Public-Key Encryption with Selective Opening Security from General Assumptions

Dali Zhu[1,2], Renjun Zhang[1,2(✉)], Shuang Hu[3,4], and Gongliang Chen[3]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{zhudali,zhangrenjun}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] SKLIATIS, Shanghai Jiao Tong University, Shanghai, China
{hushuang,chengl}@sjtu.edu.cn
[4] Virginia Commonwealth University, Richmond, VA, USA

Abstract. In a selective opening (SO) attack, the attacker can corrupt a subset of senders (or receivers) to open some of the ciphertexts and try to learn information on the plaintexts of unopened ciphertexts. It is important and practical to consider SO attack in encryption scheme. In this paper we study public key encryption (PKE) schemes with SO security. Specifically:

- First, we define a new cryptographic primitive called tweaked lossy encryption, and we prove that it has simulation-based security against sender selective opening chosen plaintext attacks (denoted by SIM-SSO-CPA).
- Second, we provide a general construction of tweaked lossy encryption scheme from extractable Σ-protocol; and we propose two instantiations of tweaked lossy encryption, based on dual-mode commitments and Twin-Cramer-Shoup scheme respectively.
- Finally, we propose a general scheme satisfying indistinguishability-based security against receiver selective opening chosen plaintext attacks (denoted by IND-RSO-CPA), and we give a construction of the scheme from explainable hash proof systems (denoted by EHPS), and we provide the security analysis.

Our results provide a new insight about the relations among PKE schemes with SO security, extractable Σ-protocol and explainable hash proof systems.

Keywords: Selective opening security · Lossy encryption · Extractable Σ-protocol · Explainable hash proof systems

This work was supported in by the National Key Research and Development Program of China-the Key Technologies for High Security Mobile Terminals (Grant No. 2017YFB0801903).

F. Guo et al. (Eds.): Inscrypt 2018, LNCS 11449, pp. 510–530, 2019.
https://doi.org/10.1007/978-3-030-14234-6_27

1 Introduction

Sender Selective Opening Security. Sender selective opening (SSO) attacks are considered in public key encryption (PKE) schemes where an adversary may corrupt some of the senders to selectively open some of the ciphertexts. Specifically, an adversary may get a collection of some arbitrary challenge ciphertexts $\boldsymbol{c} = (c_1, \ldots, c_N)$ of size N, and each ciphertext $c_i = \mathrm{Enc}_{pk}(m_i; r_i)$ is obtained by encrypting the message m_i with a fresh randomness r_i under a public key pk. The adversary may adaptively chooses a subset $I \subseteq [N]$ of ciphertexts to open, thus learning the messages $\{m_i\}_{i \in I}$ and the corresponding randomness $\{r_i\}_{i \in I}$, and then the adversary tries to break into the unopened ciphertexts $\{c_i\}_{i \in [N] \setminus I}$ and obtain additional information on the corresponding plaintexts. The SSO security of PKE schemes requires that the privacy of the unopened ciphertexts is perserved.

Several formal analysis [2,4,6,15,26,27] indicated that SSO security definition is essentially stronger than traditional security definitions due to two added features. One is to enable partial revelation of the randomness, which allows the adversary to check the relation between ciphertexts and the corresponding messages. The other is that the adversary is allowed to open a selected subset of ciphertexts, which gives it more power to learn information on the plaintexts of unopened ciphertexts.

Bellare *et al.* [3] proposed two types of SSO security: indistinguishability-based selective opening (IND-SSO) security and simulation-based selective opening (SIM-SSO) security. IND-SSO security means that no adversary can distinguish between opened messages and unopened messages. It requires the distribution on the messages to be efficiently conditionally resamplable. SIM-SSO security is a stronger notion than IND-SSO security. In particular, SIM-SSO security requires that, a simulator that sees only the opened messages can simulate the output of any real adversary; and it has no restrictions on the message distribution. Unfortunately, SIM-SSO security is much harder to achieve than IND-SSO security [2], since for many natural encryption schemes, there exist no such simulator satisfying the definitions given in [3,13].

Relations among IND-SSO, SIM-SSO and standard security definitions have been discussed in many literatures such as in [2,4,15,19,26,27]. Encryption schemes with IND-SSO security against chosen plaintext attack (IND-SSO-CPA) and SIM-SSO security against chosen plaintext attack (SIM-SSO-CPA) were first given in [3], based on lossy encryption [37,38]. In particular, Bellare *et al.* proved that lossy encryption implies IND-SSO-CPA security, and lossy encryption with efficient opening implies SIM-SSO-CPA security [3]. Later, Hemenway *et al.* [20] proposed a general construction of lossy encryption from hash proof system (HPS). Following this research, Hofheinz *et al.* [25] showed that lossy encryption with efficient weak opening algorithm implies SIM-SSO-CPA security. Note that most of those constructions are based on non-interactive cryptographic primitives; however, the all-but-many encryption (ABME) scheme [16] is built on interactive cryptographic primitives - extractable Σ-protocol, which provides a new and meaningful insight to design SSO secure encryption and related schemes.

Selective Opening Security for the Receiver. Selective opening security for the receiver (RSO) is less studied than SSO security. In the RSO setting, one sender and n receivers hold public and secret keys that are generated independently. The attacker is allowed to learn the messages together with secret keys of a subset of the receivers by corrupting them. Security is requires that the privacy of the uncorrupted receivers is perserved.

There are also indistinguishability-based selective opening for receiver (IND-RSO) security and simulation-based selective opening for receiver (SIM-RSO) security [19]. Some formal arguments show that RSO secure scheme can be constructed from non-committing encryption (NCER) schemes [7,11,36], while there are a few constructions from standard assumptions. In [19], Hazay *et al.* introduced a tweaked variant of NCER which implies IND-RSO security against chosen plaintext attack (IND-RSO-CPA), and shows that tweaked NCER can be constructed from some standard primitives.

Related Work. Deniable encryption [6] and NCER [7] can also be used to construct SSO secure encryption scheme [14,34]. In the indistinguishability-based chosen-ciphertext (IND-SSO-CCA) and simulation-based chosen ciphertext (SIM-SSO-CCA) scenarios, handling additional decryption queries makes it even more difficult to construct schemes with IND-SSO and SIM-SSO security. Several IND-SO-CCA secure schemes have been constructed by lossy trapdoor functions [38], such as all-but-N lossy trapdoor functions [20], and all-but-many lossy trapdoor functions [5,24,32]. And existing SIM-SSO-CCA secure constructions follow dedicated approaches [14,24,29,34]. Heuer *et al.* [23] showed that if a PKE scheme consists of a key encapsulation mechanism (KEM) and a blockcipher-based data encapsulation mechanism (DEM), and the DEM is simulatable, then the PKE scheme is SIM-SO-CCA secure. For sender security, Heuer *et al.* [21,22] showed that some practical schemes, such as RSA-OAEP and DHIES, are SIM-SSO-CCA secure in the random oracle model. For receiver security, IND-RSO-CCA and SIM-RSO-CCA secure constructions were proposed in [18,28,30,31].

1.1 Our Contribution

First, we define a new cryptographic primitive called tweaked lossy encryption, and we show that it is SIM-SSO-CPA secure. We remark that our construction is inspired by that of ABME given by Fujisaki [16], and by that of dual-mode commitments in [33], and by that of simulatable DEMs in [23], and by that of instance-dependent cryptographic primitives in [12,35].

Second, we provide a general construction of tweaked lossy encryption based on extractable Σ-protocol. Furthermore, we give two instantiations of the scheme based on two ways of constructing extractable Σ-protocol, namely dual-mode commitments and Twin-Cramer-Shoup scheme. These instantiations are non-trivial, specifically, the property of efficient weak opening in lossy encryption is usually restricted to specific algebraic structures that are hard to obtain, so we

need to limit the length of the plaintext to logarithmic length. While in a setting where extractable Σ-protocols are executed in parallel, there may be multiple random challenges; and each of them can be associated with a plaintext, therefore it is possible to encrypt longer plaintexts by dividing them into multiple short plaintexts and encrypting each short plaintext in parallel.

Third, we provide an IND-RSO-CPA secure scheme from explainable hash proof systems (denoted by EHPS). In [19], Hazay *et al.* proved that secure tweaked NCER implies IND-RSO-CPA secure PKE, and HPS implies tweaked NCER. Compared with the construction in [19], our tweaked NCER scheme changes the opening algorithm in several ways: for the opening algorithm in [19], an unbounded algorithm can find an appropriate secret key by searching exhaustively without a trapdoor. In our opening algorithm, EHPS provide the trapdoor information, thus the secret key can be obtained in probabilistic polynomial time. Abdalla *et al.* [1] have shown that such EHPS can be constructed.

