Selective Opening Security for Receivers

Carmit Hazay [*] Arpita Patra [†] Bogdan Warinschi [‡]

Abstract

In a selective opening (SO) attack an adversary breaks into a subset of honestly created ciphertexts and tries to learn information on the plaintexts of some untouched (but potentially related) ciphertexts. Contrary to intuition, standard security notions do not always imply security against this type of adversary, making SO security an important standalone goal. In this paper we study *receiver security*, where the attacker is allowed to obtain the decryption keys corresponding to some of the ciphertexts.

First we study the relation between two existing security definitions, one based on simulation and the other based on indistinguishability, and show that the former is strictly stronger. We continue with feasibility results for both notions which we show can be achieved from (variants of) non-committing encryption schemes. In particular, we show that indistinguishability-based SO security can be achieved from a tweaked variant of non-committing encryption which, in turn, can be instantiated from a variety of basic, well-established, assumptions. We conclude our study by showing that SO security is however strictly weaker than all variants of non-committing encryption that we consider, leaving potentially more efficient constructions as an interesting open problem.

Keywords: Selective Opening Attacks, Encryption Schemes, Non-committing Encryption

[*] Faculty of Engineering, Bar-Ilan University, Israel. Email: carmit.hazay@biu.ac.il.

[†] Dept. of Computer Science & Automation, Indian Institute of Science, India. Email: arpita@csa.iisc.ernet.in.

[‡] Department of Computer Science, University of Bristol, United Kingdom. Email: csxbw@bristol.ac.uk.

1 Introduction

Security notions for encryption come in many forms that reflect different attacker goals (e.g. one-wayness, indistinguishability for plaintexts or non-malleability of ciphertexts), variations in possible attack scenarios (e.g. chosen plaintext or ciphertext attacks) and definitional paradigms (e.g. through games or simulation). A class of attacks motivated by practical considerations are those where the adversary may perform *selective openings* (SO). Here, an adversary is allowed to break into a subset of honestly created ciphertexts leaving untouched other (potentially related) ciphertexts.

This attack scenario was first identified in the context of adaptively secure multi-party computation (MPC) where communication is over encrypted channels visible to the adversary. The standard trust model considers an adversary who, based on the information that he sees, can decide to corrupt parties and learn their internal state. In turn, this may allow the attacker to determine the parties' long term secret keys and/or the randomness used to create the ciphertexts. The broader context of Internet communication also naturally gives rise to SO attacks. Attackers that access and store large amount of encrypted internet traffic are a reality, and getting access to the internal states of honest parties can be done by leveraging design or implementation weaknesses of deployed systems. For example the Heartbleed attack allowed a remote party to extract (among other things) the encryption keys used to protect OpenSSL connections.

Security against SO attacks comes in several distinct flavors. Depending on the attack scenario, we distinguish two settings that fall under the general idea of SO attacks. In *sender security*, we have n senders and one receiver. The receiver holds a secret key relative to a public key known to all the senders. The senders encrypt messages for the receiver and the adversary is allowed to corrupt some of the senders (and learn the messages and randomness underlying some of the ciphertexts). The concern is that the messages sent by uncorrupted senders stay secret. The second scenario deals with *receiver security*. Here we consider one sender and n receivers who hold independently generated public and secret keys. The attacker is allowed to corrupt some of the receivers (and learn the secret keys that decrypt some of the observed ciphertexts). Security in this setting is concerned with the messages received by uncorrupted receivers. For each of these settings, security can be defined using either the standard indistinguishability paradigm or simulation-based definitions. Importantly, both scenarios capture realistic attacks in secure computation where usually every party acts as either a sender or a receiver at some point of time during a protocol execution.

Since most of the existent encryption schemes have been analyzed w.r.t. traditional notions of security (e.g. indistinguishability under chosen plaintext or chosen ciphertext attacks (**ind-cpa**, **ind-cca**)), a central question in this area is to understand how security against SO attacks relates to the established definitions. Despite compelling intuition that the only information that an adversary obtains is what it can glean from the opened plaintexts, progress towards confirming or disproving this conjecture has been rather slow. Perhaps the most interesting and surprising results are due to Bellare et al. [BHY09, BDWY12] who showed that selective sender security as captured via *simulation* based definitions is strictly stronger than indistinguishability under chosen plaintext attacks [GM84] (denoted by **ind-cpa** security). The gap between standard notions of security and SO security is uncomfortable: while SO attacks may naturally occur we do not have a clear understanding of the level of security that existing constructions offer nor do we have many ideas on how to achieve security against such attacks.

In this paper we study receiver security. This setting is less studied than sender security yet it corresponds to more plausible attacks (e.g. the Heartbleed attack). In a nutshell, we clarify the relation between various security notions for receiver security and propose novel constructions. Before we describe our contributions in detail we overview existing work in the area and take this opportunity to introduce more carefully the different security notions of SO security.

1.1 Related Work

Selective opening attacks were first introduced in [DNRS03] in the context of commitment schemes. In the context of encryption schemes, the first rigorous definitions were proposed by Bellare, Hofheinz and Yilek [BHY09]. They studied SO security for public key encryption (PKE), for both the receiver and the sender settings and for each setting proposed two types of definitions, indistinguishability-based and simulation-based ones. Very roughly, the indistinguishability-based definition (denoted by **ind-so**) requires that an adversary that sees a vector of ciphertexts cannot distinguish the true plaintexts of the unopened ciphertexts from independently sampled plaintexts. This is required even with access to the randomness used for generating the opened ciphertexts (in the sender corruption setting), or with access to the secret keys that decrypt the opened ciphertexts (in the receiver corruption setting). This definition requires messages to come from a distribution that is *efficiently resamplable*. A stronger security variant that does not restrict the message distribution called *full* **ind-so** has been introduced later by Böhl, Hofheinz and Kraschewski [BHK12]. The simulation based notion (denoted by **sim-so**) is reminiscent of the definitional approach from [DNRS03] and requires computational indistinguishability between an idealized execution and the real one.

The first feasibility results for security against selective opening attacks are for the sender setting and leverage an interesting relation with lossy encryption: a lossy PKE implies **ind-so** for sender security [BHY09]. Furthermore, if the PKE scheme has an efficient opening algorithm of ciphertexts, then the scheme also satisfies **sim-so** security. The work of Hemenway et al. [HLOV11] shows that lossy (and therefore **ind-so**) PKE can be constructed based on several generic cryptographic primitives.

For primitives that benefit from multiple security notions, a central question is to understand how these notions relate to each other. This type of results are important as they clarify the limitations of some of the notions and enable trade-offs between security and efficiency (to gain efficiency, a scheme with weaker guarantees may be employed, if the setting allows it). The relation between traditional security notions of encryption and security against selective opening attacks was a long-standing open problem that was solved by Bellare et al. [BDWY12]. Their result is that standard **ind-cpa** security *does not* imply **sim-so** (neither in the sender nor in the receiver setting). There is no fully satisfactory result concerning the relation between **ind-cpa** and **ind-so**. Here, the best result is that these two notions imply each other in the generic group model [HR14] and that for the chosen-chiphertext attacks variant (CCA) the two notions are distinct.

Relations between the different notions for selective opening have mainly been studied in the sender setting. Böhl et al. establish that full **ind-so** and **sim-so** are incomparable. Recently, [ROV14] introduced an even stronger variant of the full **ind-so** definition, and showed that many **ind-cpa**, **ind-so** and **sim-so** secure encryption schemes are insecure according to their new notion. They further showed that **sim-so** definition does not imply lossy encryption even without efficient openability. Finally, SO security has been considered for CCA attacks in [FHKW10, HLQ13] and in the identity-based encryption scenario in [BWY11].

1.2 Our Contribution

With only two exceptions [BHY09, BDWY12] prior work on SO security has addressed mainly the sender setting. We concentrate on the receiver setting. Though theoretically the feasibility for SO security for the receiver is implied by the existence of non-committing encryption schemes [CFGN96, Nie02, DN00, CDSMW09], the state of the art constructions still leave many interesting open problems in terms of relations between notions and feasibility results. This is the focus of this work.

For relation between notions, similarly to prior separating results in the SO setting [BHK12, HR14, ROV14], we demonstrate the existence of a separating scheme that is based on generic assumptions and can be instantiated under various concrete assumptions. For constructions, we find it useful to leverage the close relation between (variants of) non-committing encryption and security under selective opening attacks. For example, we show that **ind-so** security follows from a tweaked variant of non-committing encryption which, in turn, we show how to instantiate from a variety of standard assumptions. Interestingly, we also show a separation between SO security and non-committing encryption (which leaves open the question of potentially more efficient constructions that meet the former notion but not the latter). Below, we elaborate on our results in details.

Notation-wise, we denote the indistinguishability and simulation-based definitions in the receiver setting by **rind-so** and **rsim-so**, respectively. For the corresponding notions in the sender setting we write **sind-so** and **ssim-so**, respectively. That is, we prepend “s” or “r” to indicate if the definition is for sender security or receiver security.

The relation between rind-so and rsim-so. First, we study the relation between the indistinguishability and simulation-based security notions in the receiver setting. We establish that the **rind-so** notion is strictly weaker (and therefore easier to realize) than the notion of **rsim-so**, by presenting a concrete pubic key scheme that meets the former but not the latter level of security. Loosely speaking, a ciphertext includes a commitment to the plaintext together with encryptions of the opening information of this commitment (namely, the plaintext and the corresponding randomness). We then prove that when switching to an alternative fake mode the hiding properties of our building blocks (commitment and encryption schemes) imply that the ciphertext does not contain any information about the plaintext. Nevertheless, simulation always fails since it would require breaking the binding property of the commitment. Applying the observation that **rsim-so** implies **rind-so** security,[1] we obtain the result that **rind-so** is strictly weaker.

In more details, our separating scheme is built from a commitment scheme and a primitive called non-committing encryption for the receiver (NCER) [CHK05] that operates in two indistinguishable ciphertexts modes: valid and fake, where a fake ciphertext can be decrypted into any plaintext using a proper secret key. This property is referred to as *secret key equivocation* and is implied by the fact that fake ciphertexts are lossy which, in turn, implies **rind-so** security. Specifically, the security of our scheme implies that:

Theorem 1.1. (Informal) *There exists a PKE that is* **rind-so** *secure but is not* **rsim-so** *secure.*

Somewhat related to our work, [BDWY12] proved that the standard **ind-cpa** security does not imply **rsim-so** security via the notion of *decryption verifiability* – the idea that it is hard to decrypt a ciphertext into two distinct messages (even using two different secret keys). Specifically, [BDWY12] showed that any **ind-cpa** secure PKE that is decryption verifiable cannot be **rsim-so** secure. Compared with their result, our result implies that **rsim-so** security is strictly stronger than **rind-so** security (which may turn out to be stronger than **ind-cpa** security).

The feasibility of rind-so and rsim-so. We recall that in the sender setting, the notions **sind-so** and **ssim-so** are achievable from lossy encryption and lossy encryption with efficient openability.[2] We identify a security notion (and a variant) which plays for receiver security the role that lossy encryption plays in sender security. Specifically, we prove that NCER implies **rsim-so** and that a variant of NCER, which we

[1] This can be derived from the fact that the adversary’s view is identical for any two simulated executions with different sets of unopened messages, as the simulator never gets to see these messages.

[2] Recall that a lossy encryption scheme is a public key encryption with the additional ability to generate fake indistinguishable public keys so that a fake ciphertext (that is generated using a fake public key) is lossy and is a non-committing ciphertext with respect to the plaintext. A lossy encryption implies the existence of an opening algorithm (possibly inefficient) that can compute a randomness for a given fake ciphertext and a message.

refer as *tweaked NCER* (formally defined in Section 3.3), implies **rind-so**. Loosely speaking, the security of tweaked NCER is formalized as follows. Similarly to NCER, tweaked NCER has the ability to create fake ciphertexts that are computationally indistinguishable from real ciphertexts. Nevertheless, while in NCER a fake ciphertext can be efficiently decrypted to any plaintext (by producing a matching secret key), in tweaked NCER a fake ciphertext can only be efficiently decrypted to a concrete predetermined plaintext. Informally, our results are captured by the following theorem:

Theorem 1.2. (Informal) *Assume the existence of tweaked NCER and NCER, then there exist PKE schemes that are* **rind-so** *and* **rsim-so** *secure, respectively.*

Interestingly, we show that the converse implications do not hold. That is, a **rsim-so** secure PKE is not necessarily a tweaked NCER or a NCER. This further implies that a **rind-so** secure PKE is not necessarily a tweaked NCER or NCER. This result is reminiscent of the previous result that **sim-so** and **rind-so** secure PKE do not imply lossy encryption even without efficient openability [ROV14].

Our separating scheme is based on an arbitrary key-simulatable PKE schemes. Intuitively, in such schemes, it is possible to produce a public key without sampling the corresponding secret key. The set of obliviously sampled public keys may be larger than the the set of public keys sampled together with their associated secret key, yet it is possible to prove a public key sampled along with a secret key as one sampled without. In these schemes we also require that the two type of keys are also computationally indistinguishable. Our proof holds for the case that the set of obliviously sampled keys is indeed larger, so that not every obliviously sampled public key can be explained to possess a secret key. In summary, we prove that:

Theorem 1.3. (Informal) *Assume the existence of key-simulatable PKE, then there exists a PKE scheme that is* **rsim-so** *secure but is neither tweaked NCER nor NCER.*

The constructions that we present show that **rsim-so** (and **rind-so**) security can be achieved under the same assumptions as key-simulatable PKE – there are results that show that the latter can be constructed from a variety of hardness assumptions such as Decisional Diffie-Hellman (DDH) and Decisional Composite Residuosity (DCR). They also show that we can construct schemes from any hardness assumption that implies simulatable PKE [DN00] (where both public keys and ciphertexts can be obliviously sampled).

Realizing tweaked NCER. Finally, we demonstrate the broad applicability of this primitive and show how to construct it from various important primitives: key-simultable PKE, two-round honest-receiver statistically-hiding $\binom{2}{1}$ oblivious transfer (OT) and hash proof systems (HPS). We stress that it is not known how to build NCER under these assumptions (or any other generic assumption), which implies that tweaked NCER is much easier to realize. In addition, we prove that the two existing NCER schemes [CHK05] with security under the DDH and DCR hardness assumptions imply the tweaked NCER notion, where surprisingly, the former construction that is a secure NCER for only polynomial-size message spaces, is a tweaked NCER for exponential-size message spaces (this further hints that tweaked NCER may be constructed more efficiently than NCER). These results imply that tweaked NCER (and thus **rind-so**) can be realized based on DDH, DCR, RSA, factoring and learning with errors (LWE) hardness assumptions.

Our results are summarized in Fig. 1.

The relation between sind-so and ssim-so. As a side result, we study the relation between the indistinguishability and simulation based security definitions in the sender setting. We show that **sind-so** is strictly

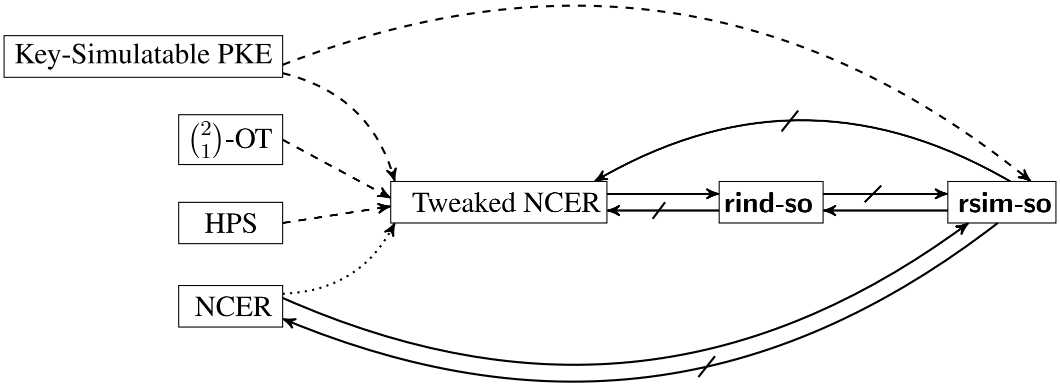

Figure 1: The arrows can be read as follows: *solid arrows* denote implication, *crossed arrows* denote counterexamples, *dashed arrows* denote assumption-wise implication and *dotted arrows* denote implication with respect to concrete instances (where the implication may not hold in general). The implication of receiver indistinguishability security by simulation security is a known result.

weaker than the notion of **ssim-so** by presenting a concrete public key scheme that meets the former but not the latter level of security. Our separating scheme is built using the two primitives lossy public key encryption and commitment scheme. We exploit the hiding properties of these building blocks to prove that our scheme implies **sind-so** security. On the other hand, simulation always fails since it implies breaking the binding property of the commitment scheme. Informally, we prove the following theorem:

Theorem 1.4. (Informal) *There exists a PKE that is* **sind-so** *secure but is not* **ssim-so** *secure.*

We stress that this was already demonstrated indirectly in [BY09] (by combining two separation results). Here we design a concrete counter example to demonstrate the same in a simpler manner. A similar result has been shown for *full* **ind-so** and **sim-so** in [BHK12], demonstrating that these definitions do *not* imply each other in the sender setting.

To sum up, we study the different levels for receiver security in the presence of SO attacks. We clarify the relation between these notions and provide constructions that meet them using the close conceptual relation between SO security and non-committing encryption. From a broader perspective, our results position more precisely SO security for the receiver in the spectrum of security notions for encryption.

2 Preliminaries

Basic notations. For $x, y \in \mathbb{N}$ with $x < y$, let $[x] := \{1, \ldots, x\}$ and $[x, y] := \{x, x+1, \ldots, y\}$. We denote the computational security parameter by k and statistical security parameter by s. A function $\mu(\cdot)$ is *negligible* in security parameter κ if for every polynomial $p(\cdot)$ there exists a value N such that for all $\kappa > N$ it holds that $\mu(k) < \frac{1}{p(\kappa)}$, where κ is either k or s. For a finite set S, we denote by $s \leftarrow S$ the process of sampling s uniformly. For a distribution X, we denote by $x \leftarrow X$ the process of sampling x from X. For a deterministic algorithm A, we write $a \leftarrow A(x)$ the process of running A on input x and assigning a the result. For a randomized algorithm A, we write $a \leftarrow A(x; r)$ the process of running A on input x and randomness r and assigning y the result. At times we skip r in the parenthesis to avoid mentioning it explicitly. We write PPT for probabilistic polynomial-time. For a PKE (or commitment) scheme C, we use the notation $\mathcal{M}_C$ and respectively $\mathcal{R}_C$ to denote the input and the randomness space of the encryption (or commitment) algorithm of C. We use bold fonts to denote vectors. If $\mathbf{m}$ is an n dimensional vector, we write $\mathbf{m}_i$ for the i-th entry in $\mathbf{m}$; if $\mathcal{I} \subseteq [n]$ is a set of indices we write $\mathbf{m}_\mathcal{I}$ for the vector of dimension $|\mathcal{I}|$ obtained by projecting $\mathbf{m}$ on the coordinates in $\mathcal{I}$.

