf{P})(u(\mathbf{b} + m\mathbf{c}))\right]$
  $\widehat{\pi}'_{12} := \left[(\mathbf{x}_1^\top + \boldsymbol{\lambda}_2^\top \mathbf{P})(u(\mathbf{d} + m\mathbf{e}))\right]$
  $\widetilde{\pi}'_{21} := \left[\mathbf{x}_2^\top (u(\mathbf{b} + m\mathbf{c}))\right]$
  $\widetilde{\pi}'_{22} := \left[\mathbf{x}_2^\top (u(\mathbf{d} + m\mathbf{e}))\right]$
  $\pi'_2 := [\mathbf{x}_2^\top \mathbf{a}];\ \widehat{\boldsymbol{\pi}}'_1 := (\widehat{\pi}'_{11}, \widehat{\pi}'_{12});\ \widetilde{\boldsymbol{\pi}}'_2 := (\widetilde{\pi}'_{21}, \widetilde{\pi}'_{22})$
  if $(\widehat{\boldsymbol{\pi}}'_1, \widetilde{\boldsymbol{\pi}}'_2, \pi'_2) \neq (\widehat{\boldsymbol{\pi}}_1, \widetilde{\boldsymbol{\pi}}_2, \pi_2)$, return $\perp$
  else return $M$

$\mathsf{Rerand}(\zeta)$
  $r, r' \leftarrow\$ \mathbb{Z}_p;\ r^* \leftarrow\$ \overline{\mathbb{G}};\ [\mathbf{x}_1'^\top] := [\mathbf{x}_1^\top + r\mathbf{x}_2^\top]$
  $e'_1 := e_1 \pi_2^r;\ \widehat{\boldsymbol{\pi}}'_1 := ((\widehat{\pi}_{11} \widetilde{\pi}_{21}^r)^{r^*}, (\widehat{\pi}_{12} \widetilde{\pi}_{22}^r)^{r^*})$
  $[\mathbf{x}_2'^\top] := [r'\mathbf{x}_2^\top];\ \pi'_2 := \pi_2^{r'};\ \widetilde{\boldsymbol{\pi}}'_2 := (\widetilde{\pi}_{21}^{r'r^*}, \widetilde{\pi}_{22}^{r'r^*})$
  $\varrho' :=$ $\mathsf{MRerand}(\mathsf{Maul}(\varrho, r^*))$
      $e'_3 := r^* \cdot e_3;\ r, r' \leftarrow\$ \mathbb{Z}_q^*$
      $([\mathbf{x}_3'^\top], e''_3) := ([\mathbf{x}_3^\top + r\mathbf{x}_4^\top], e'_3 \pi_4^r)$
      $([\mathbf{x}_4'^\top], \pi'_4) := ([r'\mathbf{x}_4^\top], \pi_4^{r'})$
      $\widehat{\boldsymbol{\pi}}'_3 := ((\widehat{\pi}_{31} \widetilde{\pi}_{41}^r)^{r^*}, (\widehat{\pi}_{32} \widetilde{\pi}_{42}^r)^{r^*})$
      $\widetilde{\boldsymbol{\pi}}'_4 := (\widetilde{\pi}_{41}^{r'r^*}, \widetilde{\pi}_{42}^{r'r^*})$
      return $\varrho' := ([\mathbf{x}_3'^\top], e''_3, \widehat{\boldsymbol{\pi}}'_3, [\mathbf{x}_4'^\top], \pi'_4, \widetilde{\boldsymbol{\pi}}'_4)$
  return $\zeta' := ([\mathbf{x}_1'^\top], e'_1, \widehat{\boldsymbol{\pi}}'_1, [\mathbf{x}_2'^\top], \pi'_2, \widetilde{\boldsymbol{\pi}}'_2, \varrho')$
```

Fig. 6. k-Lin-based anonymous Rand-RCCA-secure scheme PKE

Theorem 12. *I_2 is* $\mathsf{CSR\text{-}Smooth}$*.*

(3) Construction of I_3 and I_4. We first describe the framework of I_3 as below.