Organization. The rest of our paper is organized as follows: In Sect. 2 we present some basic notions and tools used; In Sect. 3 we define a new cryptographic primitive called tweaked lossy encryption and show that it is SIM-SSO-CPA secure; In Sect. 4 we describe a generic construction of tweaked lossy encryption scheme from extractable Σ-protocol; In Sect. 5, we give two instantiations of tweaked lossy encryption; In Sect. 6, we give a construction of tweaked NCER from EHPS, and prove that it is IND-RSO-CPA secure.

2 Preliminaries

Notation. We use $\mathbb{N}$ to denote the set of natural numbers, and $\mathbb{Z}$ the set of integers. For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, \ldots, n\}$. The length of a string x is denoted by $|x|$. We abbreviate probabilistic polynomial-time and deterministic polynomial-time as PPT and DPT respectively. Let $x \leftarrow S$ denote the process of picking up x uniformly at random from a finite set S, or the process of sampling x according to a distribution S. Given a security parameter λ, a function $\mathrm{negl}(\lambda)$ is negligible for a sufficiently large λ.

Public Key Encryption. A PKE scheme consists of three PPT algorithms. The key generation algorithm $\mathrm{Gen}(1^\lambda)$ takes a security parameter λ as input, and outputs a public/secret key pair (pk, sk) where pk is a public key and sk is the secret key. The encryption algorithm $\mathrm{Enc}(pk, m; r)$ takes the public key pk, a message $m \in \mathcal{M}$ and randomness $r \in \mathcal{R}$ as inputs, and outputs a ciphertext c, denoted as $c = \mathrm{Enc}(pk, m; r)$. The decryption algorithm $\mathrm{Dec}(sk, c)$ takes the secret key sk, a ciphertext c as inputs, and outputs either a message m denoted as $m = \mathrm{Dec}(sk, c)$ or a special "$\perp$" indicating invalid ciphertext c. We say that a PKE scheme satisfies correctness, if $\mathrm{Dec}(sk, c) = m$ with all but negligible probability for all (pk, sk) produced by $\mathrm{Gen}(1^\lambda)$ and all c produced by $\mathrm{Enc}(pk, m; r)$.

Selective Opening Security. There are several different ways of formalizing SSO security. Following the approach of [4], we consider the definition of SIM-SSO-CPA security, which requires that the output of the adversary can be simulated by an efficient simulator without seeing neither the ciphertext nor the public key. To model adaptive corruptions, the definition provides an opening oracle $\mathcal{O}$ for the adversary and the simulator.

Definition 1 (SIM-SSO-CPA Security [3,4]). *A PKE scheme* $\mathrm{PKE} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *is SIM-SSO-CPA secure iff for every polynomially bounded* $n = n(1^\lambda) > 0$*, and every stateful PPT adversary* $\mathcal{A}$*, there exists a stateful PPT simulator* $\mathcal{S}$ *such that*

$$\mathbf{Adv}^{\mathrm{sim-sso-cpa}}_{\mathrm{PKE},\mathcal{A},\mathcal{S}}(1^\lambda) = |\Pr[\mathrm{Exp}^{\mathrm{real}}_{\mathrm{PKE},\mathcal{A}}(1^\lambda) = 1] - \Pr[\mathrm{Exp}^{\mathrm{ideal}}_{\mathcal{S}}(1^\lambda) = 1]|$$

is negligible. The experiments $\mathrm{Exp}^{\mathrm{real}}_{\mathrm{PKE},\mathcal{A},Rel}$ *and* $\mathrm{Exp}^{\mathrm{ideal}}_{\mathcal{S},Rel}$ *are defined as following (Fig. 1):*

Experiment. $\mathrm{Exp}^{\mathrm{real}}_{\mathrm{PKE},\mathcal{A}}(1^\lambda)$:

$(pk, sk) \leftarrow \mathrm{Gen}(1^\lambda)$
$\mathrm{dist} \leftarrow \mathcal{A}(pk)$
$(M_i)_{i\in[n]} \leftarrow \mathrm{dist}$
$(R_i)_{i\in[n]} \leftarrow (\mathcal{R}_{\mathrm{Enc}})^n$
$(C_i)_{i\in[n]} = \mathrm{Enc}(pk, M_i; R_i)_{i\in[n]}$
$\mathbf{O} = (M_i, R_i)_{i\in[n]}$
$out_{\mathcal{A}} \leftarrow \mathcal{A}^{\mathcal{O}(\cdot)}(\mathrm{select}, (C_i)_{i\in[n]})$
$I = \mathcal{O}(\mathrm{get\ queries})$
return $((M_i)_{i\in[n]}, \mathrm{dist}, I, out_{\mathcal{A}})$

Experiment. $\mathrm{Exp}^{\mathrm{ideal}}_{\mathcal{S}}(1^\lambda)$:

$\mathrm{dist} \leftarrow \mathcal{S}(1^\lambda)$
$(M_i)_{i\in[n]} \leftarrow \mathrm{dist}$
$out_{\mathcal{S}} \leftarrow \mathcal{S}^{\mathcal{O}(\cdot)}(\mathrm{select})$
$I = \mathcal{O}(\mathrm{get\ queries})$
return $((M_i)_{i\in[n]}, \mathrm{dist}, I, out_{\mathcal{S}})$

Fig. 1. The REAL-SIM-SSO-CPA and IDEAL-SIM-SSO-CPA experiments

Interactive Proof System. Let L be a NP language and R a binary relation. An interactive proof system [17] (P, V) for L is a pair of interactive machines, where the prover P is able to convince the verifier V of true statements $x \in L$, which is defined by a binary relation R such that $(x, w) \in R$; while nobody can fool V into believing false statements $x \notin L$. An interactive proof system is zero knowledge [17] if for every interactive machine V^*, there exists an expected probabilistic polynomial-time simulator M^* that can simulate the entire interaction transcript between P and V without accessing P's certificate information. Σ-protocols are special cases of three-round honest verifier zero-knowledge proofs.

Definition 2 (Σ-protocol [10]). *A three-round public-coin protocol $\Sigma = (P_1, P_2, V)$ is a Σ-protocol for a relation R, if it satisfies the following properties:*

- *Completeness: if honest P and V are given common input x and private input w for P, where $(x, w) \in R$, then V will always accept the transcripts.*
- *Special soundness: there exists a PPT algorithm that takes x and a pair of accepting transcripts (a, e, z) and (a, e', z') for x as input, where $e \neq e'$, then it can efficiently compute w such that $(x, w) \in R$.*
- *Special honest verifier zero knowledge: there exists a PPT simulator upon input $x \in L$ and the challenge $e \in \{0,1\}^n$, then outputs an accepting transcript of the form (a, e, z). Moreover, the distribution of the simulator's output is computationally indistinguishable from that of a real execution between P and V on input x.*

Extractable Σ-protocol. The notion of extractable Σ-protocol was proposed in [16]. The following definition essentially repeats the definition of extractable Σ-protocol from [16], expect minor changes to fit our requirement. Let $L = \{L_{pk}\}_{pk}$ be a NP language composed of a series of sets L_{pk} indexed by $pk \in \mathcal{PK}$, where $\mathcal{PK}$ is the space of pk. Let $R_{pk} = \{(x, w)\}$ be a polynomial-time binary relation related to L_{pk}. An extractable Σ-protocol $\Sigma^{\text{ext}} = (\mathrm{P}_\Sigma^{\text{com}}, \mathrm{P}_\Sigma^{\text{ans}}, \mathrm{V}_\Sigma^{\text{vrfy}}, \mathrm{simP}_\Sigma^{\text{com}}, \text{Ext})$ is defined as follows: PPT algorithm $\mathrm{P}_\Sigma^{\text{com}}$ on input $(x, w) \in R_{pk}$ and a random coins r_a, then outputs a commitment $a = \mathrm{P}_\Sigma^{\text{com}}(x, w; r_a)$. DPT algorithm $\mathrm{P}_\Sigma^{\text{ans}}$ takes as input (x, w, r_a, e) and outputs $z = \mathrm{P}_\Sigma^{\text{ans}}(x, w, r_a, e)$, where e is a challenge. DPT algorithm $\mathrm{V}_\Sigma^{\text{vrfy}}$ verifies (x, a, e, z) and decide whether to accept or reject it. PPT algorithm $\mathrm{simP}_\Sigma^{\text{com}}$ takes as input (x, e) and outputs $(a, e, z) = \mathrm{simP}_\Sigma^{\text{com}}(x, e; r_z)$. Similarly, r_z is the random coins, and we additionally require that $r_z = z$, and this property can be satisfied by many sigma protocols. DPT algorithm Ext takes as input (sk, x, a) and outputs e or a special symbol $\perp$ indicating failure, where sk is a secret key corresponding to pk. A protocol Σ^{Ext} is said to be an extractable Σ-protocol on $L = \{L_{pk}\}_{pk}$ for relation R_{pk}, if for all pk, there is a set L_{pk}^{co} such that $L_{pk} \cap L_{pk}^{co} = \emptyset$, and the following properties hold:

- Completeness: if P and V follow the protocol for common input x and private input w for P, then for every $(x, w) \in R_{pk}$ and every r_a and e, the equation $\mathrm{V}_\Sigma^{\text{vrfy}}(x, \mathrm{P}_\Sigma^{\text{com}}(x, w; r_a), e, \mathrm{P}_\Sigma^{\text{ans}}(x, w, r_a, e)) = 1$ is always true.
- Special soundness: for every $x \notin L$ and every a, there exists *exactly one* e such that $\mathrm{V}_\Sigma^{\text{vrfy}}(x, a, e, z) = 1$.
- Extractability: we say that $(pk, sk^{\text{ext}}) \in R^{\text{ext}}$ if there is $e' = \text{Ext}(sk^{\text{ext}}, x, a)$ which satisfies $\mathrm{V}_\Sigma^{\text{vrfy}}(x, a, e', z) = 1$ for all $x \in L_{pk}^{co}$ and all a, so that the equation $\mathrm{V}_\Sigma^{\text{vrfy}}(x, a, e, z) = 1$ holds for an existing pair (e, z). If for all $pk \in \mathcal{PK}$, there is a sk^{ext} that satisfies the relation $(pk, sk^{\text{ext}}) \in R^{\text{ext}}$, we say that Σ^{ext} has the property of extractability on $\{L_{pk}^{co}\}_{pk}$. Combined with special soundness, for all $x \in L_{pk}^{co}$, all e and all z, we can get $e = \text{Ext}(sk, x, \mathrm{simP}_\Sigma^{\text{com}}(x, e; z)_1)$, where $\mathrm{simP}_\Sigma^{\text{com}}(x, e; z)_1$ is the first output of $\mathrm{simP}_\Sigma^{\text{com}}(x, e; z)$.

– Enhanced Honest-Verifier Statistical Zero-Knowledge (eHVSZK): for every $(pk,\ sk^{\text{ext}}) \in R^{\text{ext}}$, every $(x, w) \in R_{pk}$, and every challenge e, the following two ensembles are statistically indistinguishable:

$$\{\text{simP}_{\Sigma}^{\text{com}}(x, e; r_z)\} \approx_s \{(\text{P}_{\Sigma}^{\text{com}}(x, w; r_a), e, \text{P}_{\Sigma}^{\text{ans}}(x, w, r_a, e))\}$$

3 Selective Opening Security from Tweaked Lossy Encryption

In this section, we define a new cryptographic primitive called tweaked lossy encryption (tLPKE), then we show that tLPKE is SIM-SSO-CPA secure. We can think of tLPKE as a lightweight ABME [16].

Definition 3 (Tweaked Lossy Encryption). *A tweaked lossy encryption scheme with message space $\mathcal{M}$ is a tuple of PPT algorithms such that:*

$\text{Gen}(1^\lambda)$: *take the security parameter λ as input and output (pk, sk) where pk is the real public key and sk is the corresponding secret key.*

$\text{LGen}(1^\lambda)$: *take the security parameter λ as input and output a key pair (pk, sk) where pk is the lossy public key and sk is the lossy secret key.*

$\text{Enc}(pk, m; r)$: *take a real or lossy public key pk, a message $m \in \mathcal{M}$, and a random coin $r \in R_{\text{Enc}}$ as inputs, output a ciphertext c.*

$\text{Dec}(sk, c)$: *take a ciphertext c and a secret key sk as inputs, output either a message $m \in \mathcal{M}$ or a special symbol "$\perp$" indicating decryption failure.*

$\text{Sim} = (\text{Fake}, \text{Open})$: *is a pair of PPT and DPT algorithms such that*

- $\text{Fake}(pk, r')$: *take a lossy public key pk and a random coin $r' \leftarrow R_{\text{Enc}}$ as inputs, output a fake ciphertext e^* and an auxiliary parameter ε.*
- $\text{Open}(\varepsilon, m)$: *take an auxiliary parameter ε and a message $m \in \mathcal{M}$ as inputs, output a random coin $r \in R_{\text{Enc}}$.*

Furthermore, tLPKE satisfies the following properties:

– Correctness: for all $(pk, sk) \leftarrow \text{Gen}(1^\lambda)$, message $m \in \mathcal{M}$, $c \leftarrow \text{Enc}(pk, m; r)$, it must satisfy $\text{Dec}(sk, c) = m$.

– Key indistinguishability: for any PPT distinguisher D, there exists a negligible function $\mu(\cdot)$ such that

$$\text{Adv}_{\text{tLPKE},D}^{\text{ind-lossy-key}} = \left| \begin{array}{l} \Pr[D(pk, 1^\lambda) = 1 | (pk, sk) \leftarrow \text{Gen}(1^\lambda)] - \\ \Pr[D(pk, 1^\lambda) = 1 | (pk, sk) \leftarrow \text{LGen}(1^\lambda)] \end{array} \right| \leq \mu(\lambda)$$

– Simulatability: define the following variables:

- $\text{dist}^{\text{Enc}}(pk, m)$ denotes the random variable (c, r), where $r \leftarrow R_{\text{Enc}}$, $c = \text{Enc}(pk, m; r)$ and pk is the lossy public key.
- $\text{dist}^{\text{Sim}}(pk, m)$ denotes the random variable (e^*, r), where $(e^*,\ \varepsilon) \leftarrow \text{Fake}(pk, r')$ and pk is the lossy public key; $r' \leftarrow R_{\text{Enc}}$; $r \leftarrow \text{Open}(\varepsilon, m)$.

Therefore the following ensembles are statistically indistinguishable:

$$\{\text{dist}^{\text{Enc}}(pk, m)\} \approx_s \{\text{dist}^{\text{Sim}}(pk, m)\}$$

Remarks. The recent work of Heuer *et al.* [23] defined a simulatability property that holds for DEM. Intuitively, the encapsulation algorithm could generate a ciphertext without seeing the corresponding message. Formally, they divided the encapsulation process into two parts in sequential order, *Fake* and *Make*. Firstly, Fake algorithm outputs a ciphertext c before seeing the message m; Then, Make algorithm takes as input the message m and tries to find a possible permutation instance, under which m would be encapsulated to the ciphertext c. Our ideas are similar to those of Heuer *et al.*, the main difference is that we use Open algorithm to replace the Make algorithm. Their Make algorithm exhaustively searches for the appropriate permutation, and may be inefficient. Our Open algorithm introduces auxiliary parameters, including the lossy secret key and some random coins, which makes the Open algorithm running in probabilistic polynomial time.

Theorem 1. *The tweaked lossy encryption scheme is SIM-SSO-CPA secure.*

Proof (Sketch). The proof of Theorem 1 is similar to the proof of lossy encryption with efficient opening implies SIM-SSO-CPA security in [3], so we will only sketch it here. Consider the following sequence of games:

$Game_0$: the REAL-SIM-SSO-CPA game.

$Game_1$: the same as $Game_0$ except that the adversary is given a lossy public key and lossy ciphertexts. $Game_1$ and $Game_0$ are indistinguishable by the property of key indistinguishability.

$Game_{2,0}$: the same as $Game_1$ except that for the first ciphertext we replace $\mathrm{Enc}(pk, m_1; r_1)$ with $\mathrm{Fake}(pk, r'_1)$. In the corrupt procedure, instead of opening the first ciphertext by revealing the actual coins if $1 \in [I]$, $Game_{2,0}$ runs $\mathrm{Open}(\varepsilon, \mathbf{M}[1])$ algorithm on the actual message and returns r_1, which is the random coins used to generate the first ciphertext. If $1 \notin [I]$, $Game_{2,0}$ reveals the actual coins and returns the output. The view of adversary in this game is statistically close to that in the $Game_0$, since the variables $\{\mathrm{dist}^{\mathrm{Enc}}(pk, m)\}$ and $\{\mathrm{dist}^{\mathrm{Sim}}(pk, m)\}$ are statistically indistinguishable.