2.1 Public Key Encryption

A public key encryption (PKE) scheme PKE with message space $\mathcal{M}$ consists of three PPT algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. The key generation algorithm $\mathsf{Gen}(1^k)$ outputs a public key pk and a secret key sk. The encryption algorithm $\mathsf{Enc}_{pk}(m; r)$ takes pk and a message $m \in \mathcal{M}$ and randomness $r \in \mathcal{R}$, and outputs a ciphertext c. The decryption algorithm $\mathsf{Dec}_{sk}(c)$ takes sk and a ciphertext c and outputs a message m. For correctness, we require that $m = \mathsf{Dec}_{sk}(c)$ for all $m \in \mathcal{M}$ and all $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$ and all $c \leftarrow \mathsf{Enc}_{pk}(m)$. The standard notion of security for PKE is indistinguishability under chosen plaintext attacks, denoted by **ind-cpa** [GM84] (and the corresponding experiment is denoted as $\mathbf{Exp}^{\textsf{ind-cpa}}_{\mathsf{PKE}}$). *As a general remark, we note that whenever we refer to a secret key, we refer to the randomness used to generate it by the key generation algorithm.*

2.2 Selective Opening Security

Depending on the attack scenario, we distinguish two settings that fall under the general idea of SO attacks. In *sender security*, we have n senders and one receiver. The receiver holds a secret key relative to a public key known to all senders. The senders send messages to the receiver and the adversary is allowed to corrupt some of the senders (and learn the messages and randomness underlying some of the ciphertexts). The concern is that the messages sent by uncorrupted users stay secret. The second scenario deals with *receiver security*. Here we consider one sender and n receivers who hold independently generated public and secret keys. The attacker is allowed to learn the secret keys of some of the receivers. Security is concerned with the messages received by uncorrupted receivers.

For each of these settings we consider two definitions: (1) an indistinguishability based definition [BHY09] and (2) a simulation based definition that follows from [BHY09] which, in turn, builds on the one proposed by [DNRS03] in the context of commitments. Indistinguishability-based definitions require that an adversary that sees a vector of ciphertexts cannot distinguish the true plaintexts of the ciphertexts from independently sampled plaintexts, even in the presence of the randomness used for generating the opened ciphertexts (in the sender corruption setting), or the secret keys that decrypt the opened ciphertexts (in the receiver corruption setting). The indistinguishability based definitions use the notion of *efficiently resamplable* message distributions which we recall next following [BHK12].

Definition 2.1 (Efficiently resamplable distribution). *Let $n = n(k) > 0$ and let* Dist *be a joint distribution over $\left(\{0,1\}^k\right)^n$. We say that* Dist *is efficiently resamplable if there is a PPT algorithm* $\mathsf{Resamp}_{\mathsf{Dist}}$ *such that for any $\mathcal{I} \subseteq [n]$ and any partial vector $\mathbf{m}'_\mathcal{I} \in \left(\{0,1\}^k\right)^{|\mathcal{I}|}$,* $\mathsf{Resamp}_{\mathsf{Dist}}(\mathbf{m}'_\mathcal{I})$ *returns a vector $\mathbf{m}$ sampled from* $\mathsf{Dist}|\mathbf{m}'_\mathcal{I}$*, i.e. $\mathbf{m}'$ is sampled from* Dist *conditioned on $\mathbf{m}_\mathcal{I} = \mathbf{m}'_\mathcal{I}$.*

Below, we recall indistinguishability and simulation based definitions for security in the presence of selective opening attacks[3]. We present the definitions for sender and receiver security. To avoid heavy

[3] We remark that a stronger security notion that does not does require efficient resemplability is possible, but no constructions that satisfy this stronger notion are known.

notation we follow the following conventions when naming the security notions: we use "ind" or "sim" to indicate if the definition is indistinguishability-based or simulation-based, and prepend "s" or "r" to indicate if the definition is for sender security or receiver security; we keep "so" in the name of the notion to indicate that we deal with selective opening attacks. We also note that we consider chosen plaintext attacks only, but avoid showing this explicitly in the names of the security notions.

Definition 2.2 (Indistinguishability based SO security). *For a PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, a polynomially bounded function* $n = n(k) > 0$ *and a stateful PPT adversary* A*, consider the following two experiments; the left experiment corresponds to sender corruptions, whereas, the right experiment corresponds to receiver corruptions.*

```
Experiment. $\mathbf{Exp}^{\mathsf{sind\text{-}so}}_{\mathsf{PKE}}(\mathsf{A}, k)$
    $b \leftarrow \{0,1\}$
    $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
    $(\mathsf{Dist}, \mathsf{Resamp}_{\mathsf{Dist}}, \mathrm{state}_1) \leftarrow \mathsf{A}(pk)$
    $\mathbf{m} := (m_i)_{i \in [n]} \leftarrow \mathsf{Dist}$
    $\mathbf{r} := (r_i)_{i \in [n]} \leftarrow \mathcal{R}^n_{\mathsf{PKE}}$
    $\mathbf{e} := (e_i)_{i \in [n]} \leftarrow (\mathsf{Enc}_{pk}(m_i; r_i))_{i \in [n]}$
    $(\mathcal{I}, \mathrm{state}_2) \leftarrow \mathsf{A}(\mathbf{e}, \mathrm{state}_1)$
    $\mathbf{m}' \leftarrow \mathsf{Resamp}(\mathbf{m}_{\mathcal{I}})$
    $\mathbf{m}^* = \mathbf{m}$ if $b = 0$, else $\mathbf{m}^* = \mathbf{m}'$
    $b' \leftarrow \mathsf{A}(\mathbf{r}_{\mathcal{I}}, \mathbf{m}^*, \mathrm{state}_2)$,
    Return 1 if $b = b'$, and 0 otherwise.
```

```
Experiment. $\mathbf{Exp}^{\mathsf{rind\text{-}so}}_{\mathsf{PKE}}(\mathsf{A}, k)$
    $b \leftarrow \{0,1\}$
    $(\mathbf{pk}, \mathbf{sk}) := (pk_i, sk_i) \leftarrow (\mathsf{Gen}(1^k))_{i \in [n]}$
    $(\mathsf{Dist}, \mathsf{Resamp}_{\mathsf{Dist}}, \mathrm{state}_1) \leftarrow \mathsf{A}(\mathbf{pk})$
    $\mathbf{m} := (m_i)_{i \in [n]} \leftarrow \mathsf{Dist}$
    $\mathbf{r} := (r_i)_{i \in [n]} \leftarrow \mathcal{R}^n_{\mathsf{PKE}}$
    $\mathbf{e} := (e_i)_{i \in [n]} \leftarrow (\mathsf{Enc}_{pk_i}(m_i; r_i))_{i \in [n]}$
    $(\mathcal{I}, \mathrm{state}_2) \leftarrow \mathsf{A}(\mathbf{e}, \mathrm{state}_1)$
    $\mathbf{m}' \leftarrow \mathsf{Resamp}(\mathbf{m}_{\mathcal{I}})$
    $\mathbf{m}^* = \mathbf{m}$ if $b = 0$, else $\mathbf{m}^* = \mathbf{m}'$
    $b' \leftarrow \mathsf{A}(\mathbf{sk}_{\mathcal{I}}, \mathbf{m}^*, \mathrm{state}_2)$
    Return 1 if $b = b'$, and 0 otherwise.
```

In the above experiments we only assume adversaries that are well-behaved in that they always output efficiently resamplable distributions together with resampling algorithms.

We say that PKE *is* **sind-so** *secure if for a well-behaved PPT* A *there exists a negligible function* $\mu = \mu(k)$ *such that*

$$\mathbf{Adv}^{\mathsf{sind\text{-}so}}_{\mathsf{PKE}}(\mathsf{A}, k) := 2 \left| \Pr[\mathbf{Exp}^{\mathsf{sind\text{-}so}}_{\mathsf{PKE}}(\mathsf{A}, k) = 1] - \frac{1}{2} \right| \leq \mu.$$

We say that PKE *is* **rind-so** *secure if for a well-behaved PPT* A *there exists a negligible function* $\mu = \mu(k)$ *such that*

$$\mathbf{Adv}^{\mathsf{rind\text{-}so}}_{\mathsf{PKE}}(\mathsf{A}, k) := 2 \left| \Pr[\mathbf{Exp}^{\mathsf{rind\text{-}so}}_{\mathsf{PKE}}(\mathsf{A}, k) = 1] - \frac{1}{2} \right| \leq \mu.$$

$\Pr\left[\mathbf{Exp}^{\mathsf{sind\text{-}so}}_{\mathsf{PKE}}(\mathsf{A}, k) = 1\right]$ *and* $\Pr\left[\mathbf{Exp}^{\mathsf{rind\text{-}so}}_{\mathsf{PKE}}(\mathsf{A}, k) = 1\right]$ *denote the* **winning probability** *of* A *in the respective experiments.*

Simulation based security is defined, as usual, by comparing an idealized execution with the real one. Again, we consider both sender and receiver security.

Definition 2.3 (Simulation based SO security). *For a PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, a polynomially bounded function* $n = n(k) > 0$*, a PPT adversary* A *and a PPT algorithm* S*, we define the following pairs of experiments.*

We say that PKE *is* **ssim-so** *secure iff for every PPT* A *there is a PPT algorithm* S*, a PPT distinguisher* D *with binary output and a negligible function* $\mu = \mu(k)$ *such that*

$$\mathbf{Adv}^{\mathsf{ssim\text{-}so}}_{\mathsf{PKE}}(\mathsf{D}, k) := \left| \Pr[1 \leftarrow \mathsf{D}(\mathbf{Exp}^{\mathsf{ssim\text{-}so\text{-}real}}_{\mathsf{PKE}}(\mathsf{A}, k))] - \Pr[1 \leftarrow \mathsf{D}(\mathbf{Exp}^{\mathsf{ssim\text{-}so\text{-}ideal}}_{\mathsf{PKE}}(\mathsf{S}, k))] \right| \leq \mu.$$

```
**Experiment.** $\mathbf{Exp}^{\textsf{ssim-so-real}}_{\textsf{PKE}}(\mathsf{A}, k)$
    $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
    $(\mathsf{Dist}, \mathrm{state}_1) \leftarrow \mathsf{A}(pk)$
    $\mathbf{m} := (m_i)_{i\in[n]} \leftarrow \mathsf{Dist}$
    $\mathbf{r} := (r_i)_{i\in[n]} \leftarrow \mathcal{R}^n_{\mathsf{PKE}}$
    $\mathbf{e} := (e_i)_{i\in[n]} \leftarrow (\mathsf{Enc}_{pk}(m_i; r_i))_{i\in[n]}$
    $(\mathcal{I}, \mathrm{state}_2) \leftarrow \mathsf{A}(\mathbf{e}, \mathrm{state}_1)$
    $\mathrm{output} \leftarrow \mathsf{A}(\mathbf{r}_{\mathcal{I}}, \mathbf{m}_{\mathcal{I}}, \mathrm{state}_2)$
    *Return* $(\mathbf{m}, \mathsf{Dist}, \mathcal{I}, \mathrm{output})$.
```

```
**Experiment.** $\mathbf{Exp}^{\textsf{ssim-so-ideal}}_{\textsf{PKE}}(\mathsf{S}, k)$
    $(\mathsf{Dist}, \mathrm{state}_1) \leftarrow \mathsf{S}(\cdot)$
    $\mathbf{m} := (m_i)_{i\in[n]} \leftarrow \mathsf{Dist}$
    $(\mathcal{I}, \mathrm{state}_2) \leftarrow \mathsf{S}(\mathrm{state}_1)$
    $\mathrm{output} \leftarrow \mathsf{S}(\mathbf{m}_{\mathcal{I}}, \mathrm{state}_2)$
    *Return* $(\mathbf{m}, \mathsf{Dist}, \mathcal{I}, \mathrm{output})$.
```

```
**Experiment.** $\mathbf{Exp}^{\textsf{rsim-so-real}}_{\textsf{PKE}}(\mathsf{A}, k)$
    $(\mathbf{pk}, \mathbf{sk}) := (pk_i, sk_i) \leftarrow (\mathsf{Gen}(1^k))_{i\in[n]}$
    $(\mathsf{Dist}, \mathrm{state}_1) \leftarrow \mathsf{A}(\mathbf{pk})$
    $\mathbf{m} := (m_i)_{i\in[n]} \leftarrow \mathsf{Dist}$
    $\mathbf{r} := (r_i)_{i\in[n]} \leftarrow \mathcal{R}^n_{\mathsf{PKE}}$
    $\mathbf{e} := (e_i)_{i\in[n]} \leftarrow (\mathsf{Enc}_{pk_i}(m_i; r_i))_{i\in[n]}$
    $(\mathcal{I}, \mathrm{state}_2) \leftarrow \mathsf{A}(\mathbf{e}, \mathrm{state}_1)$
    $\mathrm{output} \leftarrow \mathsf{A}(\mathbf{sk}_{\mathcal{I}}, \mathbf{m}_{\mathcal{I}}, \mathrm{state}_2)$
    *Return* $(\mathbf{m}, \mathsf{Dist}, \mathcal{I}, \mathrm{output})$.
```

```
**Experiment.** $\mathbf{Exp}^{\textsf{rsim-so-ideal}}_{\textsf{PKE}}(\mathsf{S}, k)$
    $(\mathsf{Dist}, \mathrm{state}_1) \leftarrow \mathsf{S}(\cdot)$
    $\mathbf{m} := (m_i)_{i\in[n]} \leftarrow \mathsf{Dist}$
    $(\mathcal{I}, \mathrm{state}_2) \leftarrow \mathsf{S}(\mathrm{state}_1)$
    $\mathrm{output} \leftarrow \mathsf{S}(\mathbf{m}_{\mathcal{I}}, \mathrm{state}_2)$
    *Return* $(\mathbf{m}, \mathsf{Dist}, \mathcal{I}, \mathrm{output})$.
```

We say that PKE *is* **rsim-so** *secure iff for every PPT* A *there is a PPT algorithm* S*, a PPT distinguisher* D *with binary output and a negligible function* $\mu = \mu(k)$ *such that*

$$\mathbf{Adv}^{\textsf{rsim-so}}_{\textsf{PKE}}(\mathsf{D}, k) := \left|\Pr[1 \leftarrow \mathsf{D}(\mathbf{Exp}^{\textsf{rsim-so-real}}_{\textsf{PKE}}(\mathsf{A}, k))] - \Pr[1 \leftarrow \mathsf{D}(\mathbf{Exp}^{\textsf{rsim-so-ideal}}_{\textsf{PKE}}(\mathsf{S}, k))]\right| \leq \mu.$$

Our definitions consider non-adaptive attacks, where the adversary corrupts the parties in one go. We note that all our results will remain unaffected even in the face of an adaptive adversary [BHK12].

3 Building Blocks

Our constructions employ a number of fundamental cryptographic building blocks as well as a new primitive which we denote by tweaked NCER. We describe them and their security definitions below.

3.1 Commitment Schemes

We require a non-interactive commitment scheme (NISHCOM) that is statistically hiding.

Definition 3.1 (NISHCOM). *A non-interactive commitment scheme* nisCom *consists of two algorithms* (nisCommit, nisOpen) *defined as follows. Given a security parameter k, message* $m \in \mathcal{M}_{\mathsf{nisCom}}$ *and random coins* $r \in \mathcal{R}_{\mathsf{nisCom}}$*, PPT algorithm* nisCommit *outputs commitment c. Given k, commitment c and message m, (possibly inefficient) algorithm* nisOpen *outputs r. We require the following properties:*

- Correctness. *We require that* $c = \mathsf{nisCommit}(m; r)$ *for all* $m \in \mathcal{M}_{\mathsf{nisCom}}$ *and* $r \leftarrow \mathsf{nisOpen}(c, m)$.
- Security. *A NISHCOM* nisCom *is* **stat-hide** *secure if commitments of two distinct messages are statistically indistinguishable. Specifically, for any* unbounded powerful *adversary* A*, there exists a negligible function* $\mu = \mu(s)$ *such that* $\mathbf{Adv}^{\textsf{stat-hide}}_{\mathsf{nisCom}}(\mathsf{A}, k) := |\Pr[1 \leftarrow \mathsf{A}(c_0)] - \Pr[1 \leftarrow \mathsf{A}(c_1)]| \leq \mu$ *for* $c_i \leftarrow \mathsf{nisCommit}(m_i)$ *for* $i \in \{0, 1\}$ *and* $m_0, m_1 \in \mathcal{M}_{\mathsf{nisCom}}$.

  A NISHCOM nisCom *is* **comp-bind** *secure if no commitment can be opened to two different messages in polynomial time. Specifically, the advantage* $\mathbf{Adv}^{\textsf{comp-bind}}_{\mathsf{nisCom}}(\mathsf{A}, k)$ *of* A *defined by* $\Pr[(m_0, r_0, m_1, r_1) \leftarrow$

$\mathsf{A}(k)\ :\ \mathsf{nisCommit}(m_0; r_0) = \mathsf{nisCommit}(m_1; r_1)]$ *(with the probability over the choice of the coins of* A*) is smaller than some negligible function* $\mu = \mu(k)$*.*

A NISHCOM nisCom *is called* **secure** *it is* $\{\textbf{stat-hide}, \textbf{comp-bind}\}$ *secure.*

3.2 Non-Committing Encryption for Receiver (NCER)

A non-committing encryption for receiver (NCER) [JL00, CHK05] is a PKE with the property that there is a way to generate fake ciphertexts which can then be decrypted (with the help of a trapdoor) to any plaintext. Intuitively, fake ciphertexts are generated in a lossy way so that the plaintext is no longer well defined given the ciphertext and the public key. This leaves enough entropy for the secret key to be sampled in a way that determines the desired plaintext. We continue with a formal definition of NCER and a security notion for it referred as **ind-ncer** security.

Definition 3.2 (NCER). *An NCER* nPKE *consists of five PPT algorithms* $(\mathsf{nGen}, \mathsf{nEnc}, \mathsf{nEnc}^*, \mathsf{nDec}, \mathsf{nOpen})$ *defined as follows. Algorithms* $(\mathsf{nGen}, \mathsf{nEnc}, \mathsf{nDec})$ *form a PKE. Given the public key* pk*, the fake encryption algorithm* nEnc^* *outputs a ciphertext* e^* *and a trapdoor* t*. Given the secret key* sk*, the public key* pk*, fake ciphertext* e^**, trapdoor* t *and plaintext* m*, algorithm* nOpen *outputs* sk^**.*

- Correctness. *We require that* $m = \mathsf{nDec}_{sk}(c)$ *for all* $m \in \mathcal{M}$*, all* $(pk, sk) \leftarrow \mathsf{nGen}(1^k)$ *and all* $c \leftarrow \mathsf{nEnc}_{pk}(m)$*.*
- Security. *An NCER scheme* nPKE *is* **ind-ncer** *secure if the real and fake ciphertexts are indistinguishable. Specifically, for a PPT adversary* A*, consider the experiment* $\mathbf{Exp}^{\textsf{ind-ncer}}_{\mathsf{nPKE}}$ *defined as follows.*

```
Experiment. $\mathbf{Exp}^{\textsf{ind-ncer}}_{\mathsf{nPKE}}(\mathsf{A}, k)$
    $b \leftarrow \{0, 1\}$
    $(pk, sk_0) \leftarrow \mathsf{nGen}(1^k)$
    $m \leftarrow \mathsf{A}(pk)$
    $e_0 \leftarrow \mathsf{nEnc}_{pk}(m)$
    $(e_1, t) \leftarrow \mathsf{nEnc}^*_{pk}(1^k)$, $sk_1 \leftarrow \mathsf{nOpen}(sk_0, pk, e_1, t, m)$
    $b' \leftarrow \mathsf{A}(sk_b, e_b)$
    Return 1 if $b = b'$, and 0 otherwise.
```

We say that nPKE *is* **ind-ncer***-secure if for a PPT adversary* A*, there exists a negligible function* $\mu = \mu(k)$ *such that* $\mathbf{Adv}^{\textsf{ind-ncer}}_{\mathsf{nPKE}}(\mathsf{A}, k) := 2\left|\Pr[\mathbf{Exp}^{\textsf{ind-ncer}}_{\mathsf{nPKE}}(\mathsf{A}, k) = 1] - \frac{1}{2}\right| \leq \mu$*.*

An NCER nPKE *is* **secure** *if it is* **ind-ncer** *secure.*

3.3 Tweaked NCER

We introduce a variant of NCER which tweaks the definition of NCER in the following two ways. First, the opening algorithm nOpen may be *inefficient*. In addition, the fake encryption algorithm is required to output a fake ciphertext e^* given the secret key sk and a plaintext m, so that decryption is “correct” with respect to e^* and m. This new notion is denoted as *tweaked NCER*. We formolize this notion below.

