Simulatability versus Indistinguishability SOA: CCA Relations are Sampler-Dependent

Hans Heum

Norwegian University of Science and Technology, Trondheim, Norway

Abstract. Selective Opening Attack (SOA) is a set of security notions modelling the threat of encryption randomness and/or secret keys leaking after-the-fact. Contrary to expectation, we show that there is no general reduction from simulation-based selective opening security (SSO) to indistinguishability-based selective opening security (ISO) in the CCA setting. In particular, we show that when restricted to certain message distributions, SSO-CCA is incomparable with ISO-CCA. This contrasts the CPA case, where SSO-CPA is known to be strictly stronger than ISO-CPA relative to any message sampler. Any sampler with high enough min-entropy suffices for this "semi-separation" to appear. On the other hand, we show that restricting to distributions with very low min-entropy gives rise to an implication.
Our main result does not rely on the presence of selective openings, but rather follow from subtleties in the game structures. At a glance, this may seem to contradict known equivalences between indistinguishability, semantic security, and selective opening security under trivial openings. We reconcile the apparent contradiction by showing that the CCA landscape splits into a "high-entropy" and a "low-entropy" world, which for notions of SOA must be treated separately.

Keywords: Public-Key Encryption · Selective Opening Attacks · CCA

1 Introduction

In realistic settings, secret material such as private keys and the randomness used for encryption may leak. Intuitively, one expects that such leakage should have no impact on the security of unrelated encryptions. *Proving* this intuition, however, has proven challenging, and several generalized security notions have appeared with the goal of capturing such scenarios. Though they aim to model the same real-world scenario, these notions vary wildly in strength, from multi-user indistinguishability with adaptive corruption (equivalent to standard IND-CCA up to a multiplicative loss in the number of users [BBM00, BJLS16, HS21]), to the standard-model-unachievable notion of non-committing encryption (NCE) [Nie02].

In the middle of this landscape sit the various notions of security under Selective-Opening Attacks (SOA) [DNRS99]. One variant, known as Indistinguishability under Selective-Opening attacks (ISO) [BHY09], samples and encrypts one message per challenge, an arbitrary number of which the adversary may adaptively *open* to learn either just the message (transmission opening), the message and the encryption randomness (sender opening), or in the case of multiple users, a user's private key (receiver opening). Once the adversary is satisfied, the game enters a *challenge phase* in which the adversary is given either the full list of sampled messages, or a list of conditionally resampled messages,

Published in Communications in Cryptology, Vol. 2, No. 4. This is the author's version.
E-mail: hans@heum.me (Hans Heum)

This work is licensed under a "CC BY 4.0" license.
Date of this document: 2026-01-09.

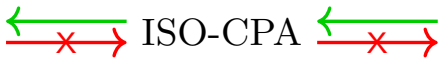


Figure 1: The hierarchy of SOA notions under randomness and/or private-key leakage, known to hold in the CPA setting, and expected to also hold in the CCA setting.

conditioned on consistency with the opened messages. Crucially, sampling is *stateful*, ensuring that messages may depend arbitrarily on each other [BY12].

A second variant is found in Simulatability under Selective-Opening attacks (SSO). Drawing inspiration from the classic notion of semantic security [GM82, Gol01], the SSO experiment initially proceeds as above, except instead of a challenge phase, the adversary halts with some arbitrary output out. A distinguisher (who is given the list of sampled messages but not their encryptions) is then tasked with guessing whether out was generated by an adversary, or by some simulator that learned nothing about the messages beyond their lengths.

Early works realized that notions of SOA are not implied by standard indistinguishability [BHY09], at least as long as opening reveals some information beyond just the sampled messages [BY12]. Many works have since been working to map out the relations between the various SOA notions, and between notions of SOA and indistinguishability [BDWY12, HR14, HPW15, FHKP16, HRW16]. From these relations, a picture gradually formed, where SSO sat as the strongest notion, followed by ISO, and with standard indistinguishability (IND) sitting at the bottom.[1] For the most part, each level of this emerging hierarchy came with a "downward" implication and an "upward" separation, and there seemed little reason to believe that any of the remaining relations should break this pattern (see Figure 1).

Recently, Brunetta, Heum and Stam [BHS24], hereafter BHS, provided a systematization, mapping out the known relations and showing that the notions indeed fall into a strict hierarchy in the CPA setting. (They additionally placed non-committing encryption as the fourth level of the hierarchy, above SSO.) However, they pointed out that in the CCA setting, several relations still only held the status of conjecture, and that attempts at lifting CPA reductions to the CCA setting ran into problems. This included the crucial implication from SSO-CCA to ISO-CCA, without which the idea of a strict and tidy CCA hierarchy to match the CPA one would crumble.

1.1 Our Contributions

We show that there does not exist a general implication from SSO-CCA to ISO-CCA (Theorem 1), dealing a potentially fatal blow to the idea of a strict CCA hierarchy. Our main result is a "semi"-separation: we show that when restricting to high-entropy message distributions, there is a PKE scheme that satisfies SSO-CCA but is trivially broken in the ISO-CCA game. Our result is a "semi"-separation, in that the separating scheme only achieves SSO-CCA security relative to high-entropy message distributions. Still, this suffices to rule out implications that hold for all message samplers simultaneously.

We then show that, when restricting to distributions with sufficiently low entropy, SSO-CCA *does* imply ISO-CCA (Theorem 2). The result is that relations among CCA notions of SOA, rather than being clean-cut like for CPA, must now be treated as *distribution-dependent*. The results are illustrated in Figure 2.

[1] The requirement for a notion of SOA to be a stronger notion than IND seems to be that the adversary is given the ability to *verify* openings. If, for instance, the adversary learned (the message and) the random coins used in encryption, then they can verify that re-encrypting the message using the coins yields the original ciphertext; likewise when learning a private key, the adversary can check its validity by decrypting. Sans this ability, a reduction can, so to speak, just lie. (See e.g. Theorem 4.)



Figure 2: Our two main results, and the resulting CCA "hierarchy": Theorem 1 shows a semi-separation from SSO-CCA to ISO-CCA, while Theorem 2 shows a partial implication.

1.1.1 SSO-CCA Does Not Imply ISO-CCA for High-Entropy Distributions

We first show that the issues encountered when trying to reduce SSO-CCA to ISO-CCA are unavoidable by showing that there exists a PKE scheme and a class of message samplers such that, relative to such samplers, the PKE scheme achieves SSO-CCA security under any opening, yet is trivially ISO-CCA insecure.

In a nutshell, the issue one encounters when trying to reduce SSO-CCA to ISO-CCA is that an ISO adversary expects to have access to a decryption oracle also in the challenge phase, for an adaptive, "CCA2" style notion of chosen-ciphertext attacks. However, in the SSO game, the challenge phase is replaced with a distinguishing phase, in which a distinguisher is tasked with distinguishing a real input from a simulated one. In order to reduce to ISO-CCA, this distinguisher must simulate the challenge phase for the ISO adversary, including a decryption oracle, without access a decryption oracle of its own.[2]

The main idea underlying the semi-separation of Theorem 1 is that of a plaintext-checking attack (PCA): access to a PCA oracle trivially breaks any notion of indistinguishability, as once the adversary knows that the challenge message is one of only a few options, they can simply ask the PCA oracle which one is the right one. However, *provided* the sampled challenge messages are of sufficiently high min-entropy, this cannot help an SSO adversary, who will have only a negligible chance of guessing any of the challenges messages.

Our separating scheme, presented in Section 3, exploits this gap by providing the adversaries in both experiments with a plaintext-checking oracle embedded in the decryption oracle, yet denying the SSO distinguisher the same luxury (it does not get access to the decryption oracle). In Lemma 2, we prove that relative to any high-entropy message sampler, the separating scheme achieves SSO-CCA security for arbitrary openings; and yet the ISO attack (Lemma 3) does not rely on any oracles other than the decryption oracle, separating even the strongest notion of SSO-CCA (with both sender and receiver opening) from the weakest notion of ISO-CCA (without any opening).

1.1.2 SSO-CCA Does Imply ISO-CCA for Low-Entropy Distributions

With our semi-separation holding relative to all message samplers of sufficiently high min-entropy, the question arises whether restricting to low-entropy message samplers opens for an implication. Upon reflection, one may realize that there must be at least *one* non-trivial message sampler relative to which an implication holds: SSO-CCA can be shown to imply IND-CCA (by lifting a reduction due to Bellare and Yilek [BY12] to the CCA setting), with the reduction employing a left-or-right sampler that imitates the standard indistinguishability game (shown in Figure 3). Further, Heuer showed that for a large class of message samplers which includes this left-or-right sampler, IND-CCA implies ISO-CCA with sender opening [Heu17]. Stringing the results together gives a partial implication from SSO-CCA to ISO-CCA, but only relative to this one sampler. Can we do better?

[2] If we instead were to consider a notion of "ISO-CCA1" wherein the adversary gets access to the decryption oracle in the game phase but not in the challenge phase, then the reduction would go through as expected, and the hierarchy would be recovered.

In Section 4 we answer this question in the affirmative. Defining a notion of a (very) "low-entropy sampler" as one whose output depends on at most a logarithmic number of random bits *even after polynomially many samples* (Definition 6), we show in Theorem 2 that restricting to such samplers opens for a strategy in which the SSO adversary can provide the distinguisher with everything it needs to faithfully simulate a decryption oracle for the ISO adversary, circumventing the above barrier.