$Game_{2,j}$: in the j-th hybrid game, we use $\mathrm{Fake}(\cdot)$ algorithm instead of $\mathrm{Enc}(\cdot)$ algorithm to generate the first j ciphertexts. In the corrupt procedure, $Game_{2,j}$ runs the $\mathrm{Open}(\varepsilon, \mathbf{M}[k])$ algorithm on the actual messages and returns r_k if $k \in [I]$ and $k \leq j$; Otherwise, $Game_{2,j}$ reveals the actual coins and returns the output.

$Game_{2,n}$: in the last hybrid, all ciphertexts are generated using the $\mathrm{Fake}(\cdot)$ algorithm. In the corrupt procedure, $Game_{2,n}$ runs the $\mathrm{Open}(\varepsilon, \mathbf{M})$ algorithm on the actual messages and returns the output.

$Game_3$: the same as $Game_{2,n}$ except that the sampling of $\mathbf{M}$ is moved before the $\mathrm{Open}(\varepsilon, \mathbf{M})$ algorithm and after the $\mathrm{Fake}(\cdot)$ algorithm. The $\mathrm{Fake}(\cdot)$ algorithm no longer requires a message vector, therefore it does not change the view of the adversary.

Now we can construct a simulator $\mathcal{S}$ that runs the adversary $\mathcal{A}$ as its subroutine, just as $\mathcal{A}$ behaves in $Game_3$. Specially, $\mathcal{S}$ chooses a lossy key pair and gives $\mathcal{A}$

the lossy public key and n fake ciphertexts under lossy public key. In case of corruption, when $\mathcal{A}$ makes a query with a set I, $\mathcal{S}$ forwards the same set I to its own challenge oracle. After receiving the messages $\mathbf{M}_I$, $\mathcal{S}$ then uses the efficient algorithm $\mathrm{Open}(\varepsilon, \mathbf{M})$ to open the fake ciphertexts to the messages $\mathbf{M}_I$. Finally, after $\mathcal{A}$ outputs a string *out*, the simulator $\mathcal{S}$ will output the same value. Since both $\mathcal{A}$ and $\mathrm{Open}(\varepsilon, \mathbf{M})$ are efficient, the simulator $\mathcal{S}$ is also efficient. Because the neighboring games are either statistically indistinguishable or computationally indistinguishable, the theorem holds.

4 Tweaked Lossy Encryption Scheme from Extractable Σ-protocol

In this section, we build a tweaked lossy encryption scheme from extractable Σ-protocol that is defined on a membership-hard language L with efficient sampling [33].

4.1 Membership-Hard Languages with Efficient Sampling

Let L be a language. Let S_L be a sampling algorithm that takes a bit b as input. If $b = 0$, S_L outputs an instance in the language L together with a corresponding witness w. If $b = 1$, S_L outputs an instance not in the language L. It is required that no PPT distinguisher can tell which bit S_L received. Let S_L^x denote the instance of the output of the sampling algorithm S_L, we now recall the formal definition of the membership-hard languages with efficient sampling.

Definition 4 ([33]). *We say that a language L satisfies membership-hard with efficient sampling, if there exists a PPT sampling algorithm S_L such for any PPT distinguisher D, the advantage $\mathrm{Adv}_D^x(n)$ defined below is negligible:*

$$\mathrm{Adv}_D^x(n) = |\Pr[D(S_L^x(0,1^n),1^n)=1] - \Pr[D(S_L(1,1^n),1^n)=1]|$$

4.2 Our Scheme

Our scheme is inspired by the general framework for constructing ABME in [16], and is also inspired by the general construction of dual-mode commitment in [33]. The detailed construction of scheme (Gen, LGen, Enc, Dec, Sim.Fake, Sim.Open) as in Definition 3 is as following (denote by **Construction 1**):

$\mathrm{Gen}(1^\lambda)$: run the sampling algorithm $S_L(1,1^\lambda)$ to get x, while $x \notin L$, where L is a membership-hard language. Let $\Sigma^{\mathrm{ext}} = (\mathrm{P}_\Sigma^{\mathrm{com}}, \mathrm{P}_\Sigma^{\mathrm{ans}}, \mathrm{V}_\Sigma^{\mathrm{vrfy}}, \mathrm{simP}_\Sigma^{\mathrm{com}}, \mathrm{Ext})$ be an extractable Σ-protocol defined on L, where Ext is a deterministic extraction algorithm, and let sk^{ext} be the secret key of Ext. The real public key is $pk = x$ and the real secret key is $sk = sk^{\mathrm{ext}}$.

$\mathrm{LGen}(1^\lambda)$: run the sampling algorithm $S_L(0,1^\lambda)$ to obtain (x,w), with w being the witness for $x \in L$. The lossy public key is $pk = x$ and the lossy secret key is $sk = w$.

$\mathrm{Enc}(pk, m; r)$: to encrypt a message $m \in \{0,1\}^n$, run $\mathrm{simP}_{\Sigma}^{\mathrm{com}}(x, m; r)$ algorithm and output (a, m, r), denoted as $(a, m, r) = \mathrm{simP}_{\Sigma}^{\mathrm{com}}(x, m; r)$. The ciphertext is $c = a$.

$\mathrm{Dec}(sk^{\mathrm{ext}}, x, c)$: take the secret key sk^{ext}, the public key x and the ciphertext c as inputs, output $m = \mathrm{Ext}(sk^{\mathrm{ext}}, x, c)$.

$\mathrm{Sim.Fake}(pk, w; r_a)$: take the lossy public key pk, witness w and a random coins r_a as inputs, output (e^*, ε) such that $e^* = \mathrm{P}_{\Sigma}^{\mathrm{com}}(x, w; r_a)$ and $\varepsilon = (pk, w, r_a)$.

$\mathrm{Sim.Open}(\varepsilon, m)$: take ε and m as inputs, and output $r = \mathrm{P}_{\Sigma}^{\mathrm{ans}}(x, w, r_a, m)$.

4.3 Security Analysis

We then prove that the scheme in **Construction 1** is a tweaked lossy encryption scheme.

Theorem 2. *The scheme in **Construction 1** is a tweaked lossy encryption scheme if L is a membership-hard language with efficient opening, and $\Sigma^{\mathrm{ext}} = (\Sigma, \mathrm{Ext})$ is an extractable Σ-protocol for L.*

Proof. **Correctness.** If $x \notin L$, according to the special soundness property of Σ-protocol, for every a, there exists a unique m such that (a, m, r) is an accepting transcription on x, therefore m can be decrypted correctly using secret key sk^{ext}.

Key indistinguishability. The real public key is an instance not in language L, while the lossy public key is an instance in language L. Since L is a membership-hard language, the real public key and the lossy public key are computationally indistinguishable.

Simulatability. While $x \in L$, Sim holds the witness for $x \in L$ and can run the real Σ-protocol: First, the prover runs the Sim.Fake algorithm $\mathrm{P}_{\Sigma}^{\mathrm{com}}(x, w; r_a)$ where $r_a \leftarrow R_{\mathrm{Enc}}$ and outputs a commitment $a = \mathrm{P}_{\Sigma}^{\mathrm{com}}(x, w; r_a)$, then sends the commitment a to the verifier; Second, the verifier sends a challenge m after receiving the commitment a; Finally, the prover runs the Sim.Open $\mathrm{P}_{\Sigma}^{\mathrm{ans}}(x, w, r_a, m)$ algorithm and outputs $r = \mathrm{P}_{\Sigma}^{\mathrm{ans}}(x, w, r_a, m)$, where r is the random coins used in the encryption algorithm, then sends r to the verifier. Since $x \in L$, according to the eHVSZK property of extractable Σ-protocol, the following two distributions are statistically indistinguishable:

$$\{\mathrm{simP}_{\Sigma}^{\mathrm{com}}(x, m; r)\} \approx_s \{(\mathrm{P}_{\Sigma}^{\mathrm{com}}(x, w; r_a), m, \mathrm{P}_{\Sigma}^{\mathrm{ans}}(x, w, r_a, m))\}$$

where $(a, m, r) = \mathrm{simP}_{\Sigma}^{\mathrm{com}}(x, m; r)$, thus the simulatability property follows readily.

5 Instantiations

In this section, we present two instantiations of tLPKE scheme from extractable Σ-protocol, based on dual-mode commitments [33] and Twin-Cramer-Shoup scheme [8,9] respectively.

5.1 Instantiation of tLPKE from Dual-Mode Commitments

Let g be a generator of a group G of prime order q, and we assume that G is efficiently samplable. Let $u = g^x$ and $v = h^x$ with $h = g^r$ where $x, r \leftarrow \mathbb{Z}_q^*$. We define the language L as:

$$L = \{(u, v) | \exists\ x : u = g^x, v = h^x\}.$$

We then construct an instantiation of the tLPKE as follows, and we denote it as **Instantiation 1**.