- $I_3.\mathsf{Setup}(1^n)$. Let $\mathcal{K}_3 = (\mathbb{Z}_q^{k+1})^2$, $\Pi_3 = \overline{\mathbb{G}}^2$ and $\mathcal{T}_3 = \overline{\mathbb{G}}$. Pick $\overline{\boldsymbol{\lambda}}_1, \overline{\boldsymbol{\lambda}}_2 \leftarrow_\$ \mathbb{Z}_q^k$ with $\overline{\boldsymbol{\lambda}}_1 \neq \overline{\boldsymbol{\lambda}}_2$, $\mathsf{ax} = (\overline{\boldsymbol{\lambda}}_1, \overline{\boldsymbol{\lambda}}_2)$ and $\widehat{H}_{(\cdot)} : \overline{\mathcal{X}} \times \mathcal{T}_3 \rightarrow \overline{\mathbb{G}}^2$ is indexed by $\mathsf{sk}_3 \in \mathcal{K}_3$ and ax.
- $I_3.\phi(\mathsf{sk}_3)$. For $\mathsf{sk}_3 = (\overline{\mathbf{b}}, \overline{\mathbf{c}}) \in (\mathbb{Z}_q^{k+1})^2$, outputs $\mathsf{pk}_3 = \big([\overline{\mathbf{P}\mathbf{b}}], [\overline{\mathbf{P}\mathbf{c}}]\big)$.
- $I_3.\mathsf{Priv}(\mathsf{sk}_3, x, \tau)$. For $\mathsf{sk}_3 = (\overline{\mathbf{b}}, \overline{\mathbf{c}}) \in (\mathbb{Z}_q^{k+1})^2$, $x = \big[\overline{\mathbf{x}}^\top\big]$ and $\tau \in \overline{\mathbb{G}}$, outputs hash value $\pi = (\pi_1, \pi_2) = \Big(\Big[(\overline{\mathbf{x}}^\top + \overline{\boldsymbol{\lambda}}_1^\top \overline{\mathbf{P}})\tau\overline{\mathbf{b}}\Big], \Big[(\overline{\mathbf{x}}^\top + \overline{\boldsymbol{\lambda}}_2^\top \overline{\mathbf{P}})\tau\overline{\mathbf{c}}\Big]\Big)$.
- $I_3.\mathsf{Pub}(\mathsf{pk}_3, x, w, \tau)$. For $\mathsf{pk}_3 = \big([\overline{\mathbf{P}\mathbf{b}}], [\overline{\mathbf{P}\mathbf{c}}]\big)$, $x = \big[\mathbf{w}^\top\overline{\mathbf{P}}\big]$ with witness $\mathbf{w}$ and $\tau \in \overline{\mathbb{G}}$, outputs $\pi = (\pi_1, \pi_2) = \Big(\Big[(\mathbf{w}^\top + \overline{\boldsymbol{\lambda}}_1^\top)\tau\overline{\mathbf{P}\mathbf{b}}\Big], \Big[(\mathbf{w}^\top + \overline{\boldsymbol{\lambda}}_2^\top)\tau\overline{\mathbf{P}\mathbf{c}}\Big]\Big)$.
- $I_3.\mathsf{RandX}(x, x', r_x)$. For $x = \big[\overline{\mathbf{x}}^\top\big]$, $x' = \big[\overline{\mathbf{x}}'^\top\big] \in \overline{\mathcal{X}}$ and $r_x \in \mathbb{Z}_q$, outputs $x^* = \big[\overline{\mathbf{x}}^\top + r_x\overline{\mathbf{x}}'^\top\big]$.
- $I_3.\mathsf{RandT}(\tau, r_\tau)$. For $\tau \in \overline{\mathbb{G}}$ and $r_\tau \in \mathbb{Z}_q$, outputs $\tau^* = r_\tau \cdot \tau$.
- $I_3.\mathsf{RandH}(\pi, \pi', r_x, r_\tau)$. For $\pi = (\pi_1, \pi_2), \pi' = (\pi'_1, \pi'_2)$, $r_x \in \mathbb{Z}_q$ and $r_\tau \in \mathbb{Z}_q$, outputs $\pi^* = ((\pi_1 \cdot (\pi'_1)^{r_x})^{r_\tau},\ (\pi_2 \cdot (\pi'_2)^{r_x})^{r_\tau})$.