Definition 3.3 (Tweaked NCER). *A tweaked NCER scheme* tPKE *is a PKE that consists of five algorithms* $(\mathsf{tGen}, \mathsf{tEnc}, \mathsf{tEnc}^*, \mathsf{tDec}, \mathsf{tOpen})$ *defined as follows. Algorithms* $(\mathsf{tGen}, \mathsf{tEnc}, \mathsf{tDec})$ *form a PKE. Given the secret key* sk *and the public key* pk*, and a plaintext* m*, the PPT fake encryption algorithm* tEnc^* *outputs a ciphertext* e^**. Given the secret key* sk *and the public key* pk*, fake ciphertext* e^* *such that*

$e^* \leftarrow \mathsf{tEnc}^*_{pk}(sk, m')$ *for some* $m' \in \mathcal{M}_{\mathsf{tPKE}}$ *and a plaintext* m*, the inefficient algorithm* tOpen *outputs* sk^* *such that* $m = \mathsf{tDec}_{sk^*}(e^*)$.

- Correctness. *We require that* $m = \mathsf{tDec}_{sk}(c)$ *for all* $m \in \mathcal{M}$*, all* $(pk, sk) \leftarrow \mathsf{tGen}(1^k)$ *and all* $c \leftarrow \mathsf{tEnc}_{pk}(m)$.
- Security. *A tweaked NCER scheme* tPKE *is* **ind-tcipher** *secure if real and fake ciphertexts are indistinguishable. Specifically, for a PPT adversary* A*, consider the experiment* $\mathbf{Exp}^{\textsf{ind-tcipher}}_{\mathsf{tPKE}}$ *defined as follows.*

```
Experiment. $\mathbf{Exp}^{\textsf{ind-tcipher}}_{\mathsf{tPKE}}(\mathsf{A}, k)$
    $b \leftarrow \{0,1\}$
    $(pk, sk) \leftarrow \mathsf{tGen}(1^k)$
    $m \leftarrow \mathsf{A}(pk)$
    $e_0 \leftarrow \mathsf{tEnc}_{pk}(m)$
    $e_1 \leftarrow \mathsf{tEnc}^*_{pk}(sk, m)$
    $b' \leftarrow \mathsf{A}(sk, e_b)$
    Return 1 if $b = b'$, and 0 otherwise.
```

```
Experiment. $\mathbf{Exp}^{\textsf{ind-tncer}}_{\mathsf{tPKE}}(\mathsf{A}, k)$
    $b \leftarrow \{0,1\}$
    $(pk, sk_0) \leftarrow \mathsf{tGen}(1^k)$
    $m \leftarrow \mathsf{A}(pk)$
    $e_0 \leftarrow \mathsf{tEnc}^*_{pk}(sk_0, m)$
    $e_1 \leftarrow \mathsf{tEnc}^*_{pk}(sk_0, m')$ for $m' \in \mathcal{M}_{\mathsf{tPKE}}$
    $sk_1 \leftarrow \mathsf{tOpen}(e_1, m)$
    $b' \leftarrow \mathsf{A}(sk_b, e_b)$
    Return 1 if $b = b'$, and 0 otherwise.
```

We say that tPKE *is* **ind-tcipher** *secure if for a PPT adversary* A*, there exists a negligible function* $\mu = \mu(k)$ *such that* $\mathbf{Adv}^{\textsf{ind-tcipher}}_{\mathsf{tPKE}}(\mathsf{A}, k) := 2\left|\Pr[\mathbf{Exp}^{\textsf{ind-tcipher}}_{\mathsf{tPKE}}(\mathsf{A}, k) = 1] - \frac{1}{2}\right| \leq \mu$.

We say that tPKE *is* **ind-tncer** *secure if for an unbounded adversary* A*, there exists a negligible function* $\mu = \mu(s)$ *such that* $\mathbf{Adv}^{\textsf{ind-tncer}}_{\mathsf{tPKE}}(\mathsf{A}, k) := 2\left|\Pr[\mathbf{Exp}^{\textsf{ind-tncer}}_{\mathsf{tPKE}}(\mathsf{A}, k) = 1] - \frac{1}{2}\right| \leq \mu$.

A tweaked NCER tPKE *is* **secure** *if it is* $\{$**ind-tcipher**, **ind-tncer**$\}$ *secure.*

3.4 Key-Simulatable PKE

A key-simulatable public key encryption scheme is a PKE in which the public keys can be generated in two modes. In the first mode a public key is picked together with a secret key, whereas the second mode implies an oblivious public key generation without the secret key. Let $\mathcal{V}$ denote the set of public keys generated in the first mode and $\mathcal{K}$ denote the set of public keys generated in the second mode. Then it is possible that $\mathcal{K}$ contains $\mathcal{V}$ (i.e., $\mathcal{V} \subseteq \mathcal{K}$). Moreover, in case $\mathcal{V} \subset \mathcal{K}$ the set of public keys from $\mathcal{K} \setminus \mathcal{V}$ is not associated with any secret key. We respectively denote the keys in $\mathcal{V}$ and $\mathcal{K} \setminus \mathcal{V}$ as *valid* and *invalid* public keys. In addition to the key generation algorithms, key-simulatable PKE also consists of an efficient key faking algorithm that explains a public key from $\mathcal{V}$, that was generated in the first mode, as an obliviously generated public key from $\mathcal{K}$ that was generated without the corresponding secret key. The security requirement asserts that it is hard to distinguish a random element from $\mathcal{K}$ from a random element from $\mathcal{V}$. The formal definition follows. We note that the notion of key-simulatable PKE is very similar to the simulatable PKE [DN00] notion with the differences that the latter notion assumes that $\mathcal{K} = \mathcal{V}$ and further supports oblivious ciphertext generation and ciphertext faking.

Definition 3.4 (Key-simulatable PKE). *A key-simulatable public key encryption* sPKE *consists of five PPT algorithms* $(\mathsf{sGen}, \mathsf{sEnc}, \mathsf{sDec}, \widetilde{\mathsf{sGen}}, \widetilde{\mathsf{sGen}}^{-1})$ *defined as follows. Algorithms* $(\mathsf{sGen}, \mathsf{sEnc}, \mathsf{sDec})$ *form a PKE. Given the security parameter k, the oblivious public key generator* $\widetilde{\mathsf{sGen}}$ *returns a public key* pk' *from* $\mathcal{K}$ *and the random coins* r' *used to sample* pk'*. Given a public key* $pk \in \mathcal{V}$*, the key faking algorithm returns some random coins* r.

- Correctness. *We require that* $m = \mathsf{sDec}_{sk}(c)$ *for all* $m \in \mathcal{M}$*, all* $(pk, sk) \leftarrow \mathsf{sGen}(1^k)$ *and all* $c \leftarrow \mathsf{sEnc}_{pk}(m)$*.*

- Security. *A key-simulatable scheme* sPKE *is* **ind-cpa** *secure if* $(\mathsf{sGen}, \mathsf{sEnc}, \mathsf{sDec})$ *is* **ind-cpa** *secure. It is called* **ksim** *secure if it is hard to distinguish an obliviously generated key from a legitimately generated key. Specifically, for a PPT adversary* A*, there exists a negligible function* $\mu = \mu(k)$ *such that* $\mathbf{Adv}^{\mathsf{ksim}}_{\mathsf{sPKE}}(\mathsf{A}, k) := \left|\Pr\left[1 \leftarrow \mathsf{A}(r, pk)\right] - \Pr\left[1 \leftarrow \mathsf{A}(r', pk')\right]\right| \leq \mu$ *where* $(pk, sk) \leftarrow \mathsf{sGen}(1^k)$*,* $r \leftarrow \widetilde{\mathsf{sGen}}^{-1}(pk)$ *and* $(pk', r') \leftarrow \widetilde{\mathsf{sGen}}(1^k)$*.*

  A key-simulatable scheme sPKE *is* **secure** *if it is* $\{$**ind-cpa**, **ksim**$\}$ *secure.*

An *extended* key-simulatable PKE is a secure key-simulatable where in addition $\mathcal{V} \subset \mathcal{K}$ and it holds that $\Pr\left[pk \in \mathcal{K} \setminus \mathcal{V} \mid (pk, r) \leftarrow \widetilde{\mathsf{sGen}}(1^k)\right]$ is non-negligible.

3.5 Lossy PKE

A lossy public-key encryption scheme is a PKE with a standard key generation algorithm (that produces "real" keys) and a lossy key generation algorithm (that produces "lossy" keys that are indistinguishable from real ones), such that ciphertexts that are computed under real keys are committing to the plaintexts while ciphertexts that are computed under lossy keys are non-committing.

Definition 3.5 (Lossy PKE). *A lossy public key encryption scheme* loPKE *consists of five algorithms* $(\mathsf{loGen}, \mathsf{loGen}^*, \mathsf{loEnc}, \mathsf{loDec}, \mathsf{loOpen})$ *defined as follows. Algorithms* $(\mathsf{loGen}, \mathsf{loEnc}, \mathsf{loDec})$ *form a PKE. Given a security parameter* k*, the PPT lossy key generation algorithm* loGen^* *outputs a public key* pk^* *where* pk^* *is called a* lossy *public key. Given a lossy public key* pk^**, plaintext* $m \in \mathcal{M}_{\mathsf{loPKE}}$ *and ciphertext* $e = \mathsf{loEnc}_{pk^*}(m)$*, the (possibly inefficient) algorithm* loOpen *outputs* $r' \in \mathcal{R}_{\mathsf{loPKE}}$*.*

- Correctness. *We require that* $m = \mathsf{sDec}_{sk}(c)$ *for all* $m \in \mathcal{M}$*, all* $(pk, sk) \leftarrow \mathsf{sGen}(1^k)$ *and all* $c \leftarrow \mathsf{sEnc}_{pk}(m)$*. We further require that* $e = \mathsf{loEnc}_{pk^*}(m; r')$ *for* $pk^* \leftarrow \mathsf{sGen}^*(1^k)$*, plaintext* $m \in \mathcal{M}$*, ciphertext* $e = \mathsf{loEnc}_{pk^*}(m)$ *and* $r' \leftarrow \mathsf{loOpen}(pk^*, m, e)$*.*

- Security. *A lossy PKE scheme* loPKE *is* **ind-lossy** *secure if lossy keys are computationally indistinguishable from real keys. Specifically, for a PPT adversary* A*, there exists a negligible function* $\mu = \mu(k)$ *such that*

  $$\mathbf{Adv}^{\mathsf{ind\text{-}lossy}}_{\mathsf{loPKE}}(\mathsf{A}, k) := |\Pr[1 \leftarrow \mathsf{A}(pk)] - \Pr[1 \leftarrow \mathsf{A}(pk^*)]| \leq \mu$$

  for $(pk, sk) \leftarrow \mathsf{loGen}(1^k)$ *and* $pk^* \leftarrow \mathsf{loGen}^*(1^k)$*.*

  A lossy PKE scheme loPKE *is* **ind-lossycipher** *secure if encryptions of two distinct messages are indistinguishable when encrypted under a lossy public key. Specifically, for an* unbounded powerful *adversary* A*, there exists a negligible function* $\mu = \mu(s)$ *such that*

  $$\mathbf{Adv}^{\mathsf{ind\text{-}lossycipher}}_{\mathsf{loPKE}}(\mathsf{A}, k) := |\Pr[1 \leftarrow \mathsf{A}(e_0)] - \Pr[1 \leftarrow \mathsf{A}(e_1)]| \leq \mu$$

  for $pk^* \leftarrow \mathsf{loGen}^*(1^k)$ *and* $e \leftarrow \mathsf{loEnc}_{pk^*}(m_0)$*,* $e_1 \leftarrow \mathsf{loEnc}_{pk^*}(m_1)$ *and any* $m_0, m_1 \in \mathcal{M}_{\mathsf{loPKE}}$ *chosen by* A *given* pk^**.*

  A lossy PKE scheme loPKE *is* **secure** *if it is* $\{$**ind-lossy**, **ind-lossycipher**$\}$ *secure.*

3.6 Statistically-Hiding $\binom{2}{1}$-OT

We recall the definition of honest-receiver two-round statistically-hiding $\binom{2}{1}$-OT. We follow the notation from [HLOV11], but modify their definition to consider statistical privacy with respect to the receiver rather than the sender. In addition, we only consider a binary plaintext space.

Definition 3.6 (Statistically-hiding OT). *Oblivious transfer is a protocol between a sender* Sen *and a receiver* $\mathsf{Rec} = (\mathsf{Rec}_q, \mathsf{Rec}_r)$*. The sender* Sen *has two input bits* s_0*,* s_1*, and the receiver has a bit* b*. The receiver* Rec_q *generates a query* q *along with some state information* sk *and sends* q *to the sender. The sender evaluates* $\mathsf{q}(s_0, s_1)$ *and sends the result* $\mathsf{rsp} = \mathsf{Sen}(\mathsf{q}, s_0, s_1)$ *to the receiver* Rec_r *who uses* sk *to obtain* s_b*. We require the following properties from* $(\mathsf{Sen}, \mathsf{Rec})$*:*

- Correctness. *For all* $s_0, s_1 \in \{0, 1\}$ *and for all* $b \in \{0, 1\}$*, there exists a negligible function* $\mu = \mu(k)$ *such that* $\Pr[s_b = \mathsf{Rec}_r(sk, \mathsf{rsp}) \mid (\mathsf{q}, sk) \leftarrow \mathsf{Rec}_q(1^k, b); \mathsf{rsp} \leftarrow \mathsf{Sen}(\mathsf{q}, s_0, s_1)] \geq 1 - \mu$*.*

- Receiver Privacy. b *remains statistically hidden from* Sen*'s view. Specifically, for an unbounded powerful adversary* A *and a negligible function* $\mu = \mu(s)$ *it holds that*

$$|\Pr[1 \leftarrow \mathsf{A}(\mathsf{q}_0)] - \Pr[1 \leftarrow \mathsf{A}(\mathsf{q}_1)]| \leq \mu$$

  for $(\mathsf{q}_0, sk) \leftarrow \mathsf{Rec}_q(1^k, 0)$ *and* $(\mathsf{q}_1, sk) \leftarrow \mathsf{Rec}_q(1^k, 1)$

- Sender Privacy. *For any* $b \in \{0, 1\}$*, for any* s_0, s_1, s_0', s_1' *such that* $s_b = s_b'$ *,the following must hold for a PPT adversary* A *and a negligible function* $\mu = \mu(k)$

$$|\Pr[1 \leftarrow \mathsf{A}(\mathsf{rsp}_0)] - \Pr[1 \leftarrow \mathsf{A}(\mathsf{rsp}_1)]| \leq \mu$$

  for $(\mathsf{q}, sk) \leftarrow \mathsf{Rec}_q(1^k, b); \mathsf{rsp}_0 \leftarrow \mathsf{Sen}(\mathsf{q}, s_0, s_1)$ *and* $\mathsf{rsp}_1 \leftarrow \mathsf{Sen}(\mathsf{q}, s_0', s_1')$

In all of the above,the probabilities are over the randomness of Rec_q *and* Sen.

3.7 Hash Proof Systems

We recall the definition of hash proof systems (HPS), introduced by Cramer and Shoup [CS02]. For simplicity we frame the description by viewing HPS as key-encapsulation mechanisms (KEM). A KEM is a public-key encryption scheme that is used for encrypting random messages that are used as encryption keys for a symmetric-key encryption scheme, which in turn encrypts the actual plaintext. A HPS can be viewed as a KEM in which ciphertexts can be generated in two modes. The ciphertexts that are generated using the first mode are referred to as *valid* ciphertexts. For such ciphertexts the encapsulated key is well defined, and can be decapsulated using the secret key and also using the public key along with the "witness" of the ciphertext validity. The ciphertexts that are generated using the second mode are referred to as *invalid* ciphertexts and essentially contain no information on the encapsulated key. That is, given a public key and an invalid ciphertext, the distribution of the encapsulated key (as it will be produced by the decapsulation process using the secret key) is almost uniform. This is achieved by introducing redundancy into the secret key: each public key has many corresponding secret keys. It might not be even possible to decapsulate the key using the public key. The only computational requirement is that the two modes are computational indistinguishable: any PPT adversary cannot distinguish with a noticeable advantage between valid ciphertexts and invalid ciphertexts.

Let $\mathcal{C}$ be the set of all ciphertexts, $\mathcal{V} \subset \mathcal{C}$ be the set of all *valid* ciphertexts, $\mathcal{K}$ be the set of all symmetric keys, $\mathcal{PK}$ be the set of all public keys and $\mathcal{SK}$ be the set of all secret keys. We assume that there are efficient algorithms for sampling $sk \in \mathcal{SK}$, $c \in \mathcal{V}$ together with a witness w, and $c \in \mathcal{C} \setminus \mathcal{V}$. Let $\Lambda_{sk} : \mathcal{C} \to \mathcal{K}$ be a hash function indexed with $sk \in \mathcal{SK}$ that maps ciphertexts in $\mathcal{C}$ to symmetric keys in $\mathcal{K}$. A hash function Λ_{sk} is *projective* if there exists a projection $\delta : \mathcal{SK} \to \mathcal{PK}$ such that $\delta(sk) \in \mathcal{PK}$ defines the action of Λ_{sk} over the subset $\mathcal{V}$. That is, for every $c \in \mathcal{V}$, the value $k = \Lambda_{sk}(c)$ is uniquely determined by $pk = \delta(sk)$ and c. In other words, even though there are many different secret keys sk corresponding to the same public key pk, the action of Λ_{sk} over the subset of valid ciphertexts in completely determined by the public key pk. In contrast, the action of Λ_{sk} over the subset of invalid ciphertexts should be completely undetermined and it might not be possible to compute Λ_{sk} from pk and $c \in \mathcal{C} \setminus \mathcal{V}$. Formally,

Definition 3.7 (HPS). *A HPS* HPS *consists of three PPT algorithms* (Param, Pub, Priv) *defined as follows. Given the security parameter k, algorithm* Param *generates the parameterized instances of the form* $(\mathbb{G}, \mathcal{K}, \mathcal{C}, \mathcal{V}, \mathcal{SK}, \mathcal{PK}, \Lambda_{(\cdot)}, \delta(\cdot))$*, where* $\mathbb{G}$ *may contain some additional structural parameters and* $\Lambda_{(\cdot)}, \delta$ *are efficiently computable functions. Given a public key* $pk = \delta(sk)$*, ciphertext* $c \in \mathcal{V}$ *together with a corresponding witness* w *for* c *being a valid ciphertext, the deterministic public evaluation algorithm* Pub *outputs the encapsulated key* $\Lambda_{sk}(c)$*. Given a secret key* sk *and a ciphertext* $c \in \mathcal{C}$*, the deterministic private evaluation algorithm* Priv *outputs the encapsulated key* $\Lambda_{sk}(c)$*.*

- Projectiveness. *Function* $\delta(\cdot) : \mathcal{SK} \to \mathcal{PK}$ *is a projection such that* $\delta(sk) \in \mathcal{PK}$ *defines the action of the hash function* Λ_{sk} *over* $\mathcal{V}$*, where for every* $\mathcal{C} \in \mathcal{V}$*, the value* $K = \Lambda_{sk}(c)$ *is uniquely determined by* $pk = \delta(sk)$*.*

- Security. *A HPS scheme* HPS *is* **ind-hps** *secure if valid and invalid ciphertexts are indistinguishable. Specifically, for a PPT adversary* A *there exists a negligible function* $\mu = \mu(k)$ *such that*

$$\mathbf{Adv}^{\textsf{ind-hps}}_{\textsf{HPS}}(\mathsf{A}, k) := |\Pr[1 \leftarrow \mathsf{A}(\mathcal{C}, \mathcal{V}, c_0)] - \Pr[1 \leftarrow \mathsf{A}(\mathcal{C}, \mathcal{V}, c_1)]| \leq \mu$$

where $\mathcal{C}$ *and* $\mathcal{V}$ *are generated using* $\mathsf{Param}(1^k)$*,* $c_0 \leftarrow \mathcal{V}$ *and* $c_1 \leftarrow \mathcal{C} \setminus \mathcal{V}$*.*

A HPS scheme HPS *is* 1*-universal if for a PPT adversary* A *there exists a negligible function* $\mu = \mu(k)$ *such that*

$$|\Pr[1 \leftarrow \mathsf{A}(pk, \mathsf{Priv}_{sk}(c))] - \Pr[1 \leftarrow \mathsf{A}(pk, K)]| \leq \mu$$

for all $c \in \mathcal{C} \setminus \mathcal{V}$*,* $sk \in \mathcal{SK}$*,* $K \leftarrow \mathcal{K}$ *and* $pk = \delta(sk)$*.*

HPS HPS *is* **secure** *if it is* **ind-hps** *secure and* 1*-universal.*

4 Selective Opening Security for the Receiver

In this section we provide negative and positive results regarding security for the receiver in the presence of selective opening attacks. First, we show that **rind-so** is strictly weaker than **rsim-so** security by constructing a scheme that meets the former but not the latter level of security. We then relate the different forms of security under SO attacks with non-committing encryption (for the receiver). Specifically, we show that secure NCER implies **rsim-so** and that secure tweaked NCER implies **rind-so**. Interestingly, we show that the converse implications do not hold. In terms of constructions, we show that tweaked NCER can be constructed from various primitives such as key-simulatable PKE, two-round honest-receiver statistically-hiding $\binom{2}{1}$-OT protocol, secure HPS and NCER. The DDH based secure NCER scheme of [CHK05] that works for polynomial message space turns out to be secure tweaked NCER for exponential message space.