- Gen(1^λ): given a security parameter λ, run $\mathcal{G}(1^\lambda)$ to obtain a tuple $(\mathbb{G}, q, g)$. Choose $r, x_1, x_2 \leftarrow \mathbb{Z}_q^*$, and compute $h = g^r$, $u = g^{x_1}$, $v = h^{x_2}$. Output the public key $pk = (\mathbb{G}, q, g, h, u, v)$ and the secret key $sk = r$.
- LGen(1^λ): given a security parameter λ, run $\mathcal{G}(1^\lambda)$ to obtain a tuple $(\mathbb{G}, q, g)$. Choose $r, x \leftarrow \mathbb{Z}_q^*$, and compute $h = g^r$, $u = g^x$, $v = h^x$. Output the lossy public key $pk = (\mathbb{G}, q, g, h, u, v)$ and the lossy secret key $sk = x$.
- Enc($pk, m; z$): to encrypt a message $m \in \{0,1\}^k$ where $k = O(\log \lambda)$, choose $z \leftarrow \mathbb{Z}_q^*$, and compute $a = g^z \cdot u^m$, $b = h^z \cdot v^m$. Output the ciphertext as $c = (a, b)$.
- Dec(sk, c): take as input the real secret key $sk = r$ and the ciphertext $c = (a, b)$, search for $m \in \{0,1\}^k$ such that

$$\frac{a^r}{b} = \left(\frac{u^r}{v}\right)^m$$

  and output m.
- Sim.Fake(pk, w): choose $w \in \mathbb{Z}_q^*$ uniformly at random, and compute $a = g^w$, $b = h^w$. Output the fake ciphertext $e^* = (a, b)$ and the auxiliary parameter $\varepsilon = (x, w)$.
- Sim.Open: take as input the auxiliary parameter $\varepsilon = (x, w)$, the message m and the ciphertext (a, b), and output the random coins $z = w - mx$.

We can see that when using the real public key, the Enc algorithm runs a simulation algorithm of the extractable Σ-protocol on L; however, when using the lossy public key, the Enc runs a real extractable Σ-protocol on L with witness x. We now give a detailed analysis.

Theorem 3. *The **Instantiation 1** is a tweaked lossy encryption scheme if DDH assumption holds.*

Proof. **Correctness.** In the decryption mode, where $(u, v) \notin L$. Since $a = g^{z+x_1 m}$ and $b = g^{rz+rx_2 m}$, implying that:

$$\begin{pmatrix} \log a \\ \log b \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ r & rx_2 \end{pmatrix} \begin{pmatrix} z \\ m \end{pmatrix}$$

Since $x_1 \neq x_2$, the determinant of $\begin{pmatrix} 1 & x_1 \\ r & rx_2 \end{pmatrix}$ is nonzero, and $\begin{pmatrix} z \\ m \end{pmatrix}$ is uniquely determined, such that

$$\frac{a^r}{b} = \left(\frac{u^r}{v}\right)^m$$

Therefore the decryption algorithm can search $m \in \{0,1\}^k$ within $O(2^k)$ steps where $k = O(\log \lambda)$.

Key Indistinguishability. The real public keys and the lossy public keys are computationally indistinguishable under the DDH assumption.

Simulatability. Consider an extractable Σ-protocol scenario, where the prover knows x such that $u = g^x$, $v = h^x$ and $(u, v) \in L$. The messages of the protocol include (a, b), the challenge m and the random coins z which is the response to the challenge m. And $((a, b), m, z)$ forms an accepting transcript of the protocol's execution for challenge m, where $m \in \{0,1\}^\lambda$, $a = g^{z+mx}$ and $b = g^{rz+rmx} = a^r$. Since r, z, x are random elements of $\mathbb{Z}_q^*$, a and b are also random elements of $\mathbb{G}$, and the ciphertext (a, b) contains no information of the plaintext. Now consider the algorithm Sim.Fake and Sim.Open. Sim.Fake picks $w \in \mathbb{Z}_q^*$ uniformly at random, and let $a = g^w$, $b = h^w$. Sim.Open takes as input the auxiliary parameter $\varepsilon = (x, w)$, the message m and the ciphertext (a, b), and it outputs $z = w - mx$. Because z, w, x are in linear relations, the distribution of z is identical to that of w and x, therefore z is a random element in $\mathbb{Z}_q^*$. According to the above analysis, we have

$$\{(g^{z+mx}, g^{rz+rmx}), m, z\} \approx_s \{(g^w, h^w), m, z\}.$$

Thus, the simulatability property holds.

5.2 Instantiation of tLPKE from Twin-Cramer-Shoup Scheme

The instantiation in Sect. 5.2 is inspired by the similar instantiation of ABME from Twin-Cramer-Shoup in [16]. Let g be a generator of a group G of prime order q, and assume G is efficiently samplable and the DDH assumption holds on G. Choose $x \leftarrow \mathbb{Z}_q^*$ uniformly at random, and set $X = g^x$. Choose $\xi \leftarrow G$, $v_0 \leftarrow \mathbb{Z}_q^*$ uniformly at random, then compute $d_0 = g^{v_0}$, $e_0 = \xi^{-1} X^{v_0}$. Compute $d = g^v$, $e = \xi X^v$, where $v \leftarrow \mathbb{Z}_q^*$. Set $\lambda = O(\log k)$. Then we define the language L_{cs} under $pk = (g, X, d_0, d_0 d, e_0 e)$:

$$L_{cs} = \{(d, e) | \exists(\tilde{v}, v) : d_0 d = g^{\tilde{v}}, e_0 e = X^{\tilde{v}}, \text{where } \tilde{v} = v_0 + v\}.$$

We then build another instantiation of the tLPKE scheme as follows (denote by **Instantiation 2**):

- Gen(1^λ): choose $x \leftarrow \mathbb{Z}_q^*$ uniformly at random, and set $X = g^x$. Choose $\xi \leftarrow G$, $v_1 \leftarrow \mathbb{Z}_q^*$ and $v_2 \leftarrow \mathbb{Z}_q^*$ uniformly at random, then compute $d_0 = g^{v_1}$ and $e_0 = \xi^{-1} X^{v_2}$. Let $d = g^v$, $e = \xi X^v$, where $v \leftarrow \mathbb{Z}_q^*$, let $\lambda = O(\log k)$, and finally output a pair of real keys (pk, sk), where $pk = (g, X, d_0, d_0 d, e_0 e)$ and $sk = x$.
- LGen(1^λ): choose $x \leftarrow \mathbb{Z}_q^*$, $\xi \leftarrow G$, $v_0 \leftarrow \mathbb{Z}_q^\star$ and $v \leftarrow \mathbb{Z}_q^*$ uniformly at random, and set $X = g^x$, $d_0 = g^{v_0}$, $e_0 = \xi^{-1} X^{v_0}$ and $d = g^v$, $e = \xi X^v$ respectively. Let $\lambda = O(\log k)$, and finally output a pair of lossy keys (pk, sk), where $pk = (g, X, d_0, d_0 d, e_0 e)$ and $sk = (v_0, v)$.

– $\mathrm{Enc}(pk, m; \boldsymbol{z})$: to encrypt a message $m \in \{0,1\}^n$, divide m into $(m_1, \ldots, m_l)$, where $l = n/\lambda$; and for all $1 \le i \le l$, $m_i \in \{0,1\}^\lambda$, choose $\boldsymbol{z} \leftarrow \mathbb{Z}_q^*$ uniformly at random, where $\boldsymbol{z} = (z_1, \ldots, z_l)$, then compute:

$$A = \begin{pmatrix} g & d_0 d \\ X & e_0 e \end{pmatrix} \begin{pmatrix} z_1 & \ldots & z_l \\ m_1 & \ldots & m_l \end{pmatrix}$$

And finally output the ciphertext $c = A$.

– $\mathrm{Dec}(sk, c)$: let $A = (\boldsymbol{a_1}, \ldots, \boldsymbol{a_l})$, where $\boldsymbol{a_i} = (a_{1,i}, a_{2,i})^T$. For every $i \in [l]$, search for appropriate $m_i \in \{0,1\}^\lambda$ such that:

$$\frac{(a_{1,i})^x}{a_{2,i}} = \left(\frac{(d_0 d)^x}{e_0 e}\right)^{m_i}, \text{ if } e_0 e \neq (d_0 d)^x.$$

If such m_i can not be found, then output the decryption failure symbol "$\perp$"; Otherwise output $m = (m_1, \ldots, m_l) \in \{0,1\}^n$.