The strategy goes as follows: after simulating the initial phase of the ISO experiment for the adversary, the reduction *also* simulates the challenge phase providing the adversary with a list of resampled messages. Ignoring its output, the reduction instead stores transcripts of the calls made to the decryption oracle. It repeats this procedure a polynomial number of times, then forwards all the resampled message lists and their corresponding oracle transcripts to the distinguisher. If one of the resampled lists happened to be equal to the real list (which the distinguisher can recognize), then the distinguisher can faithfully simulate the challenge phase for the (second half of the) ISO adversary—including the decryption oracle, as due to a derandomization argument, it is guaranteed that re-running the adversary on the same input will lead to the same oracle queries, for which the distinguisher were already given the answers.

As a useful sanity check, note how the implication captures the case of the left-or-right sampler: just like the standard indistinguishability game, the left-or-right sampler only ever draws a single challenge bit, and so its output only depends on a single bit of internal randomness, regardless of the number of samples. This clearly satisfies the (admittedly strict) notion of a low-entropy sampler, and after only two resamplings in expectation, the above reduction will already have hit upon the real challenge bit, enabling the distinguisher's simulation.

1.1.3 A New CCA Landscape

One question remains, namely, with our semi-separation holding even without openings, and with IND-CCA known to imply ISO-CCA with transmission opening relative to all samplers [BHS24] (which we further extend to SSO-CCA in Theorem 4), does this not contradict the well-known equivalence [WSI03, Gol01] of semantic security, which is just SSO without opening oracles, and standard indistinguishability? In Section 5, we resolve this apparent contradiction by showing that, *relative to high-entropy samplers*, IND-CCA is not implied by SSO-CCA (Theorem 3), highlighting the danger of carelessly stringing together implications formulated relative to message samplers.

We end by providing a new map of the CCA landscape (Figure 12), now split into four main worlds, from the world of high-entropy samplers and non-trivial openings, where all three notions are incomparable, to the world of low-entropy samplers and trivial openings, where all three notions turn equivalent.

1.2 Related Work

This work was directly inspired by Hofheinz and Rupp [HR14], who separated indistinguishability SOA (under sender opening) from standard indistinguishability in the CCA setting. Like us, they reached their separation result by exploiting the presence of a CCA oracle to mount a distinguishing attack.

As already mentioned, there is a large body of literature studying relations between notions of SOA; BHS provides an exhaustive survey. One work worth highlighting is that of Bellare and Yilek [BY12], on which much of our formalism is based. This eprint is a greatly expanded full version of an earlier merged paper with Hofheinz [BHY09], and appears to have largely flown under the radar of the SOA community. In it, they introduce the formalism of stateful message samplers, thus achieving the best-of-both-worlds in terms

of maximum adaptivity and arbitrary message relations. They are also the first to study notions of SOA under transmission opening.

The present work builds directly on the SoK (Systematization of Knowledge) of BHS (Brunetta, Heum and Stam [BHS24]), who listed the main question tackled herein as one of many interesting open problems across the landscape of SOA notions, most of which remain to be explored.

2 Preliminaries

2.1 Notation

We take the natural numbers to be $\mathbb{N} = \{1, 2, \ldots\}$. List indices start at 1, and $[n]$ denotes the set $\{1, 2, \ldots, n\}$. For a set $\mathcal{I} \subset \mathbb{N}$ of indices and a list $\mathtt{A}$ of elements x_i, we let $\mathtt{A}[\mathcal{I}]$ denote the sub-list $(x_i)_{i\in\mathcal{I}}$. In a slight abuse of notation, we denote the list enumerating n objects x_i as $[x_n] \coloneqq (x_i)_{i\in[n]}$. Assignment is denoted $\leftarrow$, and sampling (as well as assignment from randomized algorithms) is denoted $\leftarrow\!\!\$$, with $x \leftarrow\!\!\$\, \mathcal{X}$ denoting uniform sampling from the set $\mathcal{X}$. Finally, the operation $\xleftarrow{\cup}$ means adding an element to a set, and the operation $\twoheadleftarrow$ appends an element to a list.

We use code-based experiments, where by convention all sets, lists, and lazily sampled functions are initialized empty. We use $\Pr[\mathsf{Code} : \mathsf{Event} \mid \mathsf{Condition}]$ to denote the conditional probability of Event occurring when Code is executed, conditioned on $\mathsf{Condition}$. In our pseudocode, the keyword **parse** implicitly returns an error symbol $\perp_{\text{parse}}$ whenever the input format fails to meet the requirement. In our game hops, bad events are inserted directly into the pseudocode; if they are ever reached, then the experiment aborts with the output bad.

2.2 Public-Key Encryption

A public-key encryption scheme PKE consists of five algorithms: $\mathsf{PKE.Pm}$, which takes only the security parameter (in unary) and outputs (shared) public parameters pm; the keypair sampling algorithm $\mathsf{PKE.Kg}$, which takes parameters as input and samples a keypair $(\mathsf{pk}, \mathsf{sk})$; the randomness sampling algorithm $\mathsf{PKE.Rnd}$, which takes parameters as input and samples a string of random coins r to be used in encryption; the encryption algorithm $\mathsf{PKE.Enc}$, encrypting messages m belonging to some message space $\mathcal{M}$ to ciphertexts c as $\mathsf{PKE.Enc}_{\mathsf{pk}}(m)$ (which implicitly calls $r \leftarrow\!\!\$\ \mathsf{PKE.Rnd}(\mathsf{pm})$ and runs $\mathsf{PKE.Enc}_{\mathsf{pk}}(m; r)$); finally, the decryption algorithm $\mathsf{PKE.Dec}$, which decrypts ciphertexts c under a private key sk to recover message m, or else outputs the failure symbol $\perp$.

We require that our schemes satisfy perfect correctness, i.e. for all messages m in the message space, for all parameters pm and for all keypairs $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{PKE.Kg}(\mathsf{pm})$,

$$\Pr[r \leftarrow\!\!\$\ \mathsf{PKE.Rnd}(\mathsf{pm}) \ :\ \mathsf{PKE.Dec}_{\mathsf{sk}}(\mathsf{PKE.Enc}_{\mathsf{pk}}(m; r)) = m] = 1\,.$$

2.3 Asymptotic and Concrete Security

The concept of a separation presupposes the concept of security. Hence, though we prefer a concrete-security treatment, we state our main results asymptotically. We bridge the gap by phrasing our theorems as asymptotic interpretations of concrete-security lemmata.

2.3.1 Asymptotics

We next recall the standard asymptotic security definitions. Adversaries, simulators, etc., are modelled as algorithms, and any algorithm is taken to be probabilistic polynomial-time

(PPT) Turing machines, unless otherwise specified. We want security to be defined relative to a security parameter $\lambda \in \mathbb{N}$, and so all algorithms are implicitly given the security parameter in unary as part of their input.

Definition 1 (Negligible Function)**.** A function $\varepsilon : \mathbb{N} \to \mathbb{R}$ is said to be *negligible*, denoted $\varepsilon(\lambda) \leqslant \mathsf{negl}(\lambda)$, if for every constant $c > 0$ there exists an integer λ_0 such that, for every $\lambda \geqslant \lambda_0$, $|\varepsilon(\lambda)| < \lambda^{-c}$.

In addition to negligible functions, we will also need the notion of an inverse-polynomial, also known as a "noticeable" function, differing from "non-negligible" in how the polynomials are quantified.[3]

Definition 2 (Noticeable Function)**.** A function $\varepsilon : \mathbb{N} \to \mathbb{R}$ is said to be *noticeable*, denoted $\varepsilon(\lambda) \geqslant 1/\mathsf{poly}(\lambda)$, if there exists a constant $c > 0$ and integer λ_0 such that, for every $\lambda \geqslant \lambda_0$, $|\varepsilon(\lambda)| \geqslant \lambda^{-c}$.

2.3.2 Advantage

Let $\mathsf{PKE}[\lambda]$ denote the PKE scheme PKE instantiated with security parameter λ. The concrete security of $\mathsf{PKE}[\lambda]$, relative to some goal GOAL and adversarial power POWER, is defined in terms of an experiment $\Pr\left[\mathsf{Exp}^{\text{goal-power}}_{\mathsf{PKE}[\lambda],\ldots}(\mathbb{A})\right]$ and the advantage of a partaking adversary $\mathbb{A}$. (We prefer to define advantages without taking the absolute value and accepting that advantages may be negative, noting that for every adversary there exists another adversary with equal but opposite advantage that always outputs the inverse bit value of the first.)

Definition 3 (Distinguishing Advantage)**.** Let $\mathsf{PKE}[\lambda]$ be a PKE scheme instantiated with security parameter λ, and let $\mathsf{Exp}^{\text{goal-power}}_{\mathsf{PKE}[\lambda],\ldots}(\mathbb{A})$ be a distinguishing experiment for the notion GOAL-POWER, where the dots represent other possible dependencies of the notion (e.g. simulators or message samplers). Then an adversary $\mathbb{A}$'s distinguishing advantage is defined as

$$\mathsf{Adv}^{\text{goal-power}}_{\mathsf{PKE}[\lambda],\ldots}(\mathbb{A}) \coloneqq 2 \cdot \Pr\left[\mathsf{Exp}^{\text{goal-power}}_{\mathsf{PKE}[\lambda],\ldots}(\mathbb{A})\right] - 1\,.$$

Asymptotic security (defined for each notion below) is then achieved in the $\lambda \to \infty$ limit whenever the advantage, when interpreted as a function of λ, is negligible for all adversaries.

2.4 Message Samplers

Notions of selective opening attacks are formulated relative to message samplers M and resamplers S. These are fixed by the experiment, though the distribution from which M samples each message may depend on an adversarially chosen input α and the internal state s. A call to a message sampler samples a single message, and the subscript $\langle s \rangle$ means that the algorithm is implicitly updating its state after each call, matching a formalization due to Bellare and Yilek [BY12].