4.1 rind-so Secure PKE $\not\Longrightarrow$ rsim-so Secure PKE

In this section we construct a **rind-so** secure encryption scheme PKE that is not **rsim-so** secure. Our starting point is an **ind-ncer** secure scheme nPKE and a $\{$**stat-hide**, **comp-bind**$\}$ secure NISHCOM nisCom. The public key of our scheme is defined by two independent public keys of nPKE, whereas the secret key corresponds to the matched secret keys. Moreover, encrypting a plaintext is carried out by computing a commitment of the plaintext and encrypting the opening information of this commitment under the public keys. Finally, decryption is computed by decrypting the opening information of the commitment and verifying whether it is consistent with the commitment. Below we prove that PKE is **rind-so** secure but not **rsim-so** secure (due to **comp-bind** security of the commitment scheme). Our separating scheme requires that the message and randomness spaces of nisCom, denoted by $\mathcal{M}_{\mathsf{nisCom}}$ and $\mathcal{R}_{\mathsf{nisCom}}$, are compatible with the message space $\mathcal{M}_{\mathsf{nPKE}}$ of nPKE (as we encrypt the committed message as well as the randomness used for computing the commitment). We formally define compatibility as follows:

Definition 4.1. *An* **ind-ncer** *secure NCER* nPKE *and a* $\{$**stat-hide**, **comp-bind**$\}$ *secure NISHCOM* nisCom *are said to be* compatible *if* $\mathcal{M}_{\mathsf{nPKE}} = \mathcal{M}_{\mathsf{nisCom}} = \mathcal{R}_{\mathsf{nisCom}}$.

We proceed with our main theorem for this section and provide a concrete example of schemes that satisfy the compatibility criteria in Section 4.1.1.

Theorem 4.2. *Assume there exist an* **ind-ncer** *secure NCER and a* $\{$**stat-hide**, **comp-bind**$\}$ *secure NISHCOM that are compatible. Then, there exists a PKE that is* **rind-so** *secure but is not* **rsim-so** *secure.*

Proof: We describe our separating encryption scheme first. Consider a scheme nPKE = (nGen, nEnc, nEnc*, nDec, nOpen) that is secure NCER (cf. Definition 3.2) and an NISHCOM nisCom = (nisCommit, nisOpen) (cf. Definition 3.1) that are compatible. We define the encryption scheme PKE = (Gen, Enc, Dec) as follows.

```
Gen($1^k$)
    $(pk_0, sk_0) \leftarrow \mathsf{nGen}(1^k)$
    $(pk_1, sk_1) \leftarrow \mathsf{nGen}(1^k)$
    $pk = (pk_0, pk_1)$
    $sk = (sk_0, sk_1)$
    Return $(pk, sk)$

Enc$_{pk}(m)$
    $c \leftarrow \mathsf{nisCommit}(m; r)$
    $e_0 \leftarrow \mathsf{nEnc}_{pk_0}(m)$
    $e_1 \leftarrow \mathsf{nEnc}_{pk_1}(r)$
    Return $e = (e_0, e_1, c)$

Dec$_{sk}(e)$
    $e := (e_0, e_1, c)$
    $m = \mathsf{nDec}_{sk_0}(e_0)$
    $r = \mathsf{nDec}_{sk_1}(e_1)$
    if $c = \mathsf{nisCommit}(m, r)$
      Return $m$
    else Return $\perp$
```

The proof follows from Lemmas 4.1 and 4.5 below which formalize that PKE is **rind-so** secure but not **rsim-so** secure. ■

Lemma 4.1. *Assume that* nPKE *is* **ind-ncer** *secure and* nisCom *is* $\{$**stat-hide**, **comp-bind**$\}$ *secure, then* PKE *is* **rind-so** *secure.*

Proof: More precisely we show that for any PPT adversary A attacking PKE there exist a PPT adversary B and an unbounded powerful adversary C such that

$$\mathbf{Adv}^{\mathsf{rind\text{-}so}}_{\mathsf{PKE}}(\mathsf{A}, k) \leq n\left(4 \cdot \mathbf{Adv}^{\mathsf{ind\text{-}ncer}}_{\mathsf{nPKE}}(\mathsf{B}, k) + \mathbf{Adv}^{\mathsf{stat\text{-}hide}}_{\mathsf{nisCom}}(\mathsf{C}, k)\right).$$

We prove this lemma using the following sequence of experiments.

- $\mathbf{Exp}_0 = \mathbf{Exp}^{\mathsf{rind\text{-}so}}_{\mathsf{PKE}}$.
- $\mathbf{Exp}_1$ is identical to $\mathbf{Exp}_0$ except that the first component of each ciphertext in the vector $\mathbf{e}$ is computed using nEnc^* of nPKE. That is, for all $i \in [n]$ ciphertext e_i is defined by (e^*_{i0}, e_{i1}, c_i) such that $(e^*_{i0}, t_{i0}) \leftarrow \mathsf{nEnc}^*_{pk_{i0}}(1^k)$. Furthermore, if $i \in \mathcal{I}$ (i.e., A asks to open the ith ciphertext), then $\mathbf{Exp}_1$ computes $sk^*_{i0} \leftarrow \mathsf{nOpen}(sk_{i0}, e^*_{i0}, t_{i0}, m_i)$ and hands (sk^*_{i0}, sk_{i1}) to A.
- $\mathbf{Exp}_2$ is identical to $\mathbf{Exp}_1$ except that the second component of each ciphertext in the vector $\mathbf{e}$ is computed using nEnc^* of nPKE, That is, for all $i \in [n]$ ciphertext e_i is defined by $(e^*_{i0}, e^*_{i1}, c_i)$ such that $(e^*_{i1}, t_{i1}) \leftarrow \mathsf{nEnc}^*_{pk_{i1}}(1^k)$. Furthermore, if $i \in \mathcal{I}$ (i.e., A asks to open the ith ciphertext), then $\mathbf{Exp}_2$ computes $sk^*_{i1} \leftarrow \mathsf{nOpen}(sk_{i1}, e^*_{i1}, t_{i1}, r_i)$ and hands (sk^*_{i0}, sk^*_{i1}) to A, where r_i is the randomness used to compute c_i.
- $\mathbf{Exp}_3$ is identical to $\mathbf{Exp}_2$ except that the third component of each ciphertext in the vector $\mathbf{e}$ is a commitment of a dummy message. That is, for all $i \in [n]$ ciphertext e_i is defined by $(e^*_{i0}, e^*_{i1}, c^*_i)$ such that $c^*_i \leftarrow \mathsf{nisCommit}(m^*_i; r^*_i)$, where m^*_i is a dummy message from $\mathcal{M}_{\mathsf{nisCom}}$ and $r^*_i \leftarrow \mathcal{R}_{\mathsf{nisCom}}$. Furthermore, if $i \in \mathcal{I}$ then $\mathbf{Exp}_3$ first computes $r_i \leftarrow \mathsf{nisOpen}(c^*_i, m_i)$. Then it computes $sk^*_{i1} \leftarrow \mathsf{nOpen}(sk_{i1}, e^*_{i1}, t_{i1}, r_i)$ and hands (sk^*_{i0}, sk^*_{i1}) to A, where r_i is the randomness returned by $\mathsf{nisOpen}$.

We note that although the third experiment is not efficient (the experiment needs to equivocate the commitment without a trapdoor), it does not introduce a problem in our proof: an adversary that distinguishes between $\mathbf{Exp}_2$ and $\mathbf{Exp}_3$ gives rise to an unbounded adversary that breaks the statistical hiding property of the commitment scheme used by our construction.

Let ϵ_j be the advantage of A in $\mathbf{Exp}_j$, i.e. $\epsilon_j := 2\left|\Pr[\mathbf{Exp}_j(\mathsf{A}, k) = 1] - \frac{1}{2}\right|$. We first note that $\epsilon_3 = 0$ since in experiment $\mathbf{Exp}_3$ the adversary receives a vector of ciphertexts that are statistically independent of the encrypted plaintexts, implying that the adversary (even with unbounded computing power) outputs the correct bit b with probability 1/2. Next we show that $|\epsilon_0 - \epsilon_1| \leq 2n\Delta_{\mathsf{ind\text{-}ncer}}$ and $|\epsilon_1 - \epsilon_2| \leq 2n\Delta_{\mathsf{ind\text{-}ncer}}$, where $\Delta_{\mathsf{ind\text{-}ncer}} = \mathbf{Adv}^{\mathsf{ind\text{-}ncer}}_{\mathsf{nPKE}}(\mathsf{B}, k)$ for a PPT adversary B. Finally, we argue that $|\epsilon_2 - \epsilon_3| \leq n\Delta_{\mathsf{stat\text{-}hide}}$ where $\Delta_{\mathsf{stat\text{-}hide}} = \mathbf{Adv}^{\mathsf{stat\text{-}hide}}_{\mathsf{nisCom}}(\mathsf{C}, k)$ for an unbounded powerful adversary C. All together this implies that $|\epsilon_0 - \epsilon_3| \leq 4n\Delta_{\mathsf{ind\text{-}ncer}} + n\Delta_{\mathsf{stat\text{-}hide}}$ and that $\epsilon_0 \leq 4n\Delta_{\mathsf{ind\text{-}ncer}} + n\Delta_{\mathsf{stat\text{-}hide}}$, which proves the lemma.

Claim 4.2. $|\epsilon_0 - \epsilon_1| \leq 2n\Delta_{\mathsf{ind\text{-}ncer}}$, *where* $\Delta_{\mathsf{ind\text{-}ncer}} = \mathbf{Adv}^{\mathsf{ind\text{-}ncer}}_{\mathsf{nPKE}}(\mathsf{B}, k)$.

Proof: We prove the claim by introducing n intermediate hybrids experiments between $\mathbf{Exp}_0$ and $\mathbf{Exp}_1$; the difference between two consequent hybrids is bounded by a reduction to **ind-ncer** security of nPKE. More specifically, we introduce $n-1$ intermediate hybrid experiments so that $E_0 = \mathbf{Exp}_0$, $E_n = \mathbf{Exp}_1$ and the ith hybrid experiment E_i is defined recursively. That is,

- $E_0 = \mathbf{Exp}_0$.
- For $i = [n]$, E_i is identical to E_{i-1} except that the ith ciphertext e_i is computed by (e^*_{i0}, e_{i1}, c_i) where $(e^*_{i0}, t_{i0}) \leftarrow \mathsf{nEnc}^*_{pk_{i0}}(1^k)$. Furthermore, if $i \in \mathcal{I}$ (i.e., if A asks to open the ith ciphertext), then E_i computes $sk^*_{i0} \leftarrow \mathsf{nOpen}(sk_{i0}, e^*_{i0}, t_{i0}, m_i)$ and hands (sk^*_{i0}, sk_{i1}) to A.

Clearly $E_n = \mathbf{Exp}_1$ where the first component of all ciphertext is computed using nEnc^*. Let γ_i define the advantage of A in E_i, i.e. $\gamma_i := 2\left|\Pr[E_i(\mathsf{A}, k) = 1] - \frac{1}{2}\right|$. Next we show that $|\gamma_{i-1} - \gamma_i| \leq 2\Delta_{\mathsf{ind\text{-}ncer}}$ for all $i \in [n]$. This implies that $|\gamma_0 - \gamma_n| \leq 2n\Delta_{\mathsf{ind\text{-}ncer}}$. Now, since $\gamma_0 = \epsilon_0$ and $\gamma_n = \epsilon_1$ we get $|\epsilon_0 - \epsilon_1| \leq 2n\Delta_{\mathsf{ind\text{-}ncer}}$, thus proving the claim.

We fix $i \in [n]$ and prove that $|\gamma_{i-1} - \gamma_i| \leq 2\Delta_{\textsf{ind-ncer}}$. Specifically, we show that any adversary B that wishes to distinguish a real ciphertext from a fake one relative to nPKE can utilize the power of adversary A. Upon receiving pk from experiment $\mathbf{Exp}^{\textsf{ind-ncer}}_{\textsf{nPKE}}$and i, B interacts with A as follows.

1. B samples first a bit $b \leftarrow \{0,1\}$ and sets $pk_{i0} = pk$. It then uses nGen to generate the rest of the public keys to obtain $\mathbf{pk}$ (and all but $(i0)$th secret key).[4] Finally, it hands $\mathbf{pk}$ to A that returns Dist and $\mathsf{Resamp}_{\mathsf{Dist}}$.
2. B samples $\mathbf{m} \leftarrow \mathsf{Dist}(1^k)$ and outputs m_i to $\mathbf{Exp}^{\textsf{ind-ncer}}_{\textsf{nPKE}}$ that returns (sk, e). B then sets $sk_{i0} = sk$. (Note that this completes vector $\mathbf{sk}$ since B generated the rest of the secret keys in the previous step).
   - For $j \in [i-1]$, B computes the first component of ciphertext e_j by $(e_{j0}, t_{j0}) \leftarrow \mathsf{nEnc}^*_{pk_{j0}}(1^k)$. B completes e_j honestly (i.e., exactly as specified in Enc).
   - For $j = i$, B sets the first component of e_j to be e. B completes e_j honestly.
   - For $j \in [i+1, n]$, B computes ciphertext e_j honestly.

   Let $\mathbf{e} = (e_j)_{j \in [n]}$. B hands $\mathbf{e}$ to A that returns $\mathcal{I}$.
3. B resamples $\mathbf{m}' \leftarrow \mathsf{Resamp}_{\mathsf{Dist}}(\mathbf{m}_{\mathcal{I}})$. Subsequently it hands $\mathbf{m}^*$ to A as well as secret keys for all the indices that are specified in $\mathcal{I}$, where $\mathbf{m}^* = \mathbf{m}$ if $b = 0$, $\mathbf{m}^* = \mathbf{m}'$ otherwise. That is,
   - If $j \in \mathcal{I}$ lies in $[i-1]$, then B computes $sk^*_{j0} \leftarrow \mathsf{nOpen}(sk_{j0}, e_{j0}, t_{j0}, m_j)$ and hands (sk^*_{j0}, sk_{j1}).
   - If $j \in \mathcal{I}$ equals i, then B hands (sk_{j0}, sk_{j1}) where sk_{j0} is same as sk that B had received from $\mathbf{Exp}^{\textsf{ind-ncer}}_{\textsf{nPKE}}$.
   - If $j \in \mathcal{I}$ lies in $[i+1, n]$, then B returns (sk_{j0}, sk_{j1}).
4. B outputs 1 in experiment $\mathbf{Exp}^{\textsf{ind-ncer}}_{\textsf{nPKE}}$ if A wins.

Next, note that B perfectly simulates E_{i-1} if it received a real ciphertext e within (sk, e). Otherwise, B perfectly simulates E_i. This ensures that the probability that B outputs 1 in $\mathbf{Exp}^{\textsf{ind-ncer}}_{\textsf{nPKE}}$ given a real ciphertext is at least as good as the probability that A wins in E_{i-1}. On the other hand, the probability that B outputs 1 in $\mathbf{Exp}^{\textsf{ind-ncer}}_{\textsf{nPKE}}$ given a fake ciphertext is at least as good as the probability that A wins in E_i. Since the advantage of A in E_i is γ_i, its winning probability (cf. Definition 2.2) $\Pr[E_i(\mathsf{A}, k) = 1]$ in the experiment is $\frac{\gamma_i}{2} + \frac{1}{2}$. Similarly, the winning probability of A in experiment E_{i-1} is $\frac{\gamma_{i-1}}{2} + \frac{1}{2}$. Denoting the bit picked in $\mathbf{Exp}^{\textsf{ind-ncer}}_{\textsf{nPKE}}$ by c we get,

$$\underbrace{\Pr\left[1 \leftarrow \mathsf{B}(sk, e) \mid (pk, sk) \leftarrow \mathsf{nGen}(1^k) \wedge e \leftarrow \mathsf{nEnc}_{pk}(m_i)\right]}_{=\Pr[1 \leftarrow \mathsf{B} \mid c=0]} \geq \frac{\gamma_{i-1}}{2} + \frac{1}{2} \text{ and}$$

$$\underbrace{\Pr\left[1 \leftarrow \mathsf{B}(sk, e) \mid (pk, sk) \leftarrow \mathsf{nGen}(1^k) \wedge (e, t_e) \leftarrow \mathsf{nEnc}^*_{pk}(1^k) \wedge sk \leftarrow \mathsf{nOpen}(sk, e, t_e, m_i)\right]}_{=\Pr[1 \leftarrow \mathsf{B} \mid c=1]} \geq \frac{\gamma_i}{2} + \frac{1}{2}.$$

[4] Recall that each public key within $\mathbf{pk}$ includes two public keys relative to nPKE.

This implies that

$$\begin{aligned}
\Delta_{\textsf{ind-ncer}} &= \mathbf{Adv}^{\textsf{ind-ncer}}_{\textsf{nPKE}}(\mathsf{B},k) = 2\left|\Pr[\mathbf{Exp}^{\textsf{ind-ncer}}_{\textsf{nPKE}}(\mathsf{B},k)=1]-\frac{1}{2}\right| \\
&= 2\left|\Pr[0\leftarrow \mathsf{B}\mid c=0]\underbrace{\Pr(c=0)}_{=1/2}+\Pr[1\leftarrow \mathsf{B}\mid c=1]\underbrace{\Pr(c=1)}_{=1/2}-\frac{1}{2}\right| \\
&= |\Pr[0\leftarrow \mathsf{B}\mid c=0]+\Pr[1\leftarrow \mathsf{B}\mid c=1]-1| \\
&= |\Pr[1\leftarrow \mathsf{B}\mid c=0]-\Pr[1\leftarrow \mathsf{B}\mid c=1]| \\
&\geq \frac{|\gamma_{i-1}-\gamma_i|}{2}.
\end{aligned}$$

□

The following claim follows by a similar hybrid argument as the one described above.