– $\mathrm{Sim.Fake}(pk; \boldsymbol{w})$: first divide m into $(m_1, \ldots, m_i)$, where $i \in [l]$, then pick up $w_i \in \mathbb{Z}_q^*$ uniformly at random, and compute $a_{1,i} = g^{w_i}$, $a_{2,i} = X^{w_i}$. Output the fake ciphertext $c = (a_{1,i}, a_{2,i})$ and $\varepsilon = (v_0, v, \boldsymbol{w})$ for $i \in [l]$.

– $\mathrm{Sim.Open}(\varepsilon, m)$: take as input the auxiliary parameter $\varepsilon = (x, w)$, the message $(m_1, \ldots, m_i)$ and the ciphertext $c = (a_{1,i}, a_{2,i})$ where $i \in [l]$, output $\boldsymbol{z}$ where $z_i = w_i - m_i \cdot \tilde{v}$ and $\tilde{v} = v_0 + v$.

We then prove that the **Instantiation 2** is a tweaked lossy encryption scheme.

Proof. **Correctness.** When $(d, e) \notin L_{cs}$, the encryption algorithm uses the real public key pk to encrypt message and the resulting ciphertext is

$$A = \begin{pmatrix} g^{z_1}(d_0 d)^{m_1} & \ldots & g^{z_i}(d_0 d)^{m_i} & \ldots & g^{z_l}(d_0 d)^{m_l} \\ X^{z_1}(e_0 e)^{m_1} & \ldots & X^{z_i}(e_0 e)^{m_i} & \ldots & X^{z_l}(e_0 e)^{m_l} \end{pmatrix}$$

Note that $\mathrm{rank}(A) = 2$. Let $\boldsymbol{a_i} = (a_{1,i}, a_{2,i})^T$ denotes the i-th column of the matrix A, then $a_{1,i} = g^{z_i + (v_1 + v) m_i}$, and $a_{2,i} = X^{z_i + (v_2 + v) m_i} = (g^x)^{z_i + (v_2 + v) m_i}$. We can see that $e_0 e \neq (d_0 d)^x$, and therefore m_i can be recovered through the following equation:

$$\frac{(a_{1,i})^x}{a_{2,i}} = \left(\frac{(d_0 d)^x}{e_0 e}\right)^{m_i}$$

Therefore the decryption algorithm can output $m = (m_1, \ldots, m_l) \in \{0,1\}^n$ correctly.

Key indistinguishability. The lossy public key is $pk = (g, g^x, g^{v_0 + v}, g^{x(v_0 + v)})$, and the real public key is $pk = (g, g^x, g^{v_0 + v_1}, g^{x(v_0 + v_2)})$. Therefore, the lossy public key and the real public key are computationally indistinguishable under the DDH assumption.

Simulatability. When $(d, e) \in L_{cs}$, the encryption algorithm runs a real extractable Σ-protocol, where the common input is $(d_0 d, e_0 e)$ and the prover wants to prove that $\log_g d_0 d = \log_X e_0 e$. The vector $\boldsymbol{a_i} = (a_{1,i}, a_{2,i})^T$ is the

first message of the protocol, $(m_1, \ldots, m_l)$ is the parallel challenge, and $\boldsymbol{z_i}$ corresponds to the response to each challenge m_i. Therefore $(A, \boldsymbol{m}, \boldsymbol{z})$ is an accepting proof of the parallel execution of the extractable Σ-protocols, where $\boldsymbol{m} = (m_1, \ldots, m_l)$ is the challenge, and $m_i \in \{0,1\}^\lambda$, and $i \in [l]$.
Specifically, $(a_{1,i}, a_{2,i})^T = (g^{z_i + m_i\tilde{v}}, X^{z_i + m_i\tilde{v}})^T$, where $X = g^x$, $\tilde{v} = v_0 + v$, and $z_i, v_0, v \leftarrow \mathbb{Z}_q^*$. Since $\mathrm{rank}(A) = 1$, the ciphertexts will be uniformly distributed over $\mathbb{G}$. Now, the Sim.Fake algorithm first divides m into $(m_1, \ldots, m_i)$, where $i \in [l]$, then it chooses $w_i \in \mathbb{Z}_q^*$ uniformly at random and sets $a_{1,i} = g^{w_i}$, $a_{2,i} = X^{w_i}$. The Sim.Open algorithm takes as input $\varepsilon = (v_0, v, \boldsymbol{w})$, and finally outputs $z_i = w_i - m_i \cdot \tilde{v}$, where $\tilde{v} = v_0 + v$. We can see that z_i is a random element of $\mathbb{Z}_q^*$. According to the above analysis, we have:

$$\{(g^{z_i + m_i\tilde{v}}, X^{z_i + m_i\tilde{v}})^T, m_i, z_i\}_{i\in[l]} \approx_s \{(g^{w_i}, X^{w_i})^T, m_i, z_i\}_{i\in[l]}$$

Thus, the variables $\mathrm{dist}^{\mathrm{Enc}}(pk, m)$ and $\mathrm{dist}^{\mathrm{Sim}}(pk, m)$ are statistically indistinguishable and the simulatability property holds.

6 Selective Opening Security for the Receiver

Following the work in [3,4,19], we recall the definition of IND-RSO-CPA, which is restricted to efficiently conditionally resamplable distributions.

Definition 5 (Efficiently Conditionally Resamplable [3,4]). *Let* dist *be a joint distribution over $\mathcal{M}^n$, where $\mathcal{M}$ is the message space. We say that* dist *is efficiently conditionally resamplable if there is a PPT algorithm* $\mathrm{ReSamp}_{\mathrm{dist}}$*, such that for any $I \subset [n]$ and any $\boldsymbol{m}_I$:=$(m_i)_{i\in I}$, $\mathrm{ReSamp}_{\mathrm{dist}}(\boldsymbol{m}_I)$ outputs $\boldsymbol{m}'_I$, and $\boldsymbol{m}'_I$ is sampled from the distribution* dist*, conditioned on $m'_i = m_i$ for all $i \in I$.*

Experiment. $\mathrm{Exp}^{\text{IND-RSO-CPA}}_{\mathrm{PKE},\mathcal{A}}(1^\lambda)$:

$b \leftarrow \{0,1\}$
$(\boldsymbol{pk}, \boldsymbol{sk}) := (pk_i, sk_i)_{i\in n} \leftarrow \mathrm{Gen}(1^\lambda)$
$(\mathrm{dist}, \mathrm{ReSamp}_{\mathrm{dist}}) \leftarrow \mathcal{A}(pk)$
$\boldsymbol{M_0} \leftarrow \mathrm{dist}$
$\boldsymbol{C^*} \leftarrow \mathrm{Enc}(\boldsymbol{pk}, \boldsymbol{M_0})$
$I \leftarrow \mathcal{A}(\boldsymbol{C^*})$
$\boldsymbol{M_1} \leftarrow \mathrm{ReSamp}_{\mathrm{dist}}(\boldsymbol{M_{0I}})$
$b' \leftarrow \mathcal{A}(\boldsymbol{sk_I}, \boldsymbol{M_b})$
Return 1 if $b' = b$ and 0 otherwise.

Fig. 2. The IND-RSO-CPA experiment

Definition 6 (IND-RSO-CPA Security [19]). *A PKE scheme* PKE = (Gen, Enc, Dec) *is IND-RSO-CPA secure if for any polynomially bounded* $n = n(1^\lambda) > 0$*, and any stateful PPT adversary* $\mathcal{A}$*, such that*

$$\mathbf{Adv}^{\text{IND-RSO-CPA}}_{\text{PKE},\mathcal{A}}(1^\lambda) = \Pr[\text{Exp}^{\text{IND-RSO-CPA}}_{\text{PKE},\mathcal{A}}(1^\lambda) = 1] - \frac{1}{2}$$

is negligible. The experiment $\text{Exp}^{\text{IND-RSO-CPA}}_{\text{PKE},\mathcal{A}}(1^\lambda)$ *is defined as in Fig. 2.*

6.1 Tweaked NCER for Receivers

In [19], Hazay *et al.* proved that secure tweaked NCER implies IND-RSO-CPA secure PKE. The following definition essentially repeats the definition of tweaked NCER from [19] with small changes: the tGen algorithm outputs a trapdoor for solving hard problem instances, and the tOpen algorithm may receive the trapdoor as an additional input.