The ISO experiment (to be presented shortly) is defined in terms of an encryption phase, in which messages are sampled and encrypted one-by-one, and a challenge phase, in which the entire list of messages is resampled, conditioned on the opened messages. We formalize this in terms of a resampling algorithm S, which resamples list of ℓ messages conditioned on a list of adversarial inputs $\mathtt{A} = [\alpha_\ell]$, message lengths $\mathtt{L} = [|m_\ell|]$, a set of opened indices $\mathcal{J} \subseteq [\ell]$, and the opened messages themselves $\mathtt{M}[\mathcal{J}]$, in such a way that the output is a list of ℓ messages $\mathtt{M}'$ satisfies $\mathtt{M}'[\mathcal{J}] = \mathtt{M}[\mathcal{J}]$.

[3] Negating the quantifiers of Definition 1 yields the following definition for "non-negligible": $\exists c > 0$ s.th. $\forall \lambda_0 \in \mathbb{N}$, $\exists \lambda \geqslant \lambda_0$ s.th. $|\varepsilon(\lambda)| \geqslant \lambda^{-c}$. There exist functions that are non-negligible but not noticeable.

With openings, the resampling procedure must ensure consistency with whatever the adversary trivially learns from the opened messages. Such conditional resampling may not be efficient, due to non-trivial relations between opened and hidden messages.[4] Following BHS, we will often argue in terms of an ideal, possibly inefficient resampler $\mathring{\mathsf{S}}$; then, there may be an efficient but imperfect resampler S whose output is negligibly close to that of $\mathring{\mathsf{S}}$, in which case security will still follow. We will call any message sampler M for which such a resampler S exists (efficiently, conditionally) "resamplable". As an example, any message sampler that samples from a product distribution (independent samples) is trivially resamplable.

Definition 4 (Resampling Error)**.** Let M be a stateful sampling algorithm with ideal resampler $\mathring{\mathsf{S}}$, and let S be a resampling algorithm. Let $\ell \in \mathbb{N}$ correspond to the number of M calls, and define the support $\mathrm{Supp}_\ell(\mathsf{M})$ as the set of all tuples $(\mathtt{A}, \mathtt{L}, \mathcal{J}, \mathtt{M}_{\mathsf{real}}[\mathcal{J}])$, subject to $|\mathtt{A}| = |\mathtt{L}| = \ell$, that may occur in an execution of the ISO experiment (Figure 5). For $(\mathtt{A}, \mathtt{L}, \mathcal{J}, \mathtt{M}_{\mathsf{real}}[\mathcal{J}]) \in \mathrm{Supp}_\ell(\mathsf{M})$, let $\delta_{\mathring{\mathsf{S}},\mathsf{S}}(\mathtt{A}, \mathtt{L}, \mathcal{J}, \mathtt{M}_{\mathsf{real}}[\mathcal{J}])$ be the statistical distance between $\mathring{\mathsf{S}}(\mathtt{A}, \mathtt{L}, \mathcal{J}, \mathtt{M}_{\mathsf{real}}[\mathcal{J}])$ and $\mathsf{S}(\mathtt{A}, \mathtt{L}, \mathcal{J}, \mathtt{M}_{\mathsf{real}}[\mathcal{J}])$. Then the *resampling error* of S is

$$\epsilon^{\ell}_{\mathring{\mathsf{S}},\mathsf{S}}(\lambda) = \max_{(\mathtt{A},\mathtt{L},\mathcal{J},\mathtt{M}_{\mathsf{real}}[\mathcal{J}]) \in \mathrm{Supp}_\ell(\mathsf{M})} \delta_{\mathring{\mathsf{S}},\mathsf{S}}(\mathtt{A}, \mathtt{L}, \mathcal{J}, \mathtt{M}_{\mathsf{real}}[\mathcal{J}])\,.$$

We will always take message samplers M and resamplers S to be PPT, though we only assume that ideal resamplers $\mathring{\mathsf{S}}$ halt. If there exists a PPT resampler S such that $\epsilon^{\ell}_{\mathring{\mathsf{S}},\mathsf{S}}(\lambda)$ is negligible, we say that M is *resamplable*.

2.4.1 Sampler Entropy

Our main results depend on the min-entropy of the message distributions being sampled from, and so we introduce the notions of high-entropy and low-entropy message samplers. In the following, let $\mathrm{Dom}(\mathsf{M})$ and $\mathrm{Supp}(\mathsf{M})$ denote the domain resp. support of message sampler M.

Definition 5 (High-entropy Sampler)**.** For a given message sampler M, let $\mathsf{max}_{\mathsf{M}} = \mathsf{max}_{\mathsf{M}}(\lambda)$ be its *maximum guessing probability*, defined as

$$\mathsf{max}_{\mathsf{M}} \coloneqq \max_{\substack{\alpha,\, s \in \mathrm{Dom}(\mathsf{M}),\\ m \in \mathrm{Supp}(\mathsf{M})}} \Pr\left[m' \leftarrow\!\$\ \mathsf{M}_{\langle s \rangle}(\alpha)\ :\ m' = m\right].$$

A message sampler M is called *high-entropy* whenever $\mathsf{max}_{\mathsf{M}}$ is *negligible*.

Intuitively, one can view high-entropy message samplers as those for which *every* sample, regardless of input α and state s, depends on superpolynomially many random coins. These are the samplers employed in Theorem 1.

In order to show Theorem 2, we will need a much stronger notion of low-entropy sampler than what we would get by simply negating Definition 5. Instead, we take a low-entropy sampler M to be one for which any message *list*, as produced by a series of calls to M, can be resampled in polynomial time. Since the size of the space of message vectors grows exponentially with ℓ, this implies that the entropy across all samples can be at most logarithmic in λ. Hence, a low-entropy sampler is one whose input/output behaviour, across any number of calls, depends on at most $\mathcal{O}(\log(\lambda))$ bits of internal randomness.

[4] As a simple example, consider the sampler that samples a seed $s = m_0$ at the outset, and then for each call performs the repeated hashing procedure $m_i = h(m_{i-1})$. Without openings, resampling can simply start over with a new seed; with openings, resampling may require inverting the hash function or finding hash collisions.

$\underline{\text{Sampler } \mathsf{M}^{\text{ind}}_{\langle s \rangle}(\alpha)}$
if $s = \varepsilon : s \leftarrow\!\!\$\ \{0,1\}$
parse $\alpha = (m_0, m_1)$
return m_s

Figure 3: The left-or-right sampler, imitating the standard indistinguishability game.

$\underline{\text{Experiment } \mathsf{Exp}^{\text{n-ind-cca-r}}_{\mathsf{PKE}[\lambda]}(\mathbb{A})}$
$b \leftarrow\!\!\$\ \{0,1\}$
$\mathsf{pm} \leftarrow\!\!\$\ \mathsf{PKE.Pm}(1^\lambda)$
$[\mathsf{pk}_n, \mathsf{sk}_n] \leftarrow\!\!\$\ \mathsf{PKE.Kg}(\mathsf{pm})$
$\hat{b} \leftarrow\!\!\$\ \mathbb{A}^{\mathcal{E},\mathcal{D},\mathcal{R}}([\mathsf{pk}_n])$
if $\exists i \in \mathcal{I}$ s.t. $\mathcal{C}_i \neq \emptyset : \hat{b} \leftarrow\!\!\$\ \{0,1\}$
return $(b = \hat{b})$

$\underline{\text{Oracle } \mathcal{E}(i, m_0, m_1)}$
$c^* \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}_i}(m_b)$
$\mathcal{C}_i \xleftarrow{\cup} c^*$
return c^*

$\underline{\text{Oracle } \mathcal{D}(i, c)}$
if $c \in \mathcal{C}_i$: **return** $\lightning$
return $\mathsf{PKE.Dec}_{\mathsf{sk}_i}(c)$

$\underline{\text{Oracle } \mathcal{R}(i)}$
$\mathcal{I} \xleftarrow{\cup} i$
return sk_i

Figure 4: Indistinguishability against chosen-ciphertext attacks, with n users and with receiver opening (adaptive corruption).

Definition 6 (Low-entropy Sampler)**.** Let M be a stateful sampling algorithm with ideal resampler $\mathring{\mathsf{S}}$. Let $\ell = \mathsf{poly}(\lambda)$ correspond to the number of M calls, and let $\mathrm{Supp}_\ell(\mathsf{M})$ be defined as in Definition 4. Then, $\mathsf{min}_{\mathsf{M}}(\ell)$ is the *minimum resampling probability*, defined as

$$\mathsf{min}_{\mathsf{M}}(\ell) := \min_{\substack{(\mathtt{A},\mathtt{L},\mathcal{J},\mathtt{M}_{\mathsf{real}}[\mathcal{J}]) \in \mathrm{Supp}_\ell(\mathsf{M}) \\ \mathtt{M} \in \mathrm{Supp}(\mathring{\mathsf{S}})}} \Pr\left[\mathtt{M}' \leftarrow\!\!\$\ \mathring{\mathsf{S}}(\mathtt{A}, \mathtt{L}, \mathcal{J}, \mathtt{M}_{\mathsf{real}}[\mathcal{J}]) \ : \ \mathtt{M}' = \mathtt{M}\right].$$

A message sampler M is called *low-entropy* whenever $\mathsf{min}_{\mathsf{M}}(\ell)$ is *noticeable*.

Note that any low-entropy sampler is also *resamplable*: Since the space of possible message vectors is of polynomial size, the ideal resampler can enumerate its conditional support in polynomial time, then sample directly from it.

The above definitions capture several natural samplers: For example, uniform samplers over exponential message spaces clearly satisfy the definition of a high-entropy sampler. For a relevant example of a low-entropy sampler, consider the left-or-right sampler in Figure 3, employed in reductions from notions of SOA to standard indistinguishability, and whose output depends on only a single bit of internal randomness.