Claim 4.3. $|\epsilon_1-\epsilon_2|\leq 2n\Delta_{\textsf{ind-ncer}}$, *where* $\Delta_{\textsf{ind-ncer}}=\mathbf{Adv}^{\textsf{ind-ncer}}_{\textsf{nPKE}}(\mathsf{B},k)$.

Finally, we prove the following claim.

Claim 4.4. $|\epsilon_2-\epsilon_3|\leq n\Delta_{\textsf{stat-hide}}$, *where* $\Delta_{\textsf{stat-hide}}=\mathbf{Adv}^{\textsf{stat-hide}}_{\textsf{nisCom}}(\mathsf{C},k)$.

Proof: We prove the claim by introducing n intermediate hybrids experiments between $\mathbf{Exp}_2$ and $\mathbf{Exp}_3$; we show that each pair of consecutive experiments is statistically indistinguishable based on **stat-hide** security of the NISHCOM. These hybrid experiments are defined as follows:

- $H_0=\mathbf{Exp}_2$.
- For $i=[n]$, H_i is identical to H_{i-1} except that the ith ciphertext e_i in $\mathbf{e}$ is computed as $(e^*_{i0},e^*_{i1},c^*_i)$ where $c^*_i\leftarrow \mathsf{nisCommit}(m^*_i;r^*_i)$, where m^*_i is a dummy message from $\mathcal{M}_{\mathsf{nisCom}}$ and $r^*\leftarrow\mathcal{R}_{\mathsf{nisCom}}$. Furthermore, if $i\in\mathcal{I}$, then H_i computes $r_i\leftarrow\mathsf{nisOpen}(c^*_i,m_i)$ and hands (sk^*_{i0},sk^*_{i1}) to A.

We remark again that the hybrid experiments defined above are not efficient, but this is not an issue as we rely on the statistical security of the underlying NISHCOM.

Clearly, $H_n=\mathbf{Exp}_3$ where the third component of each ciphertext within $\mathbf{e}$ is computed using dummy messages. Let ν_i be the advantage of A in H_i, i.e., $\nu_i:=2\left|\Pr[H_i(\mathsf{A},k)=1]-\frac{1}{2}\right|$. Next, we show that $|\nu_{i-1}-\nu_i|\leq\Delta_{\textsf{stat-hide}}$ for all $i\in[n]$, where $\Delta_{\textsf{stat-hide}}=\mathbf{Adv}^{\textsf{stat-hide}}_{\textsf{nisCom}}(\mathsf{C},k)$. All together, this implies that $|\nu_0-\nu_n|\leq n\Delta_{\textsf{stat-hide}}$. Since $\nu_0=\epsilon_2$ and $\nu_n=\epsilon_3$ we get that $|\epsilon_2-\epsilon_3|\leq n\Delta_{\textsf{stat-hide}}$ which proves the claim.

Fix $i\in[n]$. The only difference between experiments H_{i-1} and H_i is relative to the third component of ciphertext e_i. Namely, in H_{i-1}, the third component in e_i is a commitment to m_i where m_i is the ith element in $\mathbf{m}$. On the other hand, in H_i it is a commitment to a dummy message from $\mathcal{M}_{\mathsf{nisCom}}$. As the underlying NISHCOM satisfies statistical hiding property, even an unbounded adversary C cannot distinct H_{i-1} and H_i with probability better than $\Delta_{\textsf{stat-hide}}$, so $|\nu_{i-1}-\nu_i|\leq\Delta_{\textsf{stat-hide}}$ as desired. □ □

We conclude with the proof of the following lemma.

Lemma 4.5. PKE *is not* **rsim-so** *secure.*

Proof: We then rely on a result of [BDWY12] which establishes that no decryption verifiable **ind-cpa** secure is **rsim-so**. Informally, decryption verifiability implies the existence of an algorithm W (that either outputs accept or reject), such that it is hard to find pk, sk_0, sk_1, distinct m_0, m_1 and a ciphertext e where both $W(pk, sk_0, e, m_0)$ and $W(pk, sk_1, e, m_1)$ accept. Note that it is hard to find two valid secret keys and plaintexts as required since decryption follows successfully only if the commitment that is part of the ciphertext is also correctly opened. In particular, an adversary that produces a ciphertext that can be successfully decrypted into two distinct plaintexts (under two different keys) must break the **comp-bind** security of the underlying commitment scheme.[5] This implies that PKE is not **rsim-so** secure. □

4.1.1 Compatible Secure NCER and Secure NISHCOM

We instantiate the commitment scheme with the Paillier based scheme of Damgård and Nielsen [DN02, DN03], which is comprised of the following algorithms that use public parameters (N, g) where N is a k-bit RSA composite and $g = x^N \bmod N^2$ for an uniformly random $x \leftarrow \mathbb{Z}_N^*$. Moreover,

- nisCommit, given N, g and message $m \in \mathbb{Z}_N$, pick $r \leftarrow \mathbb{Z}_N^*$ and compute $g^m \cdot r^N \bmod N^2$.
- nisOpen, given commitment c and message m, compute randomness r such that $c = g^m \cdot r^N \bmod N^2$. Namely, find first $\tilde{r}$ such that $c = \tilde{r}^N \bmod N^2$. This implies that $\tilde{r}^N = (x^N)^m \cdot r^N \bmod N^2$ for some $r \in \mathbb{Z}_N^*$, since we can fix $r = \tilde{r}/x^m$.

This scheme is computationally binding, as a commitment is simply a random Paillier encryption of zero. Furthermore, opening to two different values implies finding the Nth root of g (which breaks the underlying assumption of Paillier, i.e., DCR). Finally, the NCER can be instantiated with the scheme from [CHK05] that is also based on the DCR assumption. The message space of these two primitives is $\mathbb{Z}_N$. In addition, the randomness of the commitment scheme is $\mathbb{Z}_N^*$ and thus can be made consistent with the plaintext spaces, as it is infeasible to find an element in $\mathbb{Z}_N/\mathbb{Z}_N^*$.

4.2 Secure Tweaked NCER $\Longrightarrow$ rind-so Secure PKE

In this section we prove that every secure tweaked NCER is a **rind-so** secure PKE. Intuitively, this follows since real ciphertexts are indistinguishable from fake ones, and fake ciphertexts do not commit to any fixed plaintext. This implies that the probability of distinguishing an encryption of one message from another is exactly half, even for an unbounded adversary.

Theorem 4.3. *Assume there exists an* $\{$**ind-tcipher**, **ind-tncer**$\}$ *secure tweaked NCER, then there exists a PKE that is* **rind-so** *secure.*

Proof: More precisely, let $\mathsf{tPKE} = (\mathsf{tGen}, \mathsf{tEnc}, \mathsf{tEnc}^*, \mathsf{tDec}, \mathsf{tOpen})$ denote a secure tweaked NCER. Then we prove that tPKE is **rind-so** secure, by proving that for any PPT adversary A attacking tPKE in the **rind-so** experiment there exist a PPT adversary B and an unbounded powerful adversary C such that $\mathbf{Adv}_{\mathsf{tPKE}}^{\textsf{rind-so}}(\mathsf{A}, k) \leq 2n \left(\mathbf{Adv}_{\mathsf{tPKE}}^{\textsf{ind-tcipher}}(\mathsf{B}, k) + \mathbf{Adv}_{\mathsf{tPKE}}^{\textsf{ind-tncer}}(\mathsf{C}, k) \right)$.

We modify experiment **rind-so** step by step, defining a sequence of $2n + 1$ experiments and bound the advantage of A in the last experiment. The proof is then concluded by proving that any two intermediate consecutive experiments are indistinguishable due to either **ind-tcipher** security or **ind-tncer** security of tPKE. Specifically, we define a sequence of hybrid experiments $\{\mathbf{Exp}_i\}_{i=0}^{2n}$ as follows.

- $\mathbf{Exp}_0 = \mathbf{Exp}_{\mathsf{tPKE}}^{\textsf{rind-so}}$.

[5] Recall that the decryption algorithm verifies first whether the commitment within the ciphertext is consistent with the decrypted ciphertexts (that encrypt the committed message and its corresponding randomness for commitment).

- For all $i \in [n]$, $\mathbf{Exp}_i$ is identical to $\mathbf{Exp}_{i-1}$ except that the ith ciphertext in vector $\mathbf{e}$ is computed by $e_i^* \leftarrow \mathsf{tEnc}^*_{pk_i}(sk_i, m_i)$, so that if $i \in \mathcal{I}$ then $\mathbf{Exp}_i$ outputs the secret key sk_i computed by tGen and hands sk_i to adversary A (here we rely on the additional property of tEnc^*).

- For all $i \in [n]$, $\mathbf{Exp}_{n+i}$ is identical to $\mathbf{Exp}_{n+i-1}$ except that the ith ciphertext in vector $\mathbf{e}$ is computed by sampling a random message $m_i^* \in \mathcal{M}_{\mathsf{tPKE}}$ first and then computing $e_i^* \leftarrow \mathsf{tEnc}^*_{pk_i}(sk_i, m_i^*)$. Next, if $i \in \mathcal{I}$ then $\mathbf{Exp}_{n+i}$ computes a secret key $sk_i^* \leftarrow \mathsf{tOpen}(e_i^*, m_i)$ and hands sk_i^* to A.

Let ϵ_i denote the advantage of A in experiment $\mathbf{Exp}_i$ i.e. $\epsilon_i := \left|\Pr[\mathbf{Exp}_i(\mathsf{A}, k) = 1] - \frac{1}{2}\right|$. We first note that $\epsilon_{2n} = 0$ since in experiment $\mathbf{Exp}_{2n}$ the adversary receives a vector of ciphertexts that are statistically independent of the encrypted plaintexts, implying that the adversary outputs the correct bit b with probability 1/2. We next show that $|\epsilon_{i-1} - \epsilon_i| \leq 2\Delta_{\mathsf{ind\text{-}tcipher}}$ for any $i \in [n]$, where $\Delta_{\mathsf{ind\text{-}tcipher}} = \mathbf{Adv}^{\mathsf{ind\text{-}tcipher}}_{\mathsf{tPKE}}(\mathsf{B}, k)$ for a PPT adversary B. Finally, we prove that $|\epsilon_{n+i-1} - \epsilon_{n+i}| \leq 2\Delta_{\mathsf{ind\text{-}tncer}}$ for any $i \in [n]$, where $\Delta_{\mathsf{ind\text{-}tncer}} = \mathbf{Adv}^{\mathsf{ind\text{-}tncer}}_{\mathsf{tPKE}}(\mathsf{C}, k)$ for an unbounded powerful adversary C. Together this implies that $|\epsilon_0 - \epsilon_{2n}| \leq 2n(\Delta_{\mathsf{ind\text{-}tcipher}} + \Delta_{\mathsf{ind\text{-}tncer}})$. So we conclude that $\epsilon_0 \leq n(\Delta_{\mathsf{ind\text{-}tcipher}} + \Delta_{\mathsf{ind\text{-}tncer}}) + \epsilon_{2n} = 2n(\Delta_{\mathsf{ind\text{-}tcipher}} + \Delta_{\mathsf{ind\text{-}tncer}})$ which concludes the proof of the theorem.

Claim 4.6. $|\epsilon_{i-1} - \epsilon_i| \leq 2n\Delta_{\mathsf{ind\text{-}tcipher}}$ *for all* $i \in [n]$, *where* $\Delta_{\mathsf{ind\text{-}tcipher}} = \mathbf{Adv}^{\mathsf{ind\text{-}tcipher}}_{\mathsf{tPKE}}(\mathsf{B}, k)$.

Proof: In the following, we prove that one can design an adversary B that distinguishes a real ciphertext from a fake one in $\mathbf{Exp}^{\mathsf{ind\text{-}tcipher}}_{\mathsf{tPKE}}$, using adversary A. B interacts with A as follows:

1. Upon receiving pk from $\mathbf{Exp}^{\mathsf{ind\text{-}tcipher}}_{\mathsf{tPKE}}$ and an integer i, B sets $pk_i = pk$. It picks a bit b randomly. It then generates the rest of the public and secret key pairs using tGen for all $j \in [n] \setminus i$, obtaining $\mathbf{pk}$. It hands $\mathbf{pk}$ to A who returns Dist and $\mathsf{Resamp}_{\mathsf{Dist}}$.

2. B samples $\mathbf{m} \leftarrow \mathsf{Dist}(1^k)$ and hands m_i to $\mathbf{Exp}^{\mathsf{ind\text{-}tcipher}}_{\mathsf{tPKE}}$ which returns (sk, e). B fixes $e_i = e$ and completes $\mathbf{sk}$ by setting $sk_i = sk$. Next, for $j \in [i-1]$ it computes $e_j \leftarrow \mathsf{tEnc}^*_{pk_j}(sk_j, m_j)$, whereas for $j \in [i+1, n]$ it samples randomness $r_j \leftarrow \mathcal{R}_{\mathsf{tPKE}}$ and computes $e_j \leftarrow \mathsf{tEnc}_{pk_j}(m_j; r_j)$. Let $\mathbf{e} = (e_i)_{i \in [n]}$. B hands $\mathbf{e}$ to A who returns $\mathcal{I}$.

3. B samples $\mathbf{m}' \leftarrow \mathsf{Resamp}(\mathbf{m}_{\mathcal{I}})$ and hands A $\mathbf{m}^*$ and the following secret keys for all the indices that are specified in $\mathcal{I}$. Here $\mathbf{m}^*$ is $\mathbf{m}$ if $b = 0$ and $\mathbf{m}'$ otherwise. That is,

   - If $j \in \mathcal{I}$ lies in $[i-1]$ or in $[i+1, n]$, then B returns sk_j.
   - If $j \in \mathcal{I}$ equals i, then B returns sk.

4. B outputs 1 in $\mathbf{Exp}^{\mathsf{ind\text{-}tcipher}}_{\mathsf{tPKE}}$ if A wins.

Next, note that B perfectly simulates $\mathbf{Exp}_{i-1}$ if it receives a real ciphertext e within (sk, e). On the other hand, B perfectly simulates $\mathbf{Exp}_i$ if e is a fake ciphertext. This ensures that the probability that B outputs 1 given a real ciphertext is at least as good as the probability that A wins in $\mathbf{Exp}_{i-1}$. On the other hand, the probability that B outputs 1 given a fake ciphertext is at least as good as the probability that A wins in $\mathbf{Exp}_i$. Since the advantage of A in $\mathbf{Exp}_i$ is ϵ_i, its winning probability (cf. Definition 2.2) $\Pr[\mathbf{Exp}_i(\mathsf{A}, k) = 1]$ in the experiment is $\frac{\epsilon_i}{2} + \frac{1}{2}$. Similarly, the winning probability of A in experiment $\mathbf{Exp}_{i-1}$ is $\frac{\epsilon_{i-1}}{2} + \frac{1}{2}$. Denoting the bit picked in $\mathbf{Exp}^{\mathsf{ind\text{-}tcipher}}_{\mathsf{tPKE}}$ by c we get,

$$\underbrace{\Pr\left[1 \leftarrow \mathsf{B}(pk, sk, e, m_i) \mid (pk, sk) \leftarrow \mathsf{tGen}(1^k) \wedge e \leftarrow \mathsf{tEnc}_{pk}(m_i)\right]}_{=\Pr[1 \leftarrow \mathsf{B} \mid c=0]} \geq \frac{\epsilon_{i-1}}{2} + \frac{1}{2} \text{ and}$$

$$\underbrace{\Pr\left[1 \leftarrow \mathsf{B}(pk, sk, e^*, m_i) \mid (pk, sk) \leftarrow \mathsf{tGen}(1^k) \wedge e^* \leftarrow \mathsf{tEnc}^*_{pk}(sk, m_i)\right]}_{=\Pr[1 \leftarrow \mathsf{B} \mid c=1]} \geq \frac{\epsilon_i}{2} + \frac{1}{2}.$$

This implies that

$$\begin{aligned}
\Delta_{\textsf{ind-tcipher}} &= \mathbf{Adv}^{\textsf{ind-tcipher}}_{\textsf{tPKE}}(\mathsf{B}, k) = 2\left|\Pr[\mathbf{Exp}^{\textsf{ind-tcipher}}_{\textsf{tPKE}}(\mathsf{B}, k) = 1] - \frac{1}{2}\right| \\
&= 2\left|\Pr[0 \leftarrow \mathsf{B} \mid c = 0]\underbrace{\Pr(c = 0)}_{=1/2} + \Pr[1 \leftarrow \mathsf{B} \mid c = 1]\underbrace{\Pr(c = 1)}_{=1/2} - \frac{1}{2}\right| \\
&= |\Pr[0 \leftarrow \mathsf{B} \mid c = 0] + \Pr[1 \leftarrow \mathsf{B} \mid c = 1] - 1| \\
&= |\Pr[1 \leftarrow \mathsf{B} \mid c = 0] - \Pr[1 \leftarrow \mathsf{B} \mid c = 1]| \geq \frac{|\epsilon_{i-1} - \epsilon_i|}{2}
\end{aligned}$$

□

Claim 4.7. $|\epsilon_{n+i-1} - \epsilon_{n+i}| \leq 2n\Delta_{\textsf{ind-tcipher}}$ *for all* $i \in [n]$*, where* $\Delta_{\textsf{ind-tncer}} = \mathbf{Adv}^{\textsf{ind-tncer}}_{\textsf{tPKE}}(\mathsf{C}, k)$.

Proof: Below, we prove that one can design an unbounded powerful adversary C that distinguishes two views generated in **ind-tncer** experiment, using adversary A. C interacts with A as follows:

1. Upon receiving pk from $\mathbf{Exp}^{\textsf{ind-tncer}}_{\textsf{tPKE}}$ and an integer i, C sets $pk_i = pk$ and picks a bit b. It then generates the rest of the public and secret key pairs using tGen for all $j \in [n] \setminus \{i\}$, obtaining $\mathbf{pk}$. It hands $\mathbf{pk}$ to A who returns Dist and $\mathsf{Resamp}_{\mathsf{Dist}}$.

2. C samples $\mathbf{m} \leftarrow \mathsf{Dist}(1^k)$ hands m_i to $\mathbf{Exp}^{\textsf{ind-tncer}}_{\textsf{tPKE}}$which returns (sk, e). C fixes $e_i = e$ and completes $\mathbf{sk}$ by setting $sk_i = sk$. Next, for $j \in [i-1]$ it samples $\mathbf{m}^*_j \leftarrow \mathcal{M}_{\mathsf{tPKE}}$ and computes $e_j \leftarrow \mathsf{tEnc}^*_{pk_j}(sk_j, m^*_j)$, whereas for $j \in [i+1, n]$ it computes $e_j \leftarrow \mathsf{tEnc}^*_{pk_j}(sk_j, m_j)$. Let $\mathbf{e} = (e_j)_{j \in [n]}$. C hands $\mathbf{e}$ to A who returns $\mathcal{I}$.

3. C samples $\mathbf{m}' \leftarrow \mathsf{Resamp}(\mathbf{m}_{\mathcal{I}})$ and hands $\mathbf{m}^*$ to A and the following secret keys for all the indices that are specified in $\mathcal{I}$. Here $\mathbf{m}^*$ is $\mathbf{m}$ if $b = 0$ and $\mathbf{m}'$ otherwise. That is,

   - If $j \in \mathcal{I}$ lies in $[i-1]$, then C returns sk_j such that $sk_j = \mathsf{tOpen}(e_j, m_j)$.
   - If $j \in \mathcal{I}$ equals i, then C returns sk.
   - If $j \in \mathcal{I}$ lies in $[i+1, n]$, then C returns sk_j.