A tweaked NCER scheme tPKE is a tuple of algorithms (tGen, tEnc, tEnc*, tDec, tOpen), where (tGen, tEnc, tDec) form a PKE. The fake encryption algorithm tEnc* takes the secret key sk, the public key pk and a message m as inputs, and it outputs a ciphertext $c^* \leftarrow \text{tEnc}^*(pk, sk, m)$. The opening algorithm tOpen takes as input the secret key sk, the trapdoor τ, the public key pk, fake ciphertext c^* where $c^* \leftarrow \text{tEnc}^*(pk, sk, m')$ for some $m' \in \mathcal{M}$ and a message m, and it outputs sk^* such that $m = \text{tDec}(sk^*, c^*)$. For correctness, we want $m = \text{tDec}(sk, c)$ hold for all $m \in \mathcal{M}$, all $(pk, sk) \leftarrow \text{tGen}(1^\lambda)$ and all $c \leftarrow \text{tEnc}(pk, m)$. For security, we require that real ciphertexts and fake ciphertexts are indistinguishable, and a fake ciphertext can be decrypted to a concrete predetermined plaintext (Fig. 3).

Experiment. $\text{Exp}^{\text{ind-tcipher}}_{\text{tPKE}}(\mathcal{A})$:

$b \leftarrow \{0,1\}$
$(pk, sk) \leftarrow \text{tGen}(1^\lambda)$
$(m, state) \leftarrow \mathcal{A}(pk)$
$c_0 \leftarrow \text{tEnc}(pk, m)$
$c_1 \leftarrow \text{tEnc}^*(pk, sk, m)$
$b' \leftarrow \mathcal{A}(sk, c_b, state)$
if $b = b'$, outputs 1, else outputs 0

Experiment. $\text{Exp}^{\text{ind-tncer}}_{\text{tPKE}}(\mathcal{A})$:

$b \leftarrow \{0,1\}$
$(pk, sk_0, \tau) \leftarrow \text{tGen}(1^\lambda)$
$(m, state) \leftarrow \mathcal{A}(pk)$
$c_0 \leftarrow \text{tEnc}^*(pk, sk_0, m)$
$m' \leftarrow \mathcal{M}$
$c_1 \leftarrow \text{tEnc}^*(pk, sk_0, m')$
$sk_1 \leftarrow \text{tOpen}(\tau, sk, c_1, m)$
$b' \leftarrow \mathcal{A}(sk_b, c_b, state)$
if $b = b'$, outputs 1, else outputs 0

Fig. 3. Tweaked NCER

Definition 7 (Tweaked NCER [19]). *We say that a tweaked NCER scheme tPKE is secure if it satisfies the following two conditions:*

- *for any PPT adversary* $\mathcal{A}$, $\mathrm{Adv}^{\textit{ind-tcipher}}_{tPKE,\mathcal{A}} := |\Pr[\mathrm{Exp}^{\textit{ind-tcipher}}_{tPKE}(\mathcal{A}) = 1] - \frac{1}{2}|$ *is negligible.*
- *for any unbounded adversary* $\mathcal{A}$, $\mathrm{Adv}^{\textit{ind-tncer}}_{tPKE,\mathcal{A}} := |\Pr[\mathrm{Exp}^{\textit{ind-tncer}}_{tPKE}(\mathcal{A}) = 1] - \frac{1}{2}|$ *is negligible.*

Despite the fact that we use the trapdoor to help the tOpen algorithm output the secret key, the following lemma still follows from the corresponding proof in [19] with respect to the definition of tweaked NCER.

Lemma 1 ([19]). *If there exists an* $\{$*ind-tcipher, ind-tncer*$\}$ *secure tweaked NCER, then there exists a PKE that is IND-RSO-CPA secure.*

6.2 Explainable Hash Proof Systems

Hash Proof Systems (HPS) or Smooth Projective Hash Functions (SPHFs) refer to a family of hash functions $(\mathrm{Hash}, \mathrm{ProjHash})$ defined on a language $L \subseteq X$, and are indexed by a pair of associated keys (hk, hp), where hk is the hashing key and the secret key, and hp is the projection key and the public key. The projective property of SPHFs stipulates that, for a word $x \in L$, the hash value can be computed using either a hashing key hk or a projection key hp with a witness w for $x \in L$. In contrast, the smoothness property of SPHFs stipulates that for a word $x \notin L$, the hash value should be completely undetermined.

Explainable Hash Proof Systems (EHPS) proposed by Abdalla *et al.* [1] are SPHFs with an additional property that: given the trapdoor, we can first generate a random-looking projection key hp, and finally output a valid hashing key hk corresponding to the projection key hp and any hash value H. The following definition of EHPS essentially repeats the definition from [1], except minor changes to fit our requirement. Formally, EHPS are a tuple of algorithms defined as follows:

- $\mathrm{Setup}(1^\lambda)$: take the security parameter λ as input and output a common reference string (CRS) crs together with a trapdoor τ.
- $\mathrm{HashKG}(crs)$: take the CRS crs as input and output a hashing key hk.
- $\mathrm{ProjKG}(hk, crs)$: take the hashing key hk and the CRS crs as inputs, generate the projection key hp.
- $\mathrm{Hash}(hk, crs, x)$: take the hashing key hk, the CRS crs and any word $x \in X$ as inputs, output the hash value H.
- $\mathrm{ProjHash}(hp, crs, x, w)$: take a word $x \in L$, the projection key hp, and the witness w as inputs, output the hash value H.
- $\mathrm{SimKG}(crs, \tau, x)$: take as input crs, τ and a word $x \notin L$, output a projection key hp and an explainability key $expk$. For our purpose, we additionally require the hashing key hk to be part of the explainability key $expk$. We note that the first construction in [1] satisfy this property.

– Explain$(hp, crs, x, H, expk)$: take as input the projection key hp, the CRS crs, the word $x \notin L$, the hash value H and the explainability key $expk$, output the corresponding hashing key hk.

For any $(crs, \tau) \leftarrow \text{Setup}(1^\lambda)$, EHPS should satisfy the following properties:

– Explainability correctness: we require that $hp = \text{ProjKG}(hk, crs)$ and $H = \text{Hash}(hk, crs, x)$ hold for any hash value H and any $x \notin L$, if $(hp, expk) \leftarrow \text{SimKG}(crs, \tau, x)$ and $hk \leftarrow \text{Explain}(hp, crs, x, H, expk)$.
– Indistinguishability: for any $x \notin L$, the following two distributions are statistically indistinguishable:

$$\{(hk, hp) | H \leftarrow \Pi; (hp, expk) \leftarrow \text{SimKG}(crs, \tau, x); hk \leftarrow \text{Explain}(hp, crs, x, H, expk)\}$$
$$\approx_s \{(hk, hp) | hk \leftarrow \text{HashKG}(crs); hp \leftarrow \text{ProjKG}(hk, crs)\}.$$

6.3 IND-RSO-CPA Secure PKE from EHPS

Hazay *et al.* [19] demonstrated that HPS implies tweaked NCER, and we show that tweaked NCER can also be constructed from EHPS and it has multiple advantages. The tweaked NCER scheme we constructed is a tuple of five algorithms $(\text{tGen}, \text{tEnc}, \text{tEnc}^*, \text{tDec}, \text{tOpen})$ as follows:

– tGen: take the security parameter λ as input, output the global parameter crs and the trapdoor τ. Invoke HashKG and ProjKG algorithms to obtain $hk \leftarrow \text{HashKG}(crs)$ and $hp \leftarrow \text{ProjKG}(hk, crs)$ respectively, finally output the public key hp and secret key hk.
– tEnc: take the public key hp and the plaintext m as inputs, choose a random $x \in L$ together with the witness w; compute $e = \text{ProjHash}(hp, x, w) \oplus m$, and output the ciphertext (x, e).
– tDec: take the secret key hk and the ciphertext (x, e) as inputs, output the plaintext $m = e \oplus \text{Hash}(hk, x)$.
– tEnc^*: take the secret key hk and the plaintext m as inputs, choose a random $x^* \in X \setminus L$; compute $e^* = \text{Hash}(hk, x^*) \oplus m$, and output the fake ciphertext (x^*, e^*).
– tOpen: take as input the trapdoor τ, the secret key hk and the public key hp, fake ciphertext (x^*, e^*) and plaintext m; let $expk = (\tau, hk)$ and let $H = e^* \oplus m$, then invoke the $\text{Explain}(hp, crs, x^*, H, expk)$ algorithm, and finally output a secret key hk^*.