Remark 1. There is a significant gap between high-entropy and low-entropy message samplers as defined above, with many natural samplers falling into an intermediate "mid-entropy" category. This includes samplers that sample uniformly from any polynomial, and even constant, sized message space: Indeed, even uniform sampling of binary messages $m \in \{0,1\}$ will land in this "mid-entropy" category.

2.5 Security Notions

2.5.1 Goals and Powers

Security notions are defined with respect to a GOAL and a set of adversarial POWERs. We are concerned with notions of indistinguishability and selective opening attacks, and so we consider the goals IND, ISO (Indistinguishability under Selective Opening attacks), and SSO (Simulatability under Selective Opening attacks). If the notion is a multi-user notion, then GOAL is prefixed with the number of users $n \in \mathbb{N}$.

Experiment $\mathsf{Exp}^{\text{n-iso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A})$

$b \leftarrow\!\!\$\ \{\mathsf{real}, \mathsf{ideal}\}$
$\mathsf{challenged} \leftarrow \mathsf{false}$
$q \leftarrow 0$
$\mathsf{pm} \leftarrow\!\!\$\ \mathsf{PKE.Pm}(1^\lambda)$
$[\mathsf{pk}_n, \mathsf{sk}_n] \leftarrow\!\!\$\ \mathsf{PKE.Kg}(\mathsf{pm})$
$\hat{b} \leftarrow\!\!\$\ \mathbb{A}^{\mathcal{E},\mathcal{C},\mathcal{D},\mathcal{S},\mathcal{R}(,\mathcal{T})}([\mathsf{pk}_n])$
return $(b = \hat{b})$

Oracle $\mathcal{C}$

require $\neg\mathsf{challenged}$
$\mathsf{challenged} \leftarrow \mathsf{true}$
for $j \in [q]$:
  if $\mathtt{K}[j] \in \mathcal{I} : \mathcal{J} \xleftarrow{\cup} j$
$\mathtt{M}_{\mathsf{ideal}} \leftarrow\!\!\$\ \mathsf{S}(\mathtt{A}, \mathtt{L}, \mathcal{J}, \mathtt{M}_{\mathsf{real}}[\mathcal{J}])$
return $\mathtt{M}_b$

Oracle $\mathcal{E}(i, \alpha)$

require $\neg\mathsf{challenged}$
$q \leftarrow q + 1$
$\mathtt{K} \twoheadleftarrow i, \mathtt{A} \twoheadleftarrow \alpha$
$m \leftarrow\!\!\$\ \mathsf{M}_{\langle s \rangle}(\alpha)$
$\mathtt{L} \twoheadleftarrow |m|, \mathtt{M}_{\mathsf{real}} \twoheadleftarrow m$
$r \leftarrow\!\!\$\ \mathsf{PKE.Rnd}(\mathsf{pm})$
$c^* \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}_i}(m; r)$
$\mathtt{R} \twoheadleftarrow r, \mathcal{C}_i \xleftarrow{\cup} c^*$
return c^*

Oracle $\mathcal{D}(i, c)$

if $c \in \mathcal{C}_i$: **return** $\lightning$
return $\mathsf{PKE.Dec}_{\mathsf{sk}_i}(c)$

Oracle $\mathcal{S}(j)$

require $\neg\mathsf{challenged}$
$\mathcal{J} \xleftarrow{\cup} j$
return $(\mathtt{M}_{\mathsf{real}}[j], \mathtt{R}[j])$

Oracle $\mathcal{R}(i)$

require $\neg\mathsf{challenged}$
$\mathcal{I} \xleftarrow{\cup} i$
return sk_i

Oracle $\mathcal{T}(j)$

require $\neg\mathsf{challenged}$
$\mathcal{J} \xleftarrow{\cup} j$
return $\mathtt{M}_{\mathsf{real}}[j]$

Figure 5: Indistinguishability SOA with bi-opening. (The transmission opening oracle $\mathcal{T}$ is given to the adversary in paranthesis, since access to the sender opening oracle $\mathcal{S}$ makes access to the former redundant.)

The powers we consider are the standard CPA and (adaptive) CCA, as well as various kinds of adaptive opening. The openings considered are: T, for Transmission (or "Trivial") opening, in which only messages leak; S, for Sender opening, in which both messages and encryption randomness leak; R, for Receiver opening, in which private keys leak; B, for Bi-opening [LYHW21] (or "Both"), for which both sender and receiver opening is available; finally, N designates No opening. To summarize:

$$\text{GOAL} \in \{\text{IND}, \text{ISO}, \text{SSO}\}, \quad \text{POWER} \in \{\text{CPA-OP}, \text{CCA-OP}\},$$
$$\text{OP} \in \{\text{N}, \text{T}, \text{S}, \text{R}, \text{B}\}.$$

2.5.2 Standard Indistinguishability

We define indistinguishability against chosen ciphertext attacks (IND-CCA) next. We give a multi-user definition with receiver opening (adaptive corruption). Indistinguishability against chosen plaintext attacks (IND-CPA) follows by ignoring the decryption oracle, and single-user notions follow by setting $n = 1$ and ignoring the opening oracle.

Definition 7 (IND Security)**.** Let $\mathsf{Exp}^{\text{n-ind-cca-r}}_{\mathsf{PKE}[\lambda]}(\mathbb{A})$ be the experiment given in Figure 4, and let $\mathsf{Adv}^{\text{n-ind-cca-r}}_{\mathsf{PKE}[\lambda]}(\mathbb{A})$ be the corresponding distinguishing advantage. A PKE scheme PKE is said to be IND-CCA secure if, for every $n \leqslant \mathsf{poly}(\lambda)$ and for every PPT adversary $\mathbb{A}$, the distinguishing advantage is negligible in λ.

2.5.3 Indistinguishability under Selective Opening Attacks

Indistinguishability SOA separates the experiment into an encryption phase, in which the adversary receives challenge ciphertexts encrypting messages sampled from M, and a challenge phase, in which the adversary either receives a list of the sampled messages, or an alternative message list sampled using the resampler S.

Experiment $\mathsf{Exp}^{\text{n-sso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}}(\mathbb{A},\mathbb{D})$

$b \leftarrow\!\$\ \{\mathsf{real},\mathsf{ideal}\}$
$\mathsf{pm} \leftarrow\!\$\ \mathsf{PKE.Pm}(1^\lambda)$
$[\mathsf{pk}_n,\mathsf{sk}_n] \leftarrow\!\$\ \mathsf{PKE.Kg}(\mathsf{pm})$
if $b = \mathsf{real}$:
  $\mathsf{out} \leftarrow\!\$\ \mathbb{A}^{\mathcal{E},\mathcal{D},\mathcal{S},\mathcal{R}(,\mathcal{T})}([\mathsf{pk}_n])$
  for $j \in [|\mathtt{K}|]$:
    if $\mathtt{K}[j] \in \mathcal{I}$: $\mathcal{J} \xleftarrow{\cup} j$
if $b = \mathsf{ideal}$:
  $\mathsf{out} \leftarrow\!\$\ \mathsf{Sim}^{\mathcal{E},\mathcal{T}}(\mathsf{pm})$
$\hat{b} \leftarrow\!\$\ \mathbb{D}(\mathsf{pm},\mathtt{A},\mathtt{M}_{\mathsf{real}},\mathcal{J},\mathsf{out})$
return $(b = \hat{b})$

Oracle $\mathcal{E}(i,\alpha)$

$\mathtt{K} \twoheadleftarrow i, \mathtt{A} \twoheadleftarrow \alpha$
$m \leftarrow\!\$\ \mathsf{M}_{\langle s\rangle}(\alpha)$
$\mathtt{M}_{\mathsf{real}} \twoheadleftarrow m$
$r \leftarrow\!\$\ \mathsf{PKE.Rnd}(\mathsf{pm})$
$c^* \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}_i}(m;r)$
$\mathtt{R} \twoheadleftarrow r, \mathcal{C}_i \xleftarrow{\cup} c^*$
if $b = \mathsf{ideal}$: **return** $|m|$
else : **return** $(|m|, c^*)$

Oracle $\mathcal{D}(i,c)$

if $c \in \mathcal{C}_i$: **return** $\lightning$
return $\mathsf{PKE.Dec}_{\mathsf{sk}_i}(c)$

Oracle $\mathcal{S}(j)$

$\mathcal{J} \xleftarrow{\cup} j$
return $(\mathtt{M}_{\mathsf{real}}[j], \mathtt{R}[j])$

Oracle $\mathcal{R}(i)$

$\mathcal{I} \xleftarrow{\cup} i$
return sk_i

Oracle $\mathcal{T}(j)$

$\mathcal{J} \xleftarrow{\cup} j$
return $\mathtt{M}_{\mathsf{real}}[j]$

Figure 6: Simulatability SOA, in which a distinguisher $\mathbb{D}$ is tasked with guessing whether it received the view of adversary $\mathbb{A}$ playing the real game or a simulated view (by Sim).

Definition 8 (ISO Security)**.** Let $\mathsf{Exp}^{\text{n-iso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A})$ be the experiment given in Figure 5, and let $\mathsf{Adv}^{\text{n-iso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A})$ be the corresponding distinguishing advantage. A PKE scheme PKE is said to be ISO-CCA-B secure if for every $n \leqslant \mathsf{poly}(\lambda)$, for every resamplable message sampler M with resampler S, and for every PPT adversary $\mathbb{A}$, the distinguishing advantage is negligible in λ.

As before, the definition of CPA security follows from ignoring the CCA oracle, and security relative to other kinds of opening similarly follow by ignoring irrelevant opening oracles in Figure 5. Whenever security holds to relative to some restricted class of message samplers, we say that the scheme achieves *restricted* ISO security.

It is often useful to argue about ISO security in terms of an ideal resampler, cf. Definition 4.