4. C outputs 1 in $\mathbf{Exp}^{\textsf{ind-tncer}}_{\textsf{tPKE}}$ if A wins.

Next, note that B perfectly simulates $\mathbf{Exp}_{n+i-1}$ if it receives a real ciphertext e within (sk, e). On the other hand, B perfectly simulates $\mathbf{Exp}_{n+i}$ if e is a fake ciphertext and sk is a secret key returned by tOpen. This ensures that the probability that B outputs 1 given a real ciphertext is at least as good as the probability that A wins in $\mathbf{Exp}_{n+i-1}$. On the other hand, the probability that B outputs 1 given a fake ciphertext is at least as good as the probability that A wins in $\mathbf{Exp}_{n+i}$. Since the advantage of A in $\mathbf{Exp}_i$ is ϵ_{n+i}, its winning probability (c.f Definition 2.2) $\Pr[\mathbf{Exp}_i(\mathsf{A}, k) = 1]$ in the experiment is $\frac{\epsilon_{n+i}}{2} + \frac{1}{2}$. Similarly, the winning probability of A in experiment $\mathbf{Exp}_{n+i-1}$ is $\frac{\epsilon_{n+i-1}}{2}+\frac{1}{2}$. Denoting the bit picked in $\mathbf{Exp}^{\mathsf{ind\text{-}tncer}}_{\mathsf{tPKE}}$ by c we get,

$$\underbrace{\Pr\left[1 \leftarrow \mathsf{C}(sk, e) \mid (pk, sk) \leftarrow \mathsf{tGen}(1^k) \wedge e \leftarrow \mathsf{tEnc}^*_{pk}(sk, m_i)\right]}_{=\Pr[1 \leftarrow \mathsf{C} \mid c=0]} \geq \frac{\epsilon_{n+i-1}}{2} + \frac{1}{2} \text{ and}$$

$$\underbrace{\Pr\left[1 \leftarrow \mathsf{C}(sk^*, e^*) \mid (pk, sk) \leftarrow \mathsf{tGen}(1^k) \wedge e^* \leftarrow \mathsf{tEnc}^*_{pk}(sk, m^*) \wedge sk^* \leftarrow \mathsf{tOpen}(e^*, sk, m_i)\right]}_{=\Pr[1 \leftarrow \mathsf{C} \mid c=1]} \geq \frac{\epsilon_{n+i}}{2} + \frac{1}{2}.$$

Following a similar argument as in the previous claim, we conclude that $2\Delta_{\mathsf{ind\text{-}tncer}} \geq |\epsilon_{n+i-1} - \epsilon_{n+i}|$. □
■

4.3 Secure NCER $\Longrightarrow$ rsim-so Secure PKE

In this section we prove that secure NCER implies selective opening security in the presence of receiver corruption. Our proof is shown for the stronger simulation based security definition but holds for the indistinguishability definition as well.

Theorem 4.4. *Assume there exists an* **ind-ncer** *secure PKE, then there exists a PKE that is* **rsim-so** *secure.*

Proof: More precisely, let $\mathsf{nPKE} = (\mathsf{nGen}, \mathsf{nEnc}, \mathsf{nEnc}^*, \mathsf{nDec}, \mathsf{nOpen})$ denote a **ind-ncer** secure PKE. Then we prove that nPKE is **rsim-so** secure, by proving that for any PPT adversary A attacking nPKE in the **rsim-so** experiment there exists a PPT adversary B such that

$$\mathbf{Adv}^{\mathsf{rsim\text{-}so}}_{\mathsf{nPKE}}(\mathsf{A}, k) \leq n \cdot \mathbf{Adv}^{\mathsf{ind\text{-}ncer}}_{\mathsf{nPKE}}(\mathsf{B}, k).$$

In order to prove this theorem, we modify experiment **rsim-so-real** step by step, defining a sequence of $n+1$ experiments. We then design a simulator for the last experiment in this sequence and prove that it is identical to experiment **rsim-so-ideal**. The proof is then concluded by proving that any two intermediate consecutive experiments are computationally indistinguishable due to the ciphertext indistinguishability property (i.e. **ind-ncer** property) of nPKE. Specifically, we define a sequence of hybrid experiments $\{\mathbf{Exp}_i\}_{i=0}^{n}$ as follows.

- $\mathbf{Exp}_0 = \mathbf{Exp}^{\mathsf{rind\text{-}so}}_{\mathsf{nPKE}}$.
- For all $i \in [n]$, $\mathbf{Exp}_i$ is identical to $\mathbf{Exp}_{i-1}$ except that the ith ciphertext in vector $\mathbf{e}$ is computed by $(e_i^*, t_i) \leftarrow \mathsf{nEnc}^*_{pk_i}(1^k)$, so that if $i \in \mathcal{I}$ then $\mathbf{Exp}_i$ computes $sk_i^* \leftarrow \mathsf{nOpen}(sk_i, e_i^*, t_i, m_i))$ and hands sk_i^* to adversary A.

Next, we design a simulator S for experiment **rsim-so-ideal**. Specifically, the simulator picks n pairs of public and secret key pairs of nPKE and hands the public key vector $\mathbf{pk}$ to A. Upon receiving Dist from A, S outputs Dist. S then invokes nEnc^* of nPKE n times in order to generate $\mathbf{e}$ and hands this vector to A. Upon receiving $\mathcal{I}$ from A, S outputs $\mathcal{I}$ and receives back a vector $\mathbf{m}_{\mathcal{I}}$. For $j \in \mathcal{I}$, S invokes $sk_j^* \leftarrow \mathsf{nOpen}(sk_j, e_j, t_j, m_j)$ and returns (sk_j^*, m_j) to A. Finally, the simulator outputs whatever A outputs. It is easy to verify that experiments $\mathbf{Exp}_n$ and **rsim-so-ideal**, when executed with S, produce the same distribution.

It remains to show that $\mathbf{Exp}_{i-1}$ and $\mathbf{Exp}_i$ are computationally indistinguishable for all $i \in [n]$. Fix $i \in [n]$ and let D_i be an efficient distinguisher that distinguishes $\mathbf{Exp}_{i-1}$ and $\mathbf{Exp}_i$. Namely, let ϵ_{i-1} is the probability that D_i outputs 1 when given a distribution generated relative to $\mathbf{Exp}_{i-1}$, and likewise ϵ_i is the probability that D_i outputs 1 when given a distribution generated relative to $\mathbf{Exp}_i$. We prove that $|\epsilon_{i-1} - \epsilon_i| \leq \Delta_{\mathsf{ind\text{-}ncer}}$ for any $i \in [n]$, where $\Delta_{\mathsf{ind\text{-}ncer}} = \mathbf{Adv}^{\mathsf{ind\text{-}ncer}}_{\mathsf{nPKE}}(\mathsf{B}, k)$. All together this implies that $\epsilon_0 - \epsilon_n \leq n\Delta_{\mathsf{ind\text{-}ncer}}$ which proves the theorem.

In the following, we prove that one can design a PPT adversary B that distinguishes a real ciphertext from a fake one in experiment $\mathbf{Exp}^{\mathsf{ind\text{-}ncer}}_{\mathsf{nPKE}}$ using adversary D_i that distinguishes between experiments $\mathbf{Exp}_i$ and $\mathbf{Exp}_{i-1}$. B interacts with D_i as follows:

1. Upon receiving pk from experiment $\mathbf{Exp}^{\mathsf{ind\text{-}ncer}}_{\mathsf{nPKE}}$ and an integer i, B sets $pk_i = pk$. It then generates the rest of the public and secret key pairs using nGen for all $j \in [n] \setminus i$, obtaining $\mathbf{pk}$. It hands $\mathbf{pk}$ to D_i who returns Dist.
2. B samples $\mathbf{m} \leftarrow \mathsf{Dist}(1^k)$ and hands m_i to experiment $\mathbf{Exp}^{\mathsf{ind\text{-}ncer}}_{\mathsf{nPKE}}$ which returns (sk, e). B fixes $e_i = e$ and completes $\mathbf{sk}$ by setting $sk_i = sk$. Next, for $j \in [i-1]$ it computes $(e_j, t_j) \leftarrow \mathsf{nEnc}^*_{pk_j}(1^k)$, whereas for $j \in [i+1, n]$ it samples randomness $r_j \leftarrow \mathcal{R}_{\mathsf{nPKE}}$ and computes $e_j \leftarrow \mathsf{nEnc}_{pk_j}(m_j; r_j)$. Let $\mathbf{e} = (e_i)_{i \in [n]}$. B hands $\mathbf{e}$ to D_i who returns $\mathcal{I}$.
3. B hands D_i pairs of secret keys and plaintexts for all the indices that are specified in $\mathcal{I}$. That is,
   - If $j \in \mathcal{I}$ lies in $[i-1]$, then B computes $sk^*_j \leftarrow \mathsf{nOpen}(sk_j, e_j, t_j, m_j)$ and returns (sk^*_j, m_j).
   - If $j \in \mathcal{I}$ equals i, then B returns (sk, m).
   - If $j \in \mathcal{I}$ lies in $[i+1, n]$, then B simply returns (sk_j, m_j).
4. B outputs 1 in experiment $\mathbf{Exp}^{\mathsf{ind\text{-}ncer}}_{\mathsf{nPKE}}$ if D_i outputs 1.

Next, note that B perfectly simulates $\mathbf{Exp}_{i-1}$ if it receives a real ciphertext e within (pk, sk, e, m). On the other hand, B perfectly simulates $\mathbf{Exp}_i$ if e is a fake ciphertext. This ensures that the probability that B outputs 1 given a real ciphertext is at least as good as the probability that D_i outputs 1 in $\mathbf{Exp}_{i-1}$. On the other hand, the probability that B outputs 1 given a fake ciphertext is at least as good as the probability that D_i outputs 1 in $\mathbf{Exp}_i$. Denoting the bit picked in $\mathbf{Exp}^{\mathsf{ind\text{-}ncer}}_{\mathsf{nPKE}}$ by c we get,

$$\underbrace{\Pr\left[1 \leftarrow \mathsf{B}(pk, sk, m_i, e) \mid (pk, sk) \leftarrow \mathsf{nGen}(1^k) \wedge e \leftarrow \mathsf{nEnc}_{pk}(m_i)\right]}_{=\Pr[1 \leftarrow \mathsf{B} \mid c=0]} \geq \epsilon_{i-1} \text{ and}$$

$$\underbrace{\Pr\left[1 \leftarrow \mathsf{B}(pk, sk, m_i, e) \mid (e, t) \leftarrow \mathsf{nEnc}^*_{pk}(1^k) \wedge sk \leftarrow \mathsf{nOpen}(sk, e, t, m_i)\right]}_{=\Pr[1 \leftarrow \mathsf{B} \mid c=1]} \geq \epsilon_i.$$

This implies that

$$\begin{aligned}
\Delta_{\mathsf{ind\text{-}ncer}} &= \mathbf{Adv}^{\mathsf{ind\text{-}ncer}}_{\mathsf{nPKE}}(\mathsf{B}, k) = 2\left|\Pr[\mathbf{Exp}^{\mathsf{ind\text{-}ncer}}_{\mathsf{nPKE}}(\mathsf{B}, k) = 1] - \frac{1}{2}\right| \\
&= |\Pr[1 \leftarrow \mathsf{B} \mid c = 0] - \Pr[1 \leftarrow \mathsf{B} \mid c = 1]| \\
&\geq |\epsilon_{i-1} - \epsilon_i|
\end{aligned}$$

■

4.4 rsim-so Secure PKE $\not\Rightarrow$ Secure NCER and Tweaked NCER

In this section we prove that **rsim-so** does not imply both tweaked NCER and NCER by providing a concrete counter example based on an extended key-simulatable PKE (cf. Section 3.4). The key point in our proof is that in some cases simulatable public keys cannot be explained as valid public keys. Formally,

Theorem 4.5. *Assume there exists an* $\{$**ind-cpa**, **ksim**$\}$ *secure extended key-simulatable PKE, then there exists a PKE that is* **rsim-so** *secure but is neither a* $\{$**ind-tcipher**, **ind-tncer**$\}$ *secure tweaked NCER nor a* **ind-ncer** *secure NCER.*

Proof: We describe our separating encryption scheme first. Namely, given an extended key-simulatable PKE $\mathsf{sPKE} = (\mathsf{sGen}, \mathsf{sEnc}, \mathsf{sDec}, \widetilde{\mathsf{sGen}}, \widetilde{\mathsf{sGen}}^{-1})$ for a plaintext space $\mathcal{M}_{\mathsf{sPKE}}$, we construct a new scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with a binary plaintext space that is **rsim-so** secure, and thus also **rind-so** secure, yet it does not imply tweaked NCER. For simplicity, we assume that $\mathcal{M}_{\mathsf{sPKE}}$ is the binary space $\{0, 1\}$. The DDH based instantiation of sPKE with $\mathcal{V} \subset \mathcal{K}$ from Section 4.4.1 is defined with respect to this space.

```
$\mathsf{Gen}(1^k)$
    $\alpha \leftarrow \{0, 1\}$
    $(pk_\alpha, sk_\alpha) \leftarrow \mathsf{sGen}(1^k)$
    $(pk_{1-\alpha}, r_{1-\alpha}) \leftarrow \widetilde{\mathsf{sGen}}(1^k)$
    $pk = (pk_0, pk_1)$
    $sk = (\alpha, sk_\alpha, r_{1-\alpha})$
    Return $(pk, sk)$
```

```
$\mathsf{Enc}_{pk}(b)$
    $e_0 \leftarrow \mathsf{Enc}_{pk_0}(b)$
    $e_1 \leftarrow \mathsf{Enc}_{pk_1}(b)$
    Return $e = (e_0, e_1)$
```

```
$\mathsf{Dec}_{sk}(e)$
    $sk = (\alpha, sk_\alpha, r_{1-\alpha})$
    $e := (e_0, e_1)$
    $b = \mathsf{Dec}_{sk_\alpha}(e_\alpha)$
    Return $b$
```

The proof follows from Lemmas 4.8 and 4.11. ■

Lemma 4.8. PKE *is* **rsim-so** *secure.*

Proof: Namely, we prove that for any PPT adversaries B and C respectively attacking the **ind-cpa** and **ksim** security of sPKE there exists a PPT adversary A such that

$$\mathbf{Adv}^{\mathsf{rsim\text{-}so}}_{\mathsf{PKE}}(\mathsf{A}, k) \leq n \left(\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{sPKE}}(\mathsf{B}, k) + \mathbf{Adv}^{\mathsf{ksim}}_{\mathsf{sPKE}}(\mathsf{C}, k) \right).$$

To prove this lemma, we modify experiment **rsim-so-real** step by step, defining a sequence of $2n + 1$ experiments. We then design a simulator and prove that the last experiment in this sequence is identical to experiment **rsim-so-ideal** when executed along with the simulator. The proof is then concluded by proving that any two intermediate consecutive experiments are computationally indistinguishable due to the **ksim** and **ind-cpa** security of sPKE. Specifically, we define a sequence of hybrid experiments $\{\mathbf{Exp}_i\}_{i=0}^{2n}$ as follows.

- $\mathbf{Exp}_0 = \mathbf{Exp}^{\mathsf{rind\text{-}so}}_{\mathsf{PKE}}$.
- For all $i \in [n]$, $\mathbf{Exp}_i$ is identical to $\mathbf{Exp}_{i-1}$ except for the following change. The ith public key pk_i in vector $\mathbf{pk}$ consists of two public keys pk_{i0}, pk_{i1} that are computed using sEnc. Namely $(pk_{ij}, sk_{ij}) \leftarrow \mathsf{sGen}(1^k)$ for $j \in \{0, 1\}$. For a random α, the secret key sk_i is set to $(\alpha, sk_{i\alpha}, r_{i(1-\alpha)})$ where $r_{i(1-\alpha)} \leftarrow \widetilde{\mathsf{sGen}}^{-1}(pk_{i(1-\alpha)})$.

- For all $i \in [n]$, $\mathbf{Exp}_{n+i}$ is identical to $\mathbf{Exp}_{n+i-1}$ except for the following changes. The ith ciphertext e_i consisting of (e_{i0}, e_{i1}) in vector $\mathbf{e}$ is computed as follows: (i) a random bit α is picked and (ii) $e_{i\alpha} \leftarrow \mathsf{sEnc}_{pk_{i\alpha}}(0)$ and $e_{i(1-\alpha)} \leftarrow \mathsf{sEnc}_{pk_{i(1-\alpha)}}(1)$. Later, if $i \in \mathcal{I}$ and the plaintext is 0 then $\mathbf{Exp}_i$ sets its secret key sk_i as $(\alpha, sk_{i\alpha}, r_{i(1-\alpha)})$ where $r_{i(1-\alpha)} \leftarrow \widetilde{\mathsf{sGen}}^{-1}(pk_{i(1-\alpha)})$. Else, it sets sk_i as $((1-\alpha), sk_{i(1-\alpha)}, r_{i\alpha})$ where $r_{i\alpha} \leftarrow \widetilde{\mathsf{sGen}}^{-1}(pk_{i\alpha})$. It then hands sk_i to adversary A.

Next, we design a simulator S for experiment **rsim-so-ideal**. Specifically, the simulator picks n pairs of public and secret key pairs of sPKE using sGen. It then computes the public key vector $\mathbf{pk}$ using the public keys and hands it to A. Upon receiving Dist from A, S outputs Dist in experiment **rsim-so-ideal**. To compute the ith ciphertext e_i in vector $\mathbf{e}$, S does exactly what $\mathbf{Exp}_{n+i}$ does for computing its ith ciphertext. It hands over $\mathbf{e}$ to A. Upon receiving $\mathcal{I}$ from A, S outputs $\mathcal{I}$ in experiment **rsim-so-ideal** and receives back a bit vector $\mathbf{b}_{\mathcal{I}}$. For all $i \in \mathcal{I}$, S computes the ith secret key sk_i exactly the way $\mathbf{Exp}_{n+i}$ computes its sk_i and returns (sk_i, b_i) to A. Finally, the simulator outputs whatever A outputs. It is easy to verify that experiments $\mathbf{Exp}_{2n}$ and **rsim-so-ideal**, when executed with S, produce the same distribution.

It remains to show that $\mathbf{Exp}_{i-1}$ and $\mathbf{Exp}_i$ are computationally indistinguishable for all $i \in [2n]$. We prove this in two steps. Let D_i be an efficient distinguisher that distinguishes $\mathbf{Exp}_{i-1}$ and $\mathbf{Exp}_i$. Namely, let ϵ_{i-1} is the probability that D_i outputs 1 when given a distribution generated relative to $\mathbf{Exp}_{i-1}$, and likewise ϵ_i is the probability that D_i outputs 1 when given a distribution generated relative to $\mathbf{Exp}_{i-1}$. We show that $|\epsilon_{i-1} - \epsilon_i| \leq \Delta_{\mathsf{ksim}}$ for any $i \in [n]$, where $\Delta_{\mathsf{ksim}} = \mathbf{Adv}^{\mathsf{ksim}}_{\mathsf{sPKE}}(\mathsf{C}, k)$. Next, we prove that $|\epsilon_{n+i-1} - \epsilon_{n+i}| \leq \Delta_{\mathsf{ind\text{-}cpa}}$ for any $i \in [n]$, where $\Delta_{\mathsf{ind\text{-}cpa}} = \mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{sPKE}}(\mathsf{B}, k)$. All together this implies that $\epsilon_0 - \epsilon_{2n} \leq n(\epsilon_{\mathsf{ksim}} + \epsilon_{\mathsf{ind\text{-}cpa}})$ which proves the lemma.