As for I_4, its algorithms $I_4.\phi, I_4.\mathsf{RandX}, I_4.\mathsf{RandT}$ and $I_4.\mathsf{RandH}$ are the same as $I_3.\phi$, $I_3.\mathsf{RandX}$, $I_3.\mathsf{RandT}$ and $I_3.\mathsf{RandH}$. Besides, $I_4.\mathsf{Setup}$ is the same as $I_3.\mathsf{Setup}$ except that ax is null and the hash function family is $\widetilde{H}_{(\cdot)} : \overline{\mathcal{X}} \times \mathcal{T}_4 \rightarrow \overline{\mathbb{G}}^2$ where $\mathcal{T}_4 = \mathcal{T}_3$. $I_4.\mathsf{Priv}$ and $I_4.\mathsf{Pub}$ are equivalent to $I_3.\mathsf{Priv}$ and $I_3.\mathsf{Pub}$ with $\overline{\boldsymbol{\lambda}}_1 = \overline{\boldsymbol{\lambda}}_2 = \mathbf{0}$.

- $I_4.\mathsf{Priv}(\mathsf{sk}_4, x, \tau)$. For $\mathsf{sk}_4 = (\overline{\mathbf{b}}, \overline{\mathbf{c}}) \in (\mathbb{Z}_q^{k+1})^2$, $x = \big[\overline{\mathbf{x}}^\top\big] \in \overline{\mathcal{X}}$ and $\tau \in \overline{\mathbb{G}}$, outputs $\pi = (\pi_1, \pi_2) = \big(\big[\overline{\mathbf{x}}^\top\tau\overline{\mathbf{b}}\big], \big[\overline{\mathbf{x}}^\top\tau\overline{\mathbf{c}}\big]\big)$.
- $I_4.\mathsf{Pub}(\mathsf{pk}_4, x, w, \tau)$. For $\mathsf{pk}_4 = \big(\big[\overline{\mathbf{P}\mathbf{b}}\big], \big[\overline{\mathbf{P}\mathbf{c}}\big]\big)$, $x = \big[\mathbf{w}^\top\overline{\mathbf{P}}\big]$ with witness $\mathbf{w}$ and $\tau \in \overline{\mathbb{G}}$, outputs $\pi = (\pi_1, \pi_2) = \big(\big[\mathbf{w}^\top\tau\overline{\mathbf{P}\mathbf{b}}\big], \big[\mathbf{w}^\top\tau\overline{\mathbf{P}\mathbf{c}}\big]\big)$.

One can verify the correctness and 1-smoothness of I_3 and I_4. Analogous to the proofs of Theorem 8, 9, 10, 11 and 12, one can easily prove that if $k \geq 2$, I_3 is perfectly pairwise-rerandomizable, $\mathsf{PT\text{-}Smooth}_1$ and $\mathsf{CPR\text{-}Smooth}$ with respect to I_4, and I_4 is perfectly self-rerandomizable and $\mathsf{CSR\text{-}Smooth}$. The concrete proofs are given in the full version [29].

6.3 Concrete PKE from *k*-Lin Assumption

Figure 6 depicts the full concrete scheme PKE based on k-Lin assumption. Note that the group $\overline{\mathbb{G}}$ and $\mathbb{G}$ should be chosen relevantly to ensure that u in tag τ could be encrypted with proper group. Concretely, let $\overline{\mathbb{G}} = \mathbb{QR}_{2q+1}^*$ and $\mathbb{G} = \mathbb{QR}_{2p+1}^*$ be two groups of quadratic residues where $p = 2q+1$ and $(q, 2q+1, 4q+3)$ is a sequence of primes, called a Cunningham chain (of the first kind) of length 3.

Acknowledgement. We would like to thank all anonymous reviewers for their valuable comments. This work is supported in part by the National Natural Science Foundation of China (Grant No. 62032005, Grant No. 61702541, and Grant No. 61872087), Science Foundation of Fujian Provincial Science and Technology Agency (Grant No. 2020J02016).