Lemma 1 (BHS, Lemma 3)**.** *Let* $\mathsf{PKE}[\lambda]$ *be a PKE scheme instantiated with* λ*, and let* $\mathbb{A}$ *be an* n-ISO-CCA-B *adversary that makes at most* ℓ $\mathcal{E}$*-queries. Let* M *be a message sampler with ideal resampler* $\mathring{\mathsf{S}}$*, and let* S *be an arbitrary resampler. Then*

$$\left|\mathsf{Adv}^{\text{n-iso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A}) - \mathsf{Adv}^{\text{n-iso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathring{\mathsf{S}}}(\mathbb{A})\right| \leqslant \epsilon^{\ell}_{\mathring{\mathsf{S}},\mathsf{S}}(\lambda)\,.$$

2.5.4 Simulatability under Selective Opening Attacks

The simulatability SOA experiment proceeds identically to the encryption phase of the indistinguishability SOA experiment, except that instead of guessing a challenge bit, the adversary halts with an arbitrary output out. In the real-world experiment, this output is then forwarded to a distinguisher $\mathbb{D}$, and in the ideal-world experiment a simulated output is provided by a simulator Sim, who only sees the message lengths, and who only gets access to transmission openings. The distinguisher then tries to guess which kind of output it received.

SSO security comes in a weak version and a strong version, depending on the quantification of Sim; we default to the strong version unless otherwise noted.

Definition 9 (SSO Security)**.** Let $\mathsf{Exp}^{\text{n-sso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}}(\mathbb{A},\mathbb{D})$ be the experiment given in Figure 6, and let $\mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}}(\mathbb{A},\mathbb{D})$ be the corresponding distinguishing advantage. A PKE scheme PKE is said to be SSO-CCA-B secure if for every $n \leqslant \mathsf{poly}(\lambda)$, for every

Oracle $\mathcal{P}(i, m, c)$	$\mathsf{PKE}'.\mathsf{Enc}_{\mathsf{pk}}(m)$	$\mathsf{PKE}'.\mathsf{Dec}_{\mathsf{sk}}(c)$
$m' \leftarrow \mathsf{PKE}.\mathsf{Dec}_{\mathsf{sk}_i}(c)$	$c \leftarrow\!\!\$\ \mathsf{PKE}.\mathsf{Enc}_{\mathsf{pk}}(m)$	**if** $c[1] = \mathsf{reg}$:
return $(m = m')$	**return** (reg, c)	**return** $\mathsf{PKE}.\mathsf{Dec}_{\mathsf{sk}}(c[2])$
		else if $c[1] = \mathsf{check}$:
		return $(c[3] = \mathsf{PKE}.\mathsf{Dec}_{\mathsf{sk}}(c[2]))$
		else : **return** $\perp$

Figure 7: Left: What we want is a plaintext checking oracle, which would immediately allow an ISO distinguishing attack. Middle/right: The separating scheme PKE', which instantiates a plaintext checking oracle inside the decryption oracle.

message sampler M and PPT adversary $\mathbb{A}$, there exists a PPT simulator Sim such that for every PPT distinguisher $\mathbb{D}$, the distinguishing advantage is negligible in λ.

Quantifying the simulator after the distinguisher in the above definition (allowing the simulator to depend on the distinguisher) yields the weak version of SSO security. CPA security and security relative to other kinds of opening follow by ignoring oracles as before, and whenever security holds to relative to some restricted class of message samplers, we say that the scheme achieves *restricted* SSO security.

Our formalization differs from that of BHS by having the real-world encryption oracle also provide the adversary with message lengths. Otherwise, length-hiding encryption schemes would generally not achieve the notion (since the simulator would know the message lengths while the adversary would not), which seems artificial.

The SSO experiment does not involve any resampling, and so SSO security is well-defined for all (rather than only resamplable, cf. Definition 8) message samplers.

3 High-Entropy Separation

The basic idea underlying our high-entropy separation is to alter the CCA oracle in such a way that it can be used to distinguish, allowing for an ISO attack without breaking simulatability. In essence, the reason that this is at all possible is the same as the reason why a direct reduction runs into problems: Because in the ISO game, the adversary has access to a decryption oracle in the distinguishing (challenge) phase of the game, while in the SSO game, the distinguisher (who is now a separate entity) does not.

In fact, access to a plaintext checking attack (PCA) oracle, shown in Figure 7 (left) would already give us what we need: Now, distinguishing is trivial, and yet the oracle cannot help a SSO adversary unless they are also able to guess the possible challenge messages. We turn this observation into a separating scheme by effectively instantiating a PCA oracle in the decryption algorithm: Built from an underlying scheme PKE assumed to achieve SSO-CCA security, our separating scheme PKE' marks standard encryptions as "regular" when encrypting (Figure 7, middle). The decryption algorithm (Figure 7, right), meanwhile, now has two "modes", depending on the keyword preceding the underlying ciphertext: For reg, it decrypts the second element of the input as usual, while for check, it decrypts the second element and compares it to the third element. As desired, this construction instantiates a PCA oracle inside a decryption oracle; crucially, calls to the check functionality will not be rejected when called on challenge ciphertexts, since the oracle will only reject challenge ciphertexts marked reg.

Theorem 1 (SSO-CCA-B does not imply ISO-CCA-N)**.** *There exists a PKE scheme that achieves restricted* SSO-CCA-B *security relative to any high-entropy sampler (Definition 5), yet is* ISO-CCA-N *insecure relative to all samplers.*

Proof. Interpreting the bound of Lemma 2, we see that if $\mathsf{max}_{\mathsf{M}} \leqslant \mathsf{negl}(\lambda)$ and $\mathring{\mathsf{S}}$ is efficient (or there is an efficient resampler S whose output is negligibly close to that of $\mathring{\mathsf{S}}$, applying Lemma 1), then security follows; indeed, as $\mathsf{max}_{\mathsf{M}} \to 0$, we even get tightness. Vice versa, security does not follow whenever $\mathsf{max}_{\mathsf{M}}$ is noticeable, and as $\mathsf{max}_{\mathsf{M}} \to 1/q_d$, the bound becomes void.

If an opening oracle was called at least once, conditional resampling may be inefficient altogether, leading to an inefficient simulator a non-negligible advantage term on the right-hand-side, and security once again does not follow. However, ISO-CCA security is only well-defined relative to resamplable message samplers, and so SSO-CCA security trivially does not imply ISO-CCA relative to non-resamplable samplers.

The advantage of the attacker presented in Lemma 3 is clearly non-negligible for all relevant values of $\mathsf{max}_{\mathsf{M}}$: Setting $\ell = \mathsf{poly}(\lambda)$, the adversary achieves noticeable advantage for any message sampler M such that $\mathsf{max}_{\mathsf{M}}$ is not negligibly close to 1, with the advantage becoming overwhelming for negligible $\mathsf{max}_{\mathsf{M}}$. Further, since the attack is performed in a model without openings, resampling is guaranteed to be efficient. □

3.1 PKE' is SSO-CCA Secure Relative to High-entropy Samplers

We first show that, if PKE is SSO-CCA secure, then so is PKE', up to the guessing probability of the message sampler. While stated for bi-opening, the statement holds for any opening.

Lemma 2. *Let* PKE *be a PKE scheme, and let* PKE' *be the construction given in Figure 7. Then,*

1. *for every adversary* $\mathbb{A}$*, there is a reduction* $\mathbb{B}$ *such that*
2. *for every simulator* $\mathsf{Sim}_{\mathbb{B}}$*, there is a simulator* $\mathsf{Sim}_{\mathbb{A}}$ *such that*
3. *for every distinguisher* $\mathbb{D}_{\mathbb{A}}$*, there is a distinguisher* $\mathbb{D}_{\mathbb{B}}$*, such that*

for all message samplers M *with maximum guessing probability at most* $\mathsf{max}_{\mathsf{M}}$*, if* $\mathbb{A}$ *makes at most* q_d *calls to the decryption oracle, then*

$$\mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}'[\lambda],\mathsf{M},\mathsf{Sim}_{\mathbb{A}}}(\mathbb{A},\mathbb{D}_{\mathbb{A}}) \leqslant 2 \cdot \mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}_{\mathbb{B}}}(\mathbb{B},\mathbb{D}_{\mathbb{B}}) + q_d \cdot \mathsf{max}_{\mathsf{M}}\,.$$

Reduction $\mathbb{B}$ *and distinguisher* $\mathbb{D}_{\mathbb{B}}$ *run in time roughly equal to* $\mathbb{A}$ *and* $\mathbb{D}_{\mathbb{A}}$ *respectively, and* $\mathsf{Sim}_{\mathbb{B}}$ *runs in time roughly equal* $\mathsf{Sim}_{\mathbb{A}}$ *plus up to* q_d *calls to* $\mathring{\mathsf{S}}$*.*

Proof. The proof inserts one intermediary game hop, introducing a bad event if the adversary $\mathbb{A}$ ever calls the decryption oracle on a challenge ciphertext together with the correct decryption. The second hop switches player from the adversary to the simulator. Since the simulator does not get access to the decryption oracle, G_2 is then identical to the ideal SSO-CCA game. Thus, the second hop lends itself to a straightforward SSO-CCA reduction; the brunt of the proof lies in the analysis of the first hop.