The ciphertexts generated by tEnc algorithm are real ciphertexts, while those generated by tEnc^* algorithm are fake ciphertexts. Furthermore, the decryption of a fake ciphertext e^* is the plaintext m. Our scheme also satisfies the following properties:

– Completeness: it can be guaranteed by the projective property of EHPS.
– Security: according to the indistinguishability property of EHPS, the real and the fake ciphertexts are indistinguishable. In addition, the smoothness property of EHPS guarantees that $\text{Hash}(hk, x^*)$ is randomly distributed, for $x^* \in X \setminus L$. Hence, according to the explainability property, for a given m, there exists a hk^* corresponding to hp such that $\text{Hash}(hk^*, x^*) = e^* \oplus m$.

Compared with the construction of Hazay *et al.* [19], our construction mainly change the tOpen algorithm. Specifically, the tOpen algorithm in Hazay *et al.*'s [19] scheme allows an unbounded algorithm to find the right hk^* by searching exhaustively, while our tOpen algorithm is more likely to obtain hk^* efficiently by referring to the work in [1]. Abdalla *et al.* [1] proposed two schemes for constructing Explain$(\cdot)$ in the tOpen algorithm. The first one is running in $O(2^v)$ time, where v is the bit length of the hash value. If v is a polynomial in logarithmic space, then the Explain$(\cdot)$ algorithm is running in polynomial time. The second one is more efficient, where the Explain$(\cdot)$ algorithm runs in constant time but can only be constructed in a specific framework. We note that if we can construct a Explain$(\cdot)$ algorithm that runs in probabilistic polynomial time, then it is possible to construct a NCER scheme and thereby a SIM-RSO-CPA secure PKE scheme.

Remarks. The SimKG algorithm of EHPS is implicitly used in the construction of our tweaked NCER scheme. According to the explainability correctness of EHPS, the SimKG algorithm will generate the same public key hp as the output of ProjKG algorithm, and the explainability key $expk$ is consisted of a trapdoor τ and a secret key hk corresponding to the public key hp.

7 Conclusion

In this paper, we study PKE of security against SO attacks, which is an important topic in PKE schemes. In particular, first, we define a new cryptographic primitive called tweaked lossy encryption, mainly inspired by ABME, dual-mode commitments, simulatable DEMs, and instance-dependent cryptographic primitives; and we further show that tweaked lossy encryption satisfies the SIM-SSO-CPA security. Second, we provide a generic construction of tweaked lossy encryption from extractable Σ-protocol; in addition, we propose two instantiations of the scheme based on dual-mode commitments and Twin-Cramer-Shoup scheme respectively; and we offer solid proofs of the two instantiations satisfying our definition of tweaked lossy encryption. Finally, we further propose a generic scheme with IND-RSO-CPA security based on EHPS. Our work in this paper provides an insightful view about designing PKE schemes with SO security using cryptographic primitives, such as extractable Σ-protocol and EHPS.

Acknowledgments. The authors thank China Scholarship Council for supporting Shuang Hu's (CSC Student No. 201706230130) work, and she is a visiting student at Virginia Commonwealth University from 2017 to 2019. The authors would like to also thank Dingding Jia for helpful discussions and advice, as well as the anonymous reviewers for their invaluable comments and suggestions.

References

1. Abdalla, M., Benhamouda, F., Pointcheval, D.: Removing erasures with explainable hash proof systems. In: Fehr, S. (ed.) PKC 2017. LNCS, vol. 10174, pp. 151–174. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54365-8_7
2. Bellare, M., Dowsley, R., Waters, B., Yilek, S.: Standard security does not imply security against selective-opening. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 645–662. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_38
3. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_1
4. Böhl, F., Hofheinz, D., Kraschewski, D.: On definitions of selective opening security. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 522–539. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30057-8_31
5. Boyen, X., Li, Q.: All-but-many lossy trapdoor functions from lattices and applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10403, pp. 298–331. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_11
6. Canetti, R., Dwork, C., Naor, M., Ostrovsky, R.: Deniable encryption. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 90–104. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052229
7. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: Miller, G.L. (ed.) Twenty-Eighth Annual ACM Symposium on the Theory of Computing. STOC 1996, pp. 639–648. ACM (1996)
8. Cash, D., Kiltz, E., Shoup, V.: The twin diffie-hellman problem and applications. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_8
9. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Comput. **33**(1), 167–226 (2003)
10. Damgård, I., Groth, J.: Non-interactive and reusable non-malleable commitment schemes. In: Larmore, L.L., Goemans, M.X. (eds.) Proceedings of the 35th Annual ACM Symposium on Theory of Computing 2003, pp. 426–437. ACM (2003)
11. Damgård, I., Nielsen, J.B.: Improved non-committing encryption schemes based on a general complexity assumption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 432–450. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44598-6_27
12. Deng, Y., Lin, D.: Instance-dependent verifiable random functions and their application to simultaneous resettability. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 148–168. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72540-4_9
13. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: 40th Annual Symposium on Foundations of Computer Science. FOCS 1999, pp. 523–534 (1999)
14. Fehr, S., Hofheinz, D., Kiltz, E., Wee, H.: Encryption schemes secure against chosen-ciphertext selective opening attacks. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 381–402. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_20

15. Fuchsbauer, G., Heuer, F., Kiltz, E., Pietrzak, K.: Standard security does imply security against selective opening for markov distributions. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9562, pp. 282–305. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49096-9_12
16. Fujisaki, E.: All-but-many encryption. A new framework for fully-equipped UC commitments. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8874, pp. 426–447. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45608-8_23
17. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing. STOC 1985, pp. 291–304. ACM, New York (1985)
18. Hara, K., Kitagawa, F., Matsuda, T., Hanaoka, G., Tanaka, K.: Simulation-based receiver selective opening CCA secure PKE from standard computational assumptions. In: Catalano, D., De Prisco, R. (eds.) SCN 2018. LNCS, vol. 11035, pp. 140–159. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98113-0_8
19. Hazay, C., Patra, A., Warinschi, B.: Selective opening security for receivers. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 443–469. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_19
20. Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy encryption: constructions from general assumptions and efficient selective opening chosen ciphertext security. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 70–88. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_4
21. Heuer, F., Jager, T., Kiltz, E., Schäge, S.: On the selective opening security of practical public-key encryption schemes. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 27–51. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_2
22. Heuer, F., Jager, T., Schäge, S., Kiltz, E.: Selective opening security of practical public-key encryption schemes. IET Inf. Secur. **10**(6), 304–318 (2016)
23. Heuer, F., Poettering, B.: Selective opening security from simulatable data encapsulation. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 248–277. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_9
24. Hofheinz, D.: All-but-many lossy trapdoor functions. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 209–227. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_14
25. Hofheinz, D., Jager, T., Rupp, A.: Public-key encryption with simulation-based selective-opening security and compact ciphertexts. In: Hirt, M., Smith, A. (eds.) TCC 2016. LNCS, vol. 9986, pp. 146–168. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53644-5_6
26. Hofheinz, D., Rao, V., Wichs, D.: Standard security does not imply indistinguishability under selective opening. In: Hirt, M., Smith, A. (eds.) TCC 2016. LNCS, vol. 9986, pp. 121–145. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53644-5_5
27. Hofheinz, D., Rupp, A.: Standard versus selective opening security: separation and equivalence results. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 591–615. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_25
28. Huang, Z., Lai, J., Chen, W., Au, M.H., Peng, Z., Li, J.: Simulation-based selective opening security for receivers under chosen-ciphertext attacks. Des. Codes Cryptogr. 1–27 (2018)

29. Huang, Z., Liu, S., Qin, B.: Sender-equivocable encryption schemes secure against chosen-ciphertext attacks revisited. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 369–385. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_23
30. Jia, D., Lu, X., Li, B.: Receiver selective opening security from indistinguishability obfuscation. In: Dunkelman, O., Sanadhya, S.K. (eds.) INDOCRYPT 2016. LNCS, vol. 10095, pp. 393–410. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49890-4_22
31. Jia, D., Lu, X., Li, B.: Constructions secure against receiver selective opening and chosen ciphertext attacks. In: Handschuh, H. (ed.) CT-RSA 2017. LNCS, vol. 10159, pp. 417–431. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52153-4_24
32. Libert, B., Sakzad, A., Stehlé, D., Steinfeld, R.: All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10403, pp. 332–364. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_12
33. Lindell, Y.: An efficient transform from sigma protocols to NIZK with a CRS and non-programmable random oracle. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9014, pp. 93–109. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_5
34. Liu, S., Paterson, K.G.: Simulation-based selective opening CCA security for PKE from key encapsulation mechanisms. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 3–26. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_1
35. Micciancio, D., Ong, S.J., Sahai, A., Vadhan, S.: Concurrent zero knowledge without complexity assumptions. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 1–20. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_1
36. Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: the non-committing encryption case. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 111–126. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_8
37. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_31
38. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 187–196 (2008)