References

1. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_6
2. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 566–582. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_33
3. Benhamouda, F., Blazy, O., Chevalier, C., Pointcheval, D., Vergnaud, D.: New techniques for SPHFs and efficient one-round PAKE protocols. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 449–475. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_25
4. Benhamouda, F., Bourse, F., Lipmaa, H.: CCA-secure inner-product functional encryption from projective hash functions. In: Fehr, S. (ed.) PKC 2017, Part II. LNCS, vol. 10175, pp. 36–66. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54388-7_2
5. Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 565–582. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_33
6. Chase, M., Kohlweiss, M., Lysyanskaya, A., Meiklejohn, S.: Malleable proof systems and applications. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 281–300. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_18
7. Chen, R., Mu, Y., Yang, G., Susilo, W., Guo, F., Zhang, M.: Cryptographic reverse firewall via malleable smooth projective hash functions. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 844–876. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_31
8. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_4
9. Dodis, Y., Mironov, I., Stephens-Davidowitz, N.: Message transmission with reverse firewalls—secure communication on corrupted machines. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 341–372. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_13
10. Faonio, A., Fiore, D.: Improving the efficiency of re-randomizable and replayable CCA secure public key encryption. In: Conti, M., Zhou, J., Casalicchio, E., Spognardi, A. (eds.) ACNS 2020. LNCS, vol. 12146, pp. 271–291. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57808-4_14
11. Faonio, A., Fiore, D., Herranz, J., Ràfols, C.: Structure-preserving and re-randomizable RCCA-secure public key encryption and its applications. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019. LNCS, vol. 11923, pp. 159–190. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34618-8_6

12. Ganesh, C., Magri, B., Venturi, D.: Cryptographic reverse firewalls for interactive proof systems. In: 47th International Colloquium on Automata, Languages, and Programming (ICALP 2020). Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2020). https://doi.org/10.4230/LIPIcs.ICALP.2020.55
13. Golle, P., Jakobsson, M., Juels, A., Syverson, P.: Universal re-encryption for mixnets. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 163–178. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24660-2_14
14. Groth, J.: Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 152–170. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24638-1_9
15. Han, S., Liu, S., Lyu, L., Gu, D.: Tight leakage-resilient CCA-security from quasi-adaptive hash proof system. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 417–447. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26951-7_15
16. Krawczyk, H.: The order of encryption and authentication for protecting communications (or: how secure is SSL?). In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 310–331. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_19
17. Libert, B., Peters, T., Qian, C.: Structure-preserving chosen-ciphertext security with shorter verifiable ciphertexts. In: Fehr, S. (ed.) PKC 2017, Part I. LNCS, vol. 10174, pp. 247–276. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54365-8_11
18. Mironov, I., Stephens-Davidowitz, N.: Cryptographic reverse firewalls. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 657–686. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_22
19. Peng, K., Nieto, J.M., Desmedt, Y., Dawson, E.: Klein bottle routing: an alternative to onion routing and mix network. In: Rhee, M.S., Lee, B. (eds.) ICISC 2006. LNCS, vol. 4296, pp. 296–309. Springer, Heidelberg (2006). https://doi.org/10.1007/11927587_25
20. Pereira, O., Rivest, R.L.: Marked mix-nets. In: Brenner, M., et al. (eds.) FC 2017. LNCS, vol. 10323, pp. 353–369. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70278-0_22
21. Phan, D.H., Pointcheval, D.: OAEP 3-round: a generic and secure asymmetric encryption padding. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 63–77. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30539-2_5
22. Prabhakaran, M., Rosulek, M.: Rerandomizable RCCA encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 517–534. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74143-5_29
23. Saito, J., Ryou, J.-C., Sakurai, K.: Enhancing privacy of universal re-encryption scheme for RFID tags. In: Yang, L.T., Guo, M., Gao, G.R., Jha, N.K. (eds.) EUC 2004. LNCS, vol. 3207, pp. 879–890. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30121-9_84
24. Senftleben, M., Bucicoiu, M., Tews, E., Armknecht, F., Katzenbeisser, S., Sadeghi, A.-R.: MoP-2-MoP – mobile private microblogging. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 384–396. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45472-5_25
25. Shoup, V.: A proposal for an ISO standard for public key encryption. Cryptology ePrint Archive, Report 2001/112 (2001). http://eprint.iacr.org/2001/112
26. Syverson, P., Dingledine, R., Mathewson, N.: Tor: The second generation onion router. In: Usenix Security (2004). https://www.usenix.org/legacy/events/sec04/tech/full_papers/dingledine/dingledine_html/

27. Wee, H.: KDM-security via homomorphic smooth projective hashing. In: Cheng, C.-M., Chung, K.-M., Persiano, G., Yang, B.-Y. (eds.) PKC 2016, Part II. LNCS, vol. 9615, pp. 159–179. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49387-8_7
28. Young, A.L., Yung, M.: Semantically secure anonymity: foundations of re-encryption. In: Catalano, D., De Prisco, R. (eds.) SCN 2018. LNCS, vol. 11035, pp. 255–273. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98113-0_14
29. Wang, Y., Chen, R., Yang, G., Huang, X., Wang, B., Yung, M.: Receiver-anonymity in rerandomizable RCCA-secure cryptosystems resolved. Cryptology ePrint Archive, Report 2021/862 (2021). http://eprint.iacr.org/2021/862