Claim 4.9. *$|\epsilon_{i-1} - \epsilon_i| \leq n\Delta_{\mathsf{ksim}}$ for all $i \in [n]$, where $\Delta_{\mathsf{ksim}} = \mathbf{Adv}^{\mathsf{ksim}}_{\mathsf{sPKE}}(\mathsf{C}, k)$.*

Proof: Intuitively, the experiments $\mathbf{Exp}_{i-1}$ and $\mathbf{Exp}_i$ are computationally indistinguishable since the only difference between them is that in $\mathbf{Exp}_{i-1}$ one of the public keys in pk_i is picked using $\widetilde{\mathsf{sGen}}$, whereas both public keys in pk_i in $\mathbf{Exp}_i$ are computed using sGen. That is, one of the keys in $\mathbf{Exp}_{i-1}$ is sampled from $\mathcal{K}$. Whereas, in $\mathbf{Exp}_i$ both keys are sampled from $\mathcal{V}$, the set of valid public keys. Any distinguisher D_i for these two experiments can be successfully transformed into a distinguisher C for $\mathbf{Exp}^{\mathsf{ksim}}_{\mathsf{sPKE}}$. We now show how this can be achieved. C interacts with D_i as follows:

1. Upon receiving (pk, r) from $\mathbf{Exp}^{\mathsf{ksim}}_{\mathsf{sPKE}}$ and an integer i, C picks a random bit α and sets $pk_{i\alpha} = pk$ whereas $pk_{i(1-\alpha)}$ is computed using sGen. For all $j \in [1, i-1]$, it then generates the public and secret key pairs exactly as in $\mathbf{Exp}_{i-1}$ (that is, both public keys within every pk_j are generated using sGen). Next, for all $j \in [i+1, n]$ C generates the public and secret key pairs exactly as in $\mathbf{Exp}_i$ (that is, one of public keys within every pk_j is generated using sGen and the other key is generated using $\widetilde{\mathsf{sGen}}$). Next, it completes $\mathbf{pk}$ with the public keys chosen above and hands $\mathbf{pk}$ to D_i who returns Dist.

2. C samples $\mathbf{b} \leftarrow \mathsf{Dist}(1^k)$ and computes the ciphertext vector $\mathbf{e}$ the way it is computed in $\mathbf{Exp}_{i-1}$ or $\mathbf{Exp}_i$ (note that $\mathbf{e}$ is computed in the same way both in $\mathbf{Exp}_{i-1}$ and $\mathbf{Exp}_i$; so choosing either way makes no difference).

3. Upon receiving $\mathcal{I}$ from D_i, C hands it pairs of secret keys and plaintexts for all the indices that are specified in this set. That is,

    - If $j \in \mathcal{I}$ lies in $[i-1]$, then C returns (sk_j, b_j) such that $sk_j = (\beta, sk_{j\beta}, r_{j(1-\beta)})$ for a random $\beta \in \{0, 1\}$ and $r_{j(1-\beta)} \leftarrow \widetilde{\mathsf{sGen}}^{-1}(pk_{j(1-\beta)})$.

- If $j \in \mathcal{I}$ equals i, then C returns (sk_j, b_j) such that $sk_j = ((1-\alpha), sk_{j(1-\alpha)}, r_{j\alpha})$ with $r_{j\alpha} = r$.
- If $j \in \mathcal{I}$ lies in $[i+1, n]$, then C returns (sk_j, b_j) such that $sk_j = (\beta, sk_{j\beta}, r_{j(1-\beta)})$ where $(pk_{j\beta}, sk_{j\beta}) \leftarrow \mathsf{sGen}(1^\kappa)$ and $(pk_{j(1-\beta)}, r_{j(1-\beta)}) \leftarrow \widetilde{\mathsf{sGen}}(1^k)$ (that is, the pair $(pk_{j\beta}, sk_{j\beta})$ was generated by sGen, whereas the other public key $pk_{j(1-\beta)}$ was generated by the oblivious sampling algorithm $\widetilde{\mathsf{sGen}}$).

4. C outputs 1 in $\mathbf{Exp}^{\mathsf{ksim}}_{\mathsf{sPKE}}$ if D_i outputs 1.

Next, note that C perfectly emulates $\mathbf{Exp}_{i-1}$ if it receives a challenge public key pk that is picked obliviously using $\widetilde{\mathsf{sGen}}$. On the other hand, C perfectly emulates $\mathbf{Exp}_i$ if pk was computed using $\mathsf{sGen}(1^\kappa)$ along with its secret key and the randomness r returned by $\mathbf{Exp}^{\mathsf{ksim}}_{\mathsf{sPKE}}$ along with pk has been obtained via $\widetilde{\mathsf{sGen}}^{-1}$. This ensures that the probability that C outputs 1 given an obliviously picked public key pk is at least as good as the probability that D_i outputs 1 in $\mathbf{Exp}_{i-1}$. On the other hand, the probability that C outputs 1 given a legitimate public key pk is at least as good as the probability that D_i outputs 1 in $\mathbf{Exp}_i$. That is,

$$\Pr\left[1 \leftarrow \mathsf{C}(pk, r)\right] \geq \epsilon_{i-1} \;\text{ when } (pk, r) \leftarrow \widetilde{\mathsf{sGen}}(1^k) \text{ and}$$

$$\Pr\left[1 \leftarrow \mathsf{C}(pk, r)\right] \geq \epsilon_{i} \;\text{ when } (pk, sk) \leftarrow \mathsf{sGen}(1^k) \wedge r \leftarrow \widetilde{\mathsf{sGen}}^{-1}(pk).$$

This implies that $\Delta_{\mathsf{ksim}} = \mathbf{Adv}^{\mathsf{ksim}}_{\mathsf{sPKE}}(\mathsf{A}, k) \geq |\epsilon_{i-1} - \epsilon_i|$ and concludes our proof. We finally note that the above reduction works even for a stronger distinguisher D_i that may ask for the randomness used to sample the public key that C claims to choose obliviously for every pk_i of $\mathbf{pk}$. □

Claim 4.10. $|\epsilon_{n+i-1} - \epsilon_{n+i}| \leq n\Delta_{\mathsf{ind\text{-}cpa}}$ *for all* $i \in [n]$, $\Delta_{\mathsf{ind\text{-}cpa}} = \mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{sPKE}}(\mathsf{B}, k)$.

Proof: The difference between $\mathbf{Exp}_{n+i-1}$ and $\mathbf{Exp}_{n+i}$ is that the ith ciphertext e_i in the vector $\mathbf{e}$ consists of encryptions of the same plaintext in $\mathbf{Exp}_{n+i-1}$ and different plaintexts in $\mathbf{Exp}_{n+i}$. In the following, we prove that one can design an adversary B that can win in $\mathbf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathsf{sPKE}}$ using adversary D_i that distinguishes between experiments $\mathbf{Exp}_{n+i-1}$ and $\mathbf{Exp}_{n+i}$. B interacts with D_i as follows:

1. Upon receiving pk from $\mathbf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathsf{sPKE}}$ and an integer i, B picks a random bit α and sets $pk_{i\alpha} = pk$ whereas $pk_{i(1-\alpha)}$ is computed using sGen. It then generates the rest of the public and secret key pairs using sGen for all $j \in [n] \setminus i$. Next, it completes $\mathbf{pk}$ with the plaintexts and public keys chosen above and hands $\mathbf{pk}$ to D_i who returns Dist. B hands $(0, 1)$ to t$\mathbf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathsf{sPKE}}$ and gets back the challenge ciphertext e which either encrypts 0 or 1.

2. B samples $\mathbf{b} \leftarrow \mathsf{Dist}(1^k)$ and fixes $e_{i\alpha} = e$ and $e_{i(1-\alpha)} \leftarrow \mathsf{sEnc}_{pk_{i(1-\alpha)}}(b_i)$ where b_i is the ith bit in the vector $\mathbf{b}$. Next, for $j \in [i-1]$ B computes e_j such that it consists of the encryptions of 0 and 1 in some random order. For $j = [i+1, n]$, B computes e_j such that it consists of the encryptions of same plaintext b_j. I.e, $e_{j0} \leftarrow \mathsf{sEnc}_{pk_{i0}}(b_j)$ and $e_{j1} \leftarrow \mathsf{sEnc}_{pk_{i1}}(b_j)$.

3. Upon receiving $\mathcal{I}$ from D_i, B hands it pairs of secret keys and plaintexts for all the indices that are specified in this set. That is,

    - If $j \in \mathcal{I}$ lies in $[i-1]$, then B returns (sk_j, b_j) such that $sk_j = (\beta, sk_{j\beta}, r_{j(1-\beta)})$ if $e_{j\beta}$ encrypts b_j under public key $pk_{j\beta}$ and $r_{j(1-\beta)} \leftarrow \widetilde{\mathsf{sGen}}^{-1}(pk_{j(1-\beta)})$.

- If $j \in \mathcal{I}$ equals i, then B returns (sk_j, b_j) such that $sk_j = ((1-\alpha), sk_{j(1-\alpha)}, r_{j\alpha})$ with $r_{j\alpha} = \widetilde{\mathsf{sGen}}^{-1}(pk_{j\alpha})$. Note that $pk_{j\alpha} = pk$, where pk is received from $\mathbf{Exp}^{\textsf{ind-cpa}}_{\textsf{sPKE}}$.
- If $j \in \mathcal{I}$ lies in $[i+1, n]$, then B returns (sk_j, b_j) such that $sk_j = (\beta, sk_{j\beta}, r_{j(1-\beta)})$ for a random $\beta \in \{0,1\}$ and $r_{j(1-\beta)} \leftarrow \widetilde{\mathsf{sGen}}^{-1}(pk_{j(1-\beta)})$.

4. B outputs 1 in $\mathbf{Exp}^{\textsf{ind-cpa}}_{\textsf{sPKE}}$ if D_i outputs 1.

Next, note that B perfectly emulates $\mathbf{Exp}_{n+i-1}$ if it receives a challenge ciphertext e that encrypts b_i. On the other hand, B perfectly emulates $\mathbf{Exp}_{n+i}$ if e is the encryption of $(1-b_i)$. This ensures that the probability that B outputs 1 given an encryption of b_i is at least as good as the probability that D_i outputs 1 in $\mathbf{Exp}_{n+i-1}$. On the other hand, the probability that B outputs 1 given an encryption of $(1-b_i)$ is at least as good as the probability that D_i outputs 1 in $\mathbf{Exp}_{n+i}$. Denoting the bit picked in $\mathbf{Exp}^{\textsf{ind-cpa}}_{\textsf{sPKE}}$ by c we get,

$$\underbrace{\Pr\left[1 \leftarrow \mathsf{B}(pk, e) \mid (pk, sk) \leftarrow \mathsf{sGen}(1^k) \wedge e \leftarrow \mathsf{sEnc}_{pk}(b_i)\right]}_{=\Pr[1 \leftarrow \mathsf{B} \mid c=0]} \geq \epsilon_{n+i-1} \text{ and}$$

$$\underbrace{\Pr\left[1 \leftarrow \mathsf{B}(pk, e) \mid (pk, sk) \leftarrow \mathsf{sGen}(1^k) \wedge e \leftarrow \mathsf{sEnc}_{pk}(1-b_i)\right]}_{=\Pr[1 \leftarrow \mathsf{B} \mid c=1]} \geq \epsilon_{n+i}.$$

This implies that

$$\begin{aligned} \Delta_{\textsf{ind-cpa}} &= \mathbf{Adv}^{\textsf{ind-cpa}}_{\textsf{sPKE}}(\mathsf{B}, k) = 2\left|\Pr[\mathbf{Exp}^{\textsf{ind-cpa}}_{\textsf{sPKE}}(\mathsf{B}, k) = 1] - \frac{1}{2}\right| \\ &= \left|\Pr[1 \leftarrow \mathsf{B} \mid c = 0] - \Pr[1 \leftarrow \mathsf{B} \mid c = 1]\right| \\ &\geq \left|\epsilon_{n+i-1} - \epsilon_{n+i}\right|. \end{aligned}$$

□ □

Next, we show that PKE is neither a secure tweaked NCER nor a secure NCER.

Lemma 4.11. PKE *is neither a* $\{$**ind-tcipher**, **ind-tncer**$\}$ *secure tweaked NCER nor an* **ind-ncer** *secure NCER.*

Proof: First we show that PKE is not a secure tweaked NCER. Define the tweaked NCER $\mathsf{tPKE} = (\mathsf{tGen}, \mathsf{tEnc}, \mathsf{tEnc}^*, \mathsf{tDec}, \mathsf{tOpen})$ where $(\mathsf{tGen}, \mathsf{tEnc}, \mathsf{tDec})$ are the same as $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. We show that it is impossible to define algorithms tEnc^* and tOpen so that a ciphertext generated by tEnc^* can be opened into any plaintext from the binary plaintext space, by producing (possibly inefficiently) a matching secret key via tOpen. First, note that for any ciphertext $e \leftarrow \mathsf{tEnc}^*_{pk}$ produced with a public key pk so that $(pk, sk) \leftarrow \mathsf{tGen}(1^k)$, it must be that $e = (e_0, e_1)$ so that for a random α, $e_\alpha = \mathsf{sEnc}_{pk_\alpha}(0)$ and $e_{1-\alpha} = \mathsf{sEnc}_{pk_{1-\alpha}}(1)$, where $pk = (pk_0, pk_1)$. It is easy to see that if the above claim does not hold then e cannot be decrypted to both 0 and 1. Specifically, tOpen can produce a secret key that decrypts e into a bit b where pk_b is selected using sGen. However, in order to produce a secret key that decrypts e into $1-b$, tOpen must find a secret key that matches pk_{1-b}, which was picked from $\mathcal{K}$. Nevertheless, it may be that pk_{1-b} is an invalid public key, that is computationally indistinguishable from a valid public key, but does not have a matching secret key (namely, when $\mathcal{V} \subset \mathcal{K}$). Thus, producing a secret key in this case is impossible regardless of the computational power tOpen is given. Our proof now follows from the fact that the probability that

pk_{1-b} is an invalid key is non-negligible in k (recall that secure extended key-simulatable PKE ensures this property) and so tPKE does not satisfy **ind-tcipher** security which concludes the first part of the proof. The proof claiming that PKE is not a secure NCER follows due to similar reasons where we show that **ind-ncer** security cannot be achieved. □

4.4.1 Realizing Key-Simulatable and Extended Key-Simulatable PKE

A simple example of a $\{$**ind-cpa**, **ksim**$\}$ secure key-simulatable PKE is the ElGamal PKE [Gam85] where we set $\mathcal{K}$ to be equal to the set of *valid* public keys, i.e. $\mathcal{K} = \mathcal{V}$. In addition, note that any simulatable PKE as defined in [DN00] is also $\{$**ind-cpa**, **ksim**$\}$ secure key-simulatable PKE.

Below we provide an example of extended key-simulatable PKE with security under the DDH assumption. For simplicity we consider a binary plaintext space. Let $(g_0, g_1, p) \leftarrow \mathcal{G}(1^k)$ be an algorithm that given a security parameter k returns a group description $\mathbb{G} = \mathbb{G}_{g_0,g_1,p}$ specified by its generators g_0, g_1 and its order p. Furthermore, we set $\mathcal{K} = \mathbb{G}^2$ and $\mathcal{V} = \{(g_0^x, g_1^x) \in \mathbb{G}^2 \mid x \in \mathbb{Z}_p\}$. Then define the following extended key-simulatable PKE,

– sGen, given the security parameter k, set $(g_0, g_1, p) \leftarrow \mathcal{G}(1^k)$. Choose uniformly random $x \leftarrow \mathbb{Z}_p$ and compute $h_i = g_i^x$ for all $i \in \{0, 1\}$. Output the secret key $sk = x$ and the public key $pk = (h_0, h_1)$.

– sEnc, given the public key pk and plaintext $m \in \{0, 1\}$, choose a uniformly random $s, t \leftarrow \mathbb{Z}_p$. Output the ciphertext $\left(g_0^s g_1^t, g_0^m \cdot (h_0^s h_1^t)\right)$.

– sDec, given the secret key x and ciphertext (g_c, h_c), output $h_c \cdot (g_c^x)^{-1}$.

– $\widetilde{\mathsf{sGen}}$, given 1^k, output two random elements from $\mathbb{G}$ and their bit sequence as the randomness.

– $\widetilde{\mathsf{sGen}}^{-1}$, given a legitimate public key h_0, h_1, simply returns the bit strings of h_0, h_1 as the randomness used to sample them from $\mathbb{G}^2$ by $\widetilde{\mathsf{sGen}}$.

We remark that a public key chosen randomly from $\mathbb{G}^2$ does not necessarily correspond to a secret key. Furthermore, $\Pr\left[pk \in \mathcal{K} \setminus \mathcal{V} \mid pk \leftarrow \widetilde{\mathsf{sGen}}(1^k)\right]$ is non-negligible. This is a key property in our proof from Section 4.4 that enables us to prove that **rsim-so** does not imply NCER as well as tweaked NCER, which further implies that **rind-so** does not imply these primitives.

4.5 Realizing Tweaked NCER

4.5.1 Constructions for Polynomial Plaintext Spaces

Based on key-simulatable PKE. We prove that secure tweaked NCER can be built based on any secure key-simulatable PKE with $\mathcal{K} = \mathcal{V}$ (cf. definition 3.4). Specifically, our construction is based on the separating scheme presented in Section 4.4. In addition, we define the fake encryption algorithm so that it outputs two ciphertexts that encrypt two distinct plaintexts rather than the same plaintext twice (implying that ciphertext indistinguishability follows from the **ind-cpa** security of the underlying encryption scheme). More formally, the fake encryption algorithm can be defined as follows. Given $sk = (\alpha, sk_\alpha, r_{1-\alpha})$ and message b, a fake encryption of b is computed by $e^* = (\mathsf{sEnc}_{pk_0}(b), \mathsf{sEnc}_{pk_1}(1-b))$ if $\alpha = 0$ and $e^* = (\mathsf{sEnc}_{pk_0}(1-b), \mathsf{sEnc}_{pk_1}(b))$ otherwise. It is easy to verify that given sk, the decryption of e^* returns b and that e^* is computationally indistinguishable from a valid encryption even given the secret key. Next, we discuss the details of the non-efficient opening algorithm which is required to generate a secret key for a corresponding public key given a fake ciphertext and a message b'. In more details, assuming

$sk = (\alpha, sk_\alpha, r_{1-\alpha})$ and $pk = (pk_0, pk_1)$,

$$\mathsf{tOpen}(sk, pk, (e_0^*, e_1^*), b') = \begin{cases} (\alpha, sk_\alpha, r_{1-\alpha}) & \text{if } e_\alpha^* = \mathsf{sEnc}_{pk_\alpha}(b') \\ (1-\alpha, sk_{1-\alpha}, r_\alpha) & \text{otherwise, where } r_\alpha \leftarrow \widetilde{\mathsf{sGen}}^{-1}(pk_\alpha) \text{ and} \\ & sk_{1-\alpha} \text{ is a valid secret key of } pk_{1-\alpha}. \end{cases}$$

Note that since it holds that $\mathcal{V} = \mathcal{K}$ for the underlying sPKE scheme, there exists a secret key that corresponds to $pk_{1-\alpha}$ and it can be computed (possibly in an inefficient way). Encryption schemes for larger plaintext spaces can be obtained by repeating this basic scheme sufficiently many times.[6] Finally, we note that the scheme is $\{\textbf{ind-tcipher}, \textbf{ind-tncer}\}$ secure. Recalling that any simulatable PKE with $\mathcal{K} = \mathcal{V}$ is a key-simulatable PKE [DN00, CDSMW09], we conclude that secure tweaked NCER for a binary plaintext space can be built relying on DDH, RSA, factoring and LWE assumptions.