G_0 is identical to the real SSO-CCA game, except inserting the construction PKE' directly into the game, and as mentioned, G_2 is identical to the ideal SSO-CCA game. We therefore have (using an equivalent advantage definition):

$$\begin{aligned}\mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}'[\lambda],\mathsf{M},\mathsf{Sim}_{\mathbb{A}}}(\mathbb{A},\mathbb{D}_{\mathbb{A}}) &:= \Pr[\mathrm{G}_0 : \mathsf{real} \leftarrow\!\$\ \mathbb{D}_{\mathbb{A}}] - \Pr[\mathrm{G}_2 : \mathsf{real} \leftarrow\!\$\ \mathbb{D}_{\mathbb{A}}] \\ &= \Pr[\mathrm{G}_0 : \mathsf{real} \leftarrow\!\$\ \mathbb{D}_{\mathbb{A}}] - \Pr[\mathrm{G}_1 : \mathsf{real} \leftarrow\!\$\ \mathbb{D}_{\mathbb{A}}] \\ &\quad + \Pr[\mathrm{G}_1 : \mathsf{real} \leftarrow\!\$\ \mathbb{D}_{\mathbb{A}}] - \Pr[\mathrm{G}_2 : \mathsf{real} \leftarrow\!\$\ \mathbb{D}_{\mathbb{A}}]\,.\end{aligned}$$

Game G_1 only adds the aforementioned bad event, and so G_0 and G_1 are identical-until-bad:

$$\begin{aligned}\mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}'[\lambda],\mathsf{M},\mathsf{Sim}_{\mathbb{A}}}(\mathbb{A},\mathbb{D}_{\mathbb{A}}) &\leqslant \Pr[\mathrm{G}_1 : \mathsf{real} \leftarrow\!\$\ \mathbb{D}_{\mathbb{A}}] - \Pr[\mathrm{G}_2 : \mathsf{real} \leftarrow\!\$\ \mathbb{D}_{\mathbb{A}}] \\ &\quad + \Pr[\mathsf{bad}]\,.\end{aligned}$$

Games G_0 – G_2

$\mathsf{pm} \leftarrow\!\$\ \mathsf{PKE.Pm}(1^\lambda)$
$[\mathsf{pk}_n, \mathsf{sk}_n] \leftarrow\!\$\ \mathsf{PKE.Kg}(\mathsf{pm})$
// G_0, G_1 $(b = 0)$:
  $\mathsf{out} \leftarrow\!\$\ \mathbb{A}^{\mathcal{E},\mathcal{D},\mathcal{S},\mathcal{R}}([\mathsf{pk}_n])$
  for $j \in [|\mathtt{K}|]$:
    if $\mathtt{K}[j] \in \mathcal{I} : \mathcal{J} \xleftarrow{\cup} j$
// G_2 $(b = 1)$:
  $\mathsf{out} \leftarrow\!\$\ \mathsf{Sim}_{\mathbb{A}}^{\mathcal{E},\mathcal{T}}(\mathsf{pm})$
$\hat{b} \leftarrow\!\$\ \mathbb{D}_{\mathbb{A}}(\mathsf{pm}, \mathtt{A}, \mathtt{M}_{\mathsf{real}}, \mathcal{J}, \mathsf{out})$
return $(b = \hat{b})$

Oracle $\mathcal{E}(i, \alpha)$

$\mathtt{K} \twoheadleftarrow i, \mathtt{A} \twoheadleftarrow \alpha$
$m \leftarrow\!\$\ \mathsf{M}_{\langle s \rangle}(\alpha)$
$\mathtt{M}_{\mathsf{real}} \twoheadleftarrow m$
$r \leftarrow\!\$\ \mathsf{PKE.Rnd}(\mathsf{pm})$
$c^* \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}_i}(m; r)$
$\mathtt{R} \twoheadleftarrow r$
$\mathcal{C}_i \xleftarrow{\cup} (\mathsf{reg}, c^*)$
return $|m|$ // G_2
return $(|m|, (\mathsf{reg}, c^*))$

Oracle $\mathcal{D}(i, c)$ // G_0, G_1 only

if $c \in \mathcal{C}_i$: **return** $\not{\iota}$
if $c[1] = \mathsf{reg}$:
  return $\mathsf{PKE.Dec}_{\mathsf{sk}_i}(c[2])$
else if $c[1] = \mathsf{check}$:
  parse $c = (\mathsf{check}, c', m)$
  $d \leftarrow (m = \mathsf{PKE.Dec}_{\mathsf{sk}_i}(c'))$
  if $d = \mathsf{true} \wedge c' \in \mathcal{C}_i$:
    bad // G_1
  return d
else : **return** $\perp$

Figure 8: The game hops G_0 – G_2. Not shown are the sender and receiver opening oracles, which are exactly as in Figure 6.

For the second hop we will assume $\neg\mathsf{bad}$; then, the game G_1 can be perfectly simulated for $\mathbb{A}$ by a reduction $\mathbb{B}_2$ who always returns false to decryption oracle calls of the form $\mathcal{D}(i, (\mathsf{check}, c \in \mathcal{C}_i, m))$, and using its own decryption oracle to correctly answer requests of the form $\mathcal{D}(i, (\mathsf{check}, c \notin \mathcal{C}_i, m))$; when $\mathbb{A}$ halts with output $\mathsf{out}_{\mathbb{A}}$, $\mathbb{B}_2$ halts with the same output. The ideal game, meanwhile, remains unchanged, so by letting $\mathbb{D}_2 = \mathbb{D}_{\mathbb{A}}$ and $\mathsf{Sim}_2 = \mathsf{Sim}_{\mathbb{A}}$, we get a standard SSO-CCA advantage term.

$$\mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}'[\lambda],\mathsf{M},\mathsf{Sim}_{\mathbb{A}}}(\mathbb{A}, \mathbb{D}_{\mathbb{A}}) \leqslant \mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}_2}(\mathbb{B}_2, \mathbb{D}_2) + \Pr[\mathsf{bad}]\,.$$

We now turn to analyzing the probability of the bad event. This occurs when $\mathbb{A}$ forwards a challenge ciphertext to the decryption oracle, using the check functionality, together with a correct decryption of the message. If the message had high entropy, then bad implies that $\mathbb{A}$ effectively broke the onewayness of PKE. This can then be leveraged by a SSO-CCA reduction $\mathbb{B}_1$, who, in collusion with a distinguisher $\mathbb{D}_1$, wins the SSO-CCA game, *provided* the output of M was not too easy for a simulator Sim_1 to guess to begin with.

In detail, $\mathbb{B}_1$ simulates G_1 for $\mathbb{A}$ as before, by forwarding its oracles and using its decryption oracle to simulate the checking functionality, except answering any calls of the form $\mathcal{D}(i, (\mathsf{check}, c \in \mathcal{C}_i, m))$ with false. Additionally, for any such call, $\mathbb{B}_1$ now saves the message m into its output $\mathsf{out}_{\mathbb{B}}$, ignoring $\mathbb{A}$'s output and forwarding this list instead to $\mathbb{D}_1$ at the end. The distinguisher $\mathbb{D}_{\mathbb{B}}$ (who gets access to the real $\mathtt{M}$) then checks whether at least one challenge message is contained in the received output out. Pseudocode for the reduction and distinguisher is given in Figure 9.

Let $\mathsf{max}_{\mathsf{M}}$ be the maximum guessing probability of M (see Definition 5), and assume w.l.o.g. that $\mathbb{A}$ opened all but one message, such that Sim_2 wins if it correctly guesses a single unopened challenge.[5] Let guess be the event that the simulator's output contains an unopened challenge message. Matching $\mathbb{B}_1$, the simulator Sim_1 gets at most q_d guesses; the union bound then gives us that

$$\Pr[\mathsf{guess}] \leqslant q_d \cdot \mathsf{max}_{\mathsf{M}}\,.$$

[5] We could do slightly better by bounding the number of openings, and using the probability that the whole message *vector* is resampled instead. Further, the probability that an efficient simulator correctly resamples even a single message post-opening may be significantly smaller than $\mathsf{max}_{\mathsf{M}}$, as conditional resampling may be inefficient.

Red. $\mathbb{B}_1^{\mathcal{E},\mathcal{D},\mathcal{S},\mathcal{R}}([\mathsf{pk}_n])$

$\mathsf{out}_{\mathbb{B}} \leftarrow \emptyset$
$\mathsf{out}_{\mathbb{A}} \leftarrow\!\$\ \mathbb{A}^{\mathcal{E},\mathcal{D},\mathcal{S},\mathcal{R}}([\mathsf{pk}_n])$
return $\mathsf{out}_{\mathbb{B}}$

Dist. $\mathbb{D}_1(\mathsf{pm}, \mathtt{A}, \mathtt{M}_{\mathsf{real}}, \mathcal{J}, \mathsf{out})$

if $\exists_{i \notin \mathcal{J}}\ \mathtt{M}_{\mathsf{real}}[i] \in \mathsf{out}$:
  return real
else : **return** ideal

If $\mathbb{A}$ calls $\mathcal{E}(i, \alpha)$

$(|m|, c^*) \leftarrow \mathcal{E}_{\mathbb{B}}(i, \alpha)$
$\mathcal{C}_i \xleftarrow{\cup} c^*$
return $(|m|, c^*)$

If $\mathbb{A}$ calls $\mathcal{D}(i, c)$

if $c[1] = \mathsf{reg}$:
  return $\mathcal{D}_{\mathbb{B}}(i, c[2])$
else if $c[1] = \mathsf{check}$:
  parse $c = (\mathsf{check}, c', m)$
  if $c' \notin \mathcal{C}_i$:
    return $(m = \mathcal{D}_{\mathbb{B}}(i, c'))$
  else :
    $\mathsf{out}_{\mathbb{B}} \xleftarrow{\cup} m$
    return false
else : **return** $\perp$

Figure 9: The reduction $\mathbb{B}_1$ and distinguisher $\mathbb{D}_1$, who leverages a bad event to win the SSO-CCA game against any simulator, provided the challenges are hard to guess. Not shown are the sender and receiver opening oracles, which are simply forwarded.

Let $\mathrm{G}_{\mathsf{real}}$ be the real SSO-CCA-B game, and let $\mathrm{G}_{\mathsf{ideal}}$ be the ideal game. Then we have

$$\begin{aligned}\mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}_1}(\mathbb{B}_1, \mathbb{D}_1) \coloneqq \Pr[\mathrm{G}_{\mathsf{real}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1] \\ - \Pr[\mathrm{G}_{\mathsf{ideal}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1]\,.\end{aligned}$$

For the real case, we have

$$\begin{aligned}\Pr[\mathrm{G}_{\mathsf{real}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1] = \Pr[\mathsf{bad}] \cdot \Pr[\mathrm{G}_{\mathsf{real}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1 \mid \mathsf{bad}] \\ + \Pr[\neg\mathsf{bad}] \cdot \Pr[\mathrm{G}_{\mathsf{real}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1 \mid \neg\mathsf{bad}]\,.\end{aligned}$$

The combined strategy of $\mathbb{B}_1$ and $\mathbb{D}_1$ (Figure 9) gives us that

$$\Pr[\mathrm{G}_{\mathsf{real}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1 \mid \mathsf{bad}] = 1\,,\ \Pr[\mathrm{G}_{\mathsf{real}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1 \mid \neg\mathsf{bad}] = 0\,,$$

giving us

$$\Pr[\mathrm{G}_{\mathsf{real}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1] = \Pr[\mathsf{bad}]\,.$$

For the ideal case, we have

$$\begin{aligned}\Pr[\mathrm{G}_{\mathsf{ideal}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1] = \Pr[\mathsf{guess}] \cdot \Pr[\mathrm{G}_{\mathsf{ideal}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1 \mid \mathsf{guess}] \\ + \Pr[\neg\mathsf{guess}] \cdot \Pr[\mathrm{G}_{\mathsf{ideal}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1 \mid \neg\mathsf{guess}]\,.\end{aligned}$$

In order to fool $\mathbb{D}_1$, any simulator Sim_1 must have correctly guessed at least one challenge message, meaning that $\mathbb{D}_1$ is fooled iff. event guess:

$$\Pr[\mathrm{G}_{\mathsf{ideal}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1 \mid \mathsf{guess}] = 1\,,\ \Pr[\mathrm{G}_{\mathsf{ideal}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1 \mid \neg\mathsf{guess}] = 0\,,$$

implying

$$\Pr[\mathrm{G}_{\mathsf{ideal}}(\mathbb{D}_1) \,:\, \mathsf{real} \leftarrow\!\$\ \mathbb{D}_1] = \Pr[\mathsf{guess}]\,.$$

Collecting, we have

$$\begin{aligned}\mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}_1}(\mathbb{B}_1, \mathbb{D}_1) &= \Pr[\mathsf{bad}] - \Pr[\mathsf{guess}] \\ \implies \Pr[\mathsf{bad}] &= \mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}_1}(\mathbb{B}_1, \mathbb{D}_1) + \Pr[\mathsf{guess}] \\ &\leqslant \mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}_1}(\mathbb{B}_1, \mathbb{D}_1) + q_d \cdot \mathsf{max}_{\mathsf{M}}\,,\end{aligned}$$

Adversary $\mathbb{A}_\ell^{\mathcal{E},\mathcal{C},\mathcal{D}}(\mathsf{pk})$

foreach $i \in [\ell] : c_i \leftarrow \mathcal{E}$
$\mathtt{M}_b \leftarrow \mathcal{C}$
foreach $i \in [\ell]$:
  $d \leftarrow \mathcal{D}(\mathsf{check}, c_i, \mathtt{M}_b[i])$
  if $d = \mathsf{false}$: **return** ideal
return real

Figure 10: The ISO-CCA attacker against PKE', shown here for the case of samplers that take no input.

giving us

$$\mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}'[\lambda],\mathsf{M},\mathsf{Sim}_{\mathbb{A}}}(\mathbb{A},\mathbb{D}_{\mathbb{A}}) \leqslant \mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}_1}(\mathbb{B}_1,\mathbb{D}_1) + \mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}_2}(\mathbb{B}_2,\mathbb{D}_2) + q_d \cdot \mathsf{max}_{\mathsf{M}} .$$

Rolling the reductions, distinguishers and simulators into one then yields the statement. □

3.2 PKE' is ISO-CCA Insecure

We next show that there is an ISO-CCA attack against PKE'. The attacker has overwhelming advantage for all but the most extreme message samplers, and the attack does not rely on any opening oracles.

Lemma 3. *Let* M *be a sampler with resampler* S *and maximum guessing probability at most* $\mathsf{max}_{\mathsf{M}}$*, and let* $\ell \in \mathbb{N}$ *be a parameter. Then there is an adversary* $\mathbb{A}_\ell$ *(Figure 10) which gains the following* ISO-CCA *advantage against* PKE'*:*

$$\mathsf{Adv}^{\text{iso-cca-n}}_{\mathsf{PKE}'[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A}_\ell) \geqslant 1 - \mathsf{max}^{\ell}_{\mathsf{M}} .$$

Aside from ℓ *calls to* $\mathcal{E}$ *and* $\mathcal{D}$ *and one call to* $\mathcal{C}$*,* $\mathbb{A}_\ell$ *only makes* ℓ *bit comparisons.*

Proof. The adversary $\mathbb{A}_\ell$ first calls the challenge oracle ℓ times on whatever inputs the sampler may expect (shown without input in Figure 10), collecting ℓ challenge ciphertexts c_i, before calling the challenge oracle $\mathcal{C}$, receiving a real or resampled message vector $\mathtt{M}_b$. It then calls the plaintext checking functionality of the decryption oracle $\mathcal{D}$ up to ℓ times, each time checking that the ith message $\mathtt{M}[i]$ corresponds to the ith challenge ciphertext; if the check ever returns false, then the message vector must have been resampled, and $\mathbb{A}_\ell$ returns ideal; otherwise it returns real.

Let bad be the event that the real message vector $\mathtt{M}_{\mathsf{real}}$ was resampled exactly, so that $\mathtt{M}_{\mathsf{real}} = \mathtt{M}_{\mathsf{ideal}}$. Since there are no openings, this is equivalent to simply sampling every message again from a re-initialized M. Thus, the probability that the message sampler was exactly resampled can be bounded as

$$\Pr[\mathsf{bad}] = \Pr\left[\forall_{i\in[\ell]}\ \mathtt{M}_{\mathsf{real}}[i] = \mathtt{M}_{\mathsf{ideal}}[i]\right] \leqslant \mathsf{max}^{\ell}_{\mathsf{M}} ,$$

where the product on the right-hand side follows from $\mathsf{max}_{\mathsf{M}}$ being defined as the maximum over all sampler inputs and states (see Definition 5).

Conditioned on $\neg\mathsf{bad}$, $\mathbb{A}_\ell$ always guesses correctly, whereas if bad then $\mathbb{A}_\ell$ has even

chance of their guess being correct. Hence,

$$\begin{aligned}
\mathsf{Adv}^{\text{iso-cca-n}}_{\mathsf{PKE}'[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A}_\ell) &= 2 \cdot \Pr\left[\mathsf{Exp}^{\text{iso-cca-n}}_{\mathsf{PKE}'[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A}_\ell)\right] - 1 \\
&= 2 \cdot \Big(\Pr[\mathsf{bad}] \cdot \Pr\left[\mathsf{Exp}^{\text{iso-cca-n}}_{\mathsf{PKE}'[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A}_\ell) \;\middle|\; \mathsf{bad}\right] \\
&\quad + (1 - \Pr[\mathsf{bad}]) \cdot \Pr\left[\mathsf{Exp}^{\text{iso-cca-n}}_{\mathsf{PKE}'[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A}_\ell) \;\middle|\; \neg\mathsf{bad}\right] \Big) - 1 \\
&= 2 \cdot \left(\Pr[\mathsf{bad}] \cdot \frac{1}{2} + (1 - \Pr[\mathsf{bad}]) \right) - 1 \\
&= 1 - \Pr[\mathsf{bad}] \geqslant 1 - \mathsf{max}^{\ell}_{\mathsf{M}} \,.
\end{aligned}$$

□

4 Low-Entropy Implication

The semi-separation presented in Section 3 only held relative to high-entropy message samplers, begging the question whether restricting to lower-entropy samplers will allow an implication to go through. We next answer this question in the affirmative. Concretely we show that, even though the distinguisher does not have access to a decryption oracle (of which it needs to provide an accurate simulation to the ISO adversary), there is a way for the reduction to provide the distinguisher with all the information it needs in order to provide an exact simulation of a decryption oracle to the adversary—*provided* the reduction was able to resample the real message vector in polynomially many attempts.[6] This then yields a partial implication that holds relative to any low-entropy message sampler.

Theorem 2 (Restricted SSO-CCA implies restricted ISO-CCA)**.** *For any PKE scheme and opening* $\mathrm{OP} \in \{\mathrm{N}, \mathrm{T}, \mathrm{S}, \mathrm{R}, \mathrm{B}\}$*, restricted* SSO-CCA-OP *security implies restricted* ISO-CCA-OP *security, relative to any low-entropy message sampler (Definition 6).*

Proof. The parameter $p_\ell = (1 - \mathsf{min}_{\mathsf{M}}(q_e))^\ell$ in the bound of Lemma 4 represents the probability that the strategy of the reduction failed, i.e. that it was not able to resample the real message vector in ℓ attempts. Observe that $p_\ell \to 1$ makes the bound vacuous, so that the implication fails for high-entropy samplers; in fact, the $+\,p_\ell$ term means any non-negligible p_ℓ will void the implication. Meanwhile, $p_\ell \to 0$ gives tightness, and for $p_\ell \leqslant \mathsf{negl}(\lambda)$, SSO security implies ISO security.

With M being a low-entropy sampler (and noting that $q_e \leqslant \mathsf{poly}(\lambda)$), we have that $\mathsf{min}_{\mathsf{M}}(q_e) \geqslant 1/\mathsf{poly}(\lambda)$. For concreteness, let $q(\lambda)$ be a polynomial of degree d such that $\mathsf{min}_{\mathsf{M}}(q_e) \geqslant 1/q(\lambda)$. Setting ℓ to be a degree $d+1$ polynomial, say, $\ell(\lambda) = \lambda \cdot q(\lambda)$, and using $1 - x \leqslant e^{-x}$, we get

$$p_\ell \leqslant \left(1 - \frac{1}{q(\lambda)}\right)^{\lambda \cdot q(\lambda)} \leqslant \exp\left(-\frac{\lambda \cdot q(\lambda)}{q(\lambda)}\right) = \exp\left(-\lambda\right) \,,$$

which decreases faster than any inverse-polynomial, hence $p_\ell \leqslant \mathsf{negl}(\lambda)$. □

Lemma 4. *Let* PKE *be a PKE scheme, let* M *be a message sampler with ideal resampler* $\mathring{\mathsf{S}}$ *and minimum resampling probability* $\mathsf{min}_{\mathsf{M}}(q_e)$ *after* q_e *samples (Definition 6). Then for any adversary* $\mathbb{A}$ *that makes* q_e *encryption queries, there exists a (non-blackbox) reduction* $\mathbb{B}$ *such that, for all simulators* Sim*, there exists a distinguisher* $\mathbb{D}$ *such that*

$$\mathsf{Adv}^{\text{n-iso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathring{\mathsf{S}}}(\mathbb{A}) \leqslant \frac{2}{1 - p_\ell} \cdot \mathsf{Adv}^{\text{n-sso-cca-b}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}}(\mathbb{B}, \mathbb{D}) + p_\ell \,,$$

[6] Note how this strategy fails for high-entropy samplers, as it had to.

Red. $\mathbb{B}^{\mathcal{E},\mathcal{D},\mathcal{S}}(\mathsf{pk})$

$q \leftarrow 0,\ \mathsf{out}_{\mathbb{B}} \leftarrow \varepsilon$
$\sigma \leftarrow\!\!\$\ \mathbb{A}_1^{\mathcal{E},\mathcal{D},\mathcal{S}}(\mathsf{pk})$
$\mathsf{out}_{\mathbb{B}} \twoheadleftarrow \sigma$
foreach $k \in [\ell]$:
  $\mathtt{M} \leftarrow\!\!\$\ \mathring{\mathsf{S}}(\mathtt{A}, \mathtt{L}, \mathcal{J}, \mathtt{M}_{\mathsf{real}}[\mathcal{J}])$
  $\mathtt{D} \leftarrow \varepsilon$
  $\hat{d} \leftarrow \mathbb{A}_2^{\mathcal{D}}(\mathtt{M}; \sigma)$
  $\mathsf{out}_{\mathbb{B}} \twoheadleftarrow (\mathtt{M}, \mathtt{D})$
return $\mathsf{out}_{\mathbb{B}}$

If $\mathbb{A}_1$ calls $\mathcal{D}(c)$

return $\mathcal{D}_{\mathbb{B}}(c)$

If $\mathbb{A}_1$ calls $\mathcal{E}(\alpha)$

$q \leftarrow q + 1,\ \mathtt{A} \twoheadleftarrow \alpha$
$(c^*, |m|) \leftarrow \mathcal{E}_{\mathbb{B}}(\alpha)$
$\mathtt{C} \twoheadleftarrow c^*,\ \mathtt{L} \twoheadleftarrow |m|$
return c^*

If $\mathbb{A}_1$ calls $\mathcal{S}(j)$

$\mathcal{J} \xleftarrow{\cup} j$
$(m, r) \leftarrow \mathcal{S}_{\mathbb{B}}(j)$
$\mathtt{M}_{\mathsf{real}}[j] \leftarrow m$
return (m, r)

If $\mathbb{A}_2$ calls $\mathcal{D}(c)$

$m \leftarrow \mathcal{D}_{\mathbb{B}}(c)$
$\mathtt{D} \twoheadleftarrow m$
return m

Dist. $\mathbb{D}(\mathtt{A}, \mathtt{M}_{\mathsf{real}}, \mathcal{J}, \mathsf{out})$

$d \leftarrow\!\!\$\ \{\mathsf{real}, \mathsf{ideal}\}$
if $\exists_{(\mathtt{M},\mathtt{D})\in\mathsf{out}}\ \mathtt{M} = \mathtt{M}_{\mathsf{real}}$:
  $\mathtt{D}_{\mathsf{real}} \leftarrow \mathtt{D}$
else : **return** d
$\sigma \leftarrow \mathsf{out}[1]$
$(\mathtt{M}_{\mathsf{ideal}}, \mathtt{D}_{\mathsf{ideal}}) \leftarrow \mathsf{out}[2]$
if $\mathsf{time}(\mathbb{A}_2^{\mathtt{D}_d}(\mathtt{M}_d; \sigma)) \leqslant T$
  $\hat{d} \leftarrow \mathbb{A}_2^{\mathtt{D}_d}(\mathtt{M}_d; \sigma)$
  if $d = \hat{d}$: **return** real
else : **return** ideal

Figure 11: The reduction $\mathbb{B}$ (left) and distinguisher $\mathbb{D}$ (right) of Lemma 4.

where $p_\ell = (1 - \mathsf{min}_{\mathsf{M}}(q_e))^\ell$. *The runtime of* $\mathbb{B}$ *is upper bounded by* ℓ *times that of* $\mathbb{A}$ *and* ℓ *invocations of* $\mathring{\mathsf{S}}$ *(plus some small overhead), and the runtime of* $\mathbb{D}$ *is upper bounded by that of* $\mathbb{A}$ *(plus some small overhead).*

Proof. For the sake of simplicity, we show the theorem for the case $n = 1$, with sender openings only. Since nothing crucial happens in the first stage of the game (pre ISO challenge), it is easy to verify that nothing significant changes in the proof if we add more users and a receiver opening oracle to both notions. (Essentially, the pseudocode of $\mathbb{B}$ will update with variables that store decrypted challenges that were encrypted under opened keys.)

We may without loss of generality assume that $\mathbb{A}$ calls $\mathcal{C}$ exactly once. This allows us to split $\mathbb{A}$ in two: Let $\mathbb{A}_1$ be $\mathbb{A}$ up until the point it calls $\mathcal{C}$, and denote by σ its internal state at that point, and let $\mathbb{A}_2$ be $\mathbb{A}$ after it receives the response $\mathtt{M}_b$ from $\mathcal{C}$, continuing from state σ. Crucially, $\mathbb{A}_2$ still gets access to a decryption oracle, but none of the others (technically, the reduction simulates their behaviour for $\mathbb{A}_2$ by always returning an error message, cf. Figure 5.)

Absent a decryption oracle, we might have w.l.o.g. assumed that $\mathbb{A}_2$ behaved deterministically, i.e. that all of $\mathbb{A}$'s random coins were pre-sampled and stored in σ. With $\mathbb{A}_2$ having access to a decryption oracle, we have to be more careful: Though we assume perfect correctness of the underlying PKE scheme, the existence of schemes with random decryption failures shows that we cannot in general rule out decryption oracles as a source of randomness. However, fixing the responses of the decryption oracle ensures that $\mathbb{A}_2$ behaves deterministically (and, without this source of randomness, $\mathbb{A}_2$ will always make the same calls to the oracle). This will be crucial below.[7]

Let $\mathbb{B}$ be as given in Figure 11 (left), running $\mathbb{A}_1$ while forwarding all oracle calls, and collecting its own copies of $\mathtt{A}$, $\mathtt{L}$, $\mathcal{J}$, and $\mathtt{M}_{\mathsf{real}}[\mathcal{J}]$ along the way. Once $\mathbb{A}_1$ terminates (by calling $\mathcal{C}$) with state σ, $\mathbb{B}$ stores σ in the first register of its output $\mathsf{out}_{\mathbb{B}}$, then repeats the following steps ℓ times:

1. Call the ideal resampler $\mathring{\mathsf{S}}$ to receive an alternative message vector $\mathtt{M}$;

[7] Such a derandomization procedure is not possible for quantum adversaries, and so whether a similar implication holds in the post-quantum setting remains open.

2. Run $\mathbb{A}_2$ on state σ and input $\mathtt{M}$;
3. Forward the decryption oracle, and store the responses in a fresh list $\mathtt{D}$;
4. Add tuple $(\mathtt{M}, \mathtt{D})$ to $\mathsf{out}_{\mathbb{B}}$.

The distinguisher $\mathbb{D}$ (Figure 11, right) checks if any of the resampled message vectors provided in out (real or simulated) matches the original message vector $\mathtt{M}_{\mathsf{real}}$. If not, it halts with a uniform guess; otherwise, depending on a drawn challenge bit d, it re-runs $\mathbb{A}_2$ on state σ and either the real message vector $\mathtt{M}_{\mathsf{real}}$ or a resampled one $\mathtt{M}_{\mathsf{ideal}}$, taken from the second register of out.[8]. Now, it may use the corresponding decryption transcript $\mathtt{D}_d$ to answer decryption oracle queries along the way (denoted $\mathbb{A}_2^{\mathtt{D}_d}$, in a slight abuse of notation). Due to the determinism of $\mathbb{A}_2$ under the same oracle responses, this is guaranteed to run $\mathbb{A}_2$ identically to its previous run on the same input, providing a faithful simulation of a decryption oracle as desired.

In the ideal game, there is no guarantee that $\mathsf{out}[1]$ (which was now produced by Sim) contains a valid σ (i.e. one that could have been generated by $\mathbb{A}_1$), and thus the runtime of $\mathbb{A}_2$ on such a state will no longer be bound by that of $\mathbb{A}$ itself. Therefore, the distinguisher explicitly bounds the runtime of