Based on statistically-hiding $\binom{2}{1}$-OT. Let $(\mathsf{Sen}, \mathsf{Rec})$ be a two-round honest-receiver statistically-hiding $\binom{2}{1}$ oblivious transfer protocol (cf. Definition 3.6). Then, we show how to construct a tweaked NCER. Intuitively, we view the first message of the protocol as the public key, whereas the second message is viewed as a ciphertext that encrypts one of the sender's inputs (since the receiver only learns one of these inputs). Moreover, a valid encryption algorithm employs the sender with the same input twice, whereas a fake encryption algorithm invokes the sender with an input and its compliment, ensuring that there is a secret key that decrypts into each one of these inputs by the statistical security of the receiver. More formally, we define algorithms $(\mathsf{tGen}, \mathsf{tEnc}, \mathsf{tEnc}^*, \mathsf{tDec}, \mathsf{tOpen})$ as follows.

- tGen, given the security parameter 1^k, set $(\mathsf{q}, sk) \leftarrow \mathsf{Rec}_q(1^k, \alpha)$ for a random $\alpha \leftarrow \{0,1\}$. Set $pk = \mathsf{q}$, and sk to be the sk returned by Rec_q.
- tEnc, given the public key q and plaintext $m \in \{0,1\}$, output $\mathsf{Sen}(\mathsf{q}, m, m)$.
- tDec, given the secret key sk and ciphertext $c = \mathsf{rsp}$, output $\mathsf{Rec}_r(sk, \mathsf{rsp})$.
- tEnc^*, given the secret key sk, public key q and plaintext $m \in \{0,1\}$, output $\mathsf{Sen}(\mathsf{q}, m, 1-m)$ if $\alpha = 0$. Otherwise, output $\mathsf{Sen}(\mathsf{q}, 1-m, m)$. Note that given the pair (pk, sk) it is possible to learn α. Furthermore, $\mathsf{tDec}(sk, \mathsf{rsp}^*) = m$ where rsp^* is a fake ciphertext since fake encryption follows according to the receiver's input α.
- tOpen, given the secret key sk, public key pk, fake ciphertext rsp^* and plaintext m, output sk' that is consistent with q. Note that such sk' exists since the receiver's input α is statistically hidden given q.

It is easy to verify that ciphertext indistinguishability holds due to the sender's privacy property. Furthermore, any two fake ciphertexts with sender's input $s_0 = 0$ and $s_1 = 1$, or $s_0 = 1$ and $s_1 = 0$, are identically distributed where the probability distribution is over the randomness for tEnc^* and the choice of α. Finally, since we it holds that simulatable PKE implies statistically-hiding OT, it implies that secure tweaked NCER is obtained from the same assumptions that are specified earlier.

[6] We note that this construction was discussed in [HLAWW13] in the context of weak hash proof systems and leakage resilient PKE.

4.5.2 Constructions for Exponential Plaintext Spaces

Based on NCER. We show that the DCR based secure NCER of [CHK05] is also a secure tweaked NCER. Specifically, let $(p', q') \leftarrow \mathcal{G}(1^n)$ be an algorithm that given a security parameter k returns two random n bit primes p' and q' such that $p = 2p' + 1$ and $q = 2q' + 1$ are also primes. Let $N = pq$ and $N' = p'q'$. Define $(\mathsf{tGen}, \mathsf{tEnc}, \mathsf{tEnc}^*, \mathsf{tDec}, \mathsf{tOpen})$ as follows.

- tGen, given the security parameter k, run $(p', q') \leftarrow \mathcal{G}(1^n)$ and set $p = 2p' + 1$, $q = 2q' + 1$, $N = pq$ and $N' = p'q'$. Choose random $x_0, x_1 \leftarrow \mathbb{Z}_{N^2/4}$ and a random $g' \in \mathbb{Z}^*_{N^2}$ and compute $g_0 = g'^{2N}$, $h_0 = g_0^{x_0}$ and $h_1 = g_0^{x_1}$. Output public key $pk = (N, g_0, h_0, h_1)$ and secret key $sk = (x_0, x_1)$.
- tEnc, given the public key pk and a plaintext $m \in \mathbb{Z}_N$, choose a uniformly random $t \leftarrow \mathbb{Z}_{N/4}$ and output ciphertext
$$c \leftarrow \mathsf{tEnc}_{pk}(m; t) = \left(g_0^t \bmod N^2, (1+N)^m h_0^t \bmod N^2, h_1^t \bmod N^2\right).$$
- tDec, given the secret key (x_0, x_1) and a ciphertext (c_0, c_1, c_2), check whether $c_0^{2x_1} = (c_2)^2$; if not output $\perp$. Then set $\hat{m} = (c_1/c_0^{x_0})^{N+1}$. If $\hat{m} = 1 + mN$ for some $m \in \mathbb{Z}_N$, then output m; else output $\perp$.
- tEnc^*, given the public key pk, secret key sk and a message m, choose uniformly random $t \leftarrow \mathbb{Z}_{\phi(N)/4}$, compute the fake ciphertext
$$c^* \leftarrow (c_0^*, c_1^*, c_2^*) = \left((1+N) \cdot g_0^t \bmod N^2, (1+N)^m \cdot (c_0^*)^{x_0} \bmod N^2, (c_0^*)^{x_1} \bmod N^2\right).$$
- tOpen, given N', (x_0, x_1), a ciphertext (c_0, c_1, c_2) such that $(c_0, c_1, c_2) \leftarrow \mathsf{tEnc}^*_{pk}(sk, m)$ and a plaintext $m^* \in \mathbb{Z}_N$, output $sk^* = (x_0^*, x_1)$, where $x_0^* \leftarrow \mathbb{Z}_{NN'}$ is the unique solution to the equations $x_0^* = x \bmod N'$ and $x_0^* = x_0 + m - m^* \bmod N$. These equations have a unique solution due to the fact that $gcd(N, N') = 1$ and the solution can be obtained employing Chinese Remainder Theorem.

  It can be verified that the secret key sk^* matches the public key pk and also decrypts the 'simulated' ciphertext to the required message m^*. The first and third components of pk remain the same since x_1 has not been changed. Now $g^{x_0^*} = g^{x_0^* \bmod N'} = g^{x_0 \bmod N'} = g^{x_0} = h_0$. Using the fact that the order of $(1+N)$ in $\mathbb{Z}^*_{N^2}$ is N, we have
$$\begin{aligned}\left(\frac{c_1}{c_0^{x_0^*}}\right)^{N+1} &= \left(\frac{(1+N)^{x_0+m} g_0^{tx_0}}{(1+N)^{x_0^*} g_0^{tx_0^*}}\right)^{N+1} \\ &= \left((1+N)^{x_0+m-x_0^* \bmod N}\right)^{N+1} = ((1+N)^m)^{N+1} = (1+mN).\end{aligned}$$

It is easy to verify that real and fake ciphertexts are computationally indistinguishable under the DCR assumption since the only difference is with respect to the first element (which is an $2N$th power in a real ciphertext and not an $2N$th power in a simulated ciphertext). The other two elements are powers of the first element. Furthermore $sk = (x_0, x_1)$ and $sk^* = (x_0^*, x_1)$ are statistically close since $x_0 \leftarrow \mathbb{Z}_{N^2/4}$ and $x_0^* \leftarrow \mathbb{Z}_{NN'}$ and the uniform distribution over $\mathbb{Z}_{NN'}$ and $\mathbb{Z}_{N^2/4}$ is statistically close.

Based on HPS. Finally, we demonstrate that HPS (cf. Definition 3.7) imply tweaked NCER. Let $\mathsf{HPS} = (\mathsf{Param}, \mathsf{Pub}, \mathsf{Priv})$ denote a secure HPS. We define a secure tweaked NCER as follows.

- tGen, given the security parameter 1^k, invoke $\mathsf{Param}(1^k)$ and generate instances of $(\mathbb{G}, \mathcal{K}, \mathcal{C}, \mathcal{V}, \mathcal{SK}, \mathcal{PK}, \Lambda_{(\cdot)}, \delta(\cdot))$. Choose a random $sk \in \mathcal{SK}$ and let $pk = \delta(sk) \in \mathcal{PK}$. Output the pair (pk, sk).
- tEnc, given the public key pk and plaintext $m \in \mathcal{M}$, choose a random $c \leftarrow \mathcal{V}$ together with a corresponding witness w for c being a valid ciphertext. Let $e = \mathsf{Pub}_{pk}(c, w) \oplus m$, then output the ciphertext (c, e).
- tDec, given the secret key sk and ciphertext (c, e), output $m = e \oplus \mathsf{Priv}_{sk}(c)$.
- tEnc^*, given the secret key sk and plaintext $m \in \mathcal{M}$, choose a random $c^* \leftarrow \mathcal{C}/\mathcal{V}$. Let $e^* = \mathsf{Priv}_{sk}(c^*) \oplus m$, then output the ciphertext (c^*, e^*). It is easy to verify that the decryption of a fake ciphertext (c^*, e^*) outputs the encrypted plaintext m.
- tOpen, given the secret key sk and the public key pk, fake ciphertext (c^*, e^*) and plaintext $m \in \mathcal{M}$ output sk^* that decrypts this ciphertext into m (the existence of such a secret key is implied by the 1-universality property of HPS).

First, ciphertext indistinguishability holds due to the **ind-hps** property of the HPS. In addition, the 1-universal property ensures that encapsulated keys that were generated by $\mathsf{Priv}_{sk}(c^*)$ for $c^* \leftarrow \mathcal{C}/\mathcal{V}$ are identically distributed to encapsulated keys that are equivocated by algorithm tOpen.

5 Selective Opening Security for the Sender

In this section we prove **sind-so** is strictly weaker than **ssim-so** security by constructing a scheme that meets the former but not the latter level of security. We prove this statement by constructing a **sind-so** secure encryption scheme that is not **ssim-so** secure. Our starting point is a lossy encryption scheme $\mathsf{loPKE} = (\mathsf{loGen}, \mathsf{loGen}^*, \mathsf{loEnc}, \mathsf{loDec})$. We then modify loPKE by adding a (statistically hiding) commitment to each ciphertext such that the new scheme, denoted by PKE, becomes committing. Next, we prove that PKE is **sind-so** secure by showing that the scheme remains lossy and is therefore **sind-so** secure according to [BHY09]. Finally, using the result from [BDWY12] we claim that PKE is not **ssim-so** secure. Our separating scheme requires that the message space $\mathcal{M}_{\mathsf{nisCom}}$ of nisCom and the message space $\mathcal{M}_{\mathsf{loPKE}}$ of loPKE are the same (since we encrypt and commit to the same message). We formalize this requirement and define compatibility between the above primitives as follows:

Definition 5.1. *Assume that* loPKE *and* nisCom *are* $\{\textbf{ind-lossy}, \textbf{ind-lossycipher}\}$ *secure lossy PKE and* $\{\textbf{stat-hide}, \textbf{comp-bind}\}$ *secure NISHCOM, respectively. We say that these primitives are* compatible *if* $\mathcal{M}_{\mathsf{loPKE}} = \mathcal{M}_{\mathsf{nisCom}}$.

We proceed with our main theorem for this section and further provide a concrete example of schemes that satisfy the compatibility criteria.

Theorem 5.2. *Assume there exists a* $\{\textbf{ind-lossy}, \textbf{ind-lossycipher}\}$ *secure lossy PKE and a* $\{\textbf{stat-hide}, \textbf{comp-bind}\}$ *secure NISHCOM that are compatible. Then, there exists a PKE that is* **sind-so** *secure but is not* **ssim-so** *secure.*

```
$\mathsf{Gen}(1^k)$
    $(pk, sk) \leftarrow \mathsf{loGen}(1^k)$
    Return $(pk, sk)$
```

```
$\mathsf{Gen}^*(1^k)$
    $(pk^*, sk^*) \leftarrow \mathsf{loGen}^*(1^k)$
    Return $(pk^*, sk^*)$
```

```
$\mathsf{Enc}_{pk}(m)$
    $e_0 \leftarrow \mathsf{loEnc}_{pk}(m)$
    $e_1 \leftarrow \mathsf{nisCommit}(m)$
    Return $e = (e_0, e_1)$
```

```
$\mathsf{Dec}_{sk}(e)$
    $e := (e_0, e_1)$
    $m = \mathsf{loDec}_{sk}(e_0)$
    Return $m$
```

Proof: We describe our separating encryption scheme first. Specifically, given a secure lossy PKE $\mathsf{loPKE} = (\mathsf{loGen}, \mathsf{loGen}^*, \mathsf{loEnc}, \mathsf{loDec})$ and a secure NISHCOM $(\mathsf{nisCommit}, \mathsf{nisOpen})$, we define our separating scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Gen}^*, \mathsf{Enc}, \mathsf{Dec})$ as follows.

The proof follows from Lemmas 5.1 and 5.2 presented and proven below. ■

Lemma 5.1. PKE *is* **sind-so** *secure.*

Proof: According to [BHY09], any lossy encryption satisfies **sind-so** security. We thus prove that PKE is a lossy encryption scheme. It is easy to verify that correctness on real keys and indistinguishability of real and lossy keys follow due to the underlying lossy encryption scheme loPKE. Next, lossyness under lossy keys follows from the lossyness under lossy keys of loPKE and the statistical hiding property of NISHCOM. Namely, since NISHCOM satisfies statistical hiding there exists a (possibly inefficient) algorithm that given a message $m \in \mathcal{M}_{\mathsf{nisCom}}$ and commitment c outputs randomness $r \in \mathcal{R}_{\mathsf{nisCom}}$ such that $c = \mathsf{nisCommit}(m; r)$, where $\mathcal{R}_{\mathsf{nisCom}}$ is the randomness space of the commitment scheme. Therefore, one can define an algorithm loOpen for PKE (as required for a lossy encryption), that is combined of two sub-algorithms: (1) algorithm loOpen of loPKE which exists under the assumption that loPKE is a lossy encryption, and (2) the algorithm specified above for the commitment scheme. This implies that PKE is a lossy encryption and concludes the proof. □

Lemma 5.2. PKE *is not* **ssim-so** *secure.*

Proof: We note that PKE is a binding scheme in the sense of [BDWY12] due to the use of a commitment scheme. Specifically, explaining a ciphertext in two different ways (i.e., generating two random strings for two different messages), implies breaking the computational binding property of $\mathsf{nisCommit}$. By applying the result of [BDWY12] that no binding scheme is **ssim-so**, the proof is concluded. □

5.1 Compatible Secure Lossy PKE and Secure NISHCOM

Our construction requires secure lossy PKE and a secure NISHCOM that share the same plaintext space. One potential instantiation is to consider the lossy PKE from [HLOV11] that is defined based on the rerandomizable encryption El Gamal encryption scheme specified by the algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, that is defined relative to a group $\mathbb{G}$ of prime order p (where the plaintext space of polynomial size might also be a subset of $\mathbb{Z}_p$). Specifically, the key generation algorithm for the lossy encryption scheme invokes Gen and generates a pair of keys (pk_{EG}, sk_{EG}), and fixes $pk = (\mathsf{Enc}_{pk_{EG}}(0), \mathsf{Enc}_{pk_{EG}}(1))$, whereas $sk = sk_{EG}$. Moreover, given a public key (c_0, c_1) an encryption of the bit $b \in \{0, 1\}$ is defined by a rerandomization of ciphertext c_b, whereas decryption follows by decrypting the ciphertext using sk as in El Gamal. Finally, the lossy key generation algorithm is defined by fixing $pk^* = (\mathsf{Enc}_{pk_{EG}}(0), \mathsf{Enc}_{pk_{EG}}(0))$.

In addition, we instantiate the NISHCOM with Pedersen commitment scheme [Ped91] which can operate over a polynomial size subset of $\mathbb{Z}_p$, i.e. the same message space of the above secure lossy PKE. More concretely, let $\{\mathbb{G}_k, g_k, h_k\}_{k \in \mathbb{N}}$ be a family of finite groups, along with fixed generators g_k, h_k of $\mathbb{G}_k$, that are all parameterized by the security parameter. Then, assuming the hardness of computing $\log_{g_k} h_k$, Pedersen's commitment, defined by $\mathsf{nisCommit}(m; r) := g_k^m h_k^r$, is a secure NISHCOM where **comp-bind** security holds due to the hardness of discrete log assumption.

Acknowledgements

Carmit Hazay acknowledges support from the Israel Ministry of Science and Technology (grant No. 3-10883). Arpita Patra acknowledges support from project entitled 'ISEA - Part II' funded by Department of Electronics and Information Technology (DeitY) of Govt. of India. Part of this work was carried out while Bogdan Warinschi was a visiting Microsoft Research, Cambridge, UK and IMDEA, Madrid, Spain. He has been supported in part by ERC Advanced Grant ERC-2010-AdG-267188-CRIPTO, by EPSRC via grant EP/H043454/1, and has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 609611 (PRACTICE).

References

[BDWY12] Mihir Bellare, Rafael Dowsley, Brent Waters, and Scott Yilek. Standard security does not imply security against selective-opening. In *EUROCRYPT*, pages 645–662, 2012.

[BHK12] Florian Böhl, Dennis Hofheinz, and Daniel Kraschewski. On definitions of selective opening security. In *Public Key Cryptography*, pages 522–539, 2012.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT*, pages 1–35, 2009.

[BWY11] Mihir Bellare, Brent Waters, and Scott Yilek. Identity-based encryption secure against selective opening attack. In *TCC*, pages 235–252, 2011.

[BY09] Mihir Bellare and Scott Yilek. Encryption schemes secure under selective opening attack. *IACR Cryptology ePrint Archive*, 2009:101, 2009.

[CDSMW09] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Improved non-committing encryption with applications to adaptively secure protocols. In *ASIACRYPT*, pages 287–302, 2009.

[CFGN96] Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *STOC*, pages 639–648, 1996.

[CHK05] Ran Canetti, Shai Halevi, and Jonathan Katz. Adaptively-secure, non-interactive public-key encryption. In *TCC*, pages 150–168, 2005.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64, 2002.

[DN00] Ivan Damgård and Jesper Buus Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In *CRYPTO*, pages 432–450, 2000.

[DN02] Ivan Damgård and Jesper Buus Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In *CRYPTO*, pages 581–596, 2002.

[DN03] Ivan Damgård and Jesper Buus Nielsen. Universally composable efficient multiparty computation from threshold homomorphic encryption. In *CRYPTO*, pages 247–264, 2003.

[DNRS03] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. *J. ACM*, 50(6):852–921, 2003.

[FHKW10] Serge Fehr, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Encryption schemes secure against chosen-ciphertext selective opening attacks. In *EUROCRYPT*, pages 381–402, 2010.

[Gam85] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[HLAWW13] Carmit Hazay, Adriana López-Alt, Hoeteck Wee, and Daniel Wichs. Leakage-resilient cryptography from minimal assumptions. In *EUROCRYPT*, pages 160–176, 2013.

[HLOV11] Brett Hemenway, Benoît Libert, Rafail Ostrovsky, and Damien Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In *ASIACRYPT*, pages 70–88, 2011.

[HLQ13] Zhengan Huang, Shengli Liu, and Baodong Qin. Sender-equivocable encryption schemes secure against chosen-ciphertext attacks revisited. In *Public Key Cryptography*, pages 369–385, 2013.

[HR14] Dennis Hofheinz and Andy Rupp. Standard versus selective opening security: Separation and equivalence results. In *TCC*, pages 591–615, 2014.

[JL00] Stanislaw Jarecki and Anna Lysyanskaya. Adaptively secure threshold cryptography: Introducing concurrency, removing erasures. In *EUROCRYPT*, pages 221–242, 2000.

[Nie02] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, pages 111–126, 2002.

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO*, pages 129–140, 1991.

[ROV14] Vanishree Rao R. Ostrovsky and Ivan Visconti. On selective-opening sattacks against encryption schemes. In *SCN*, 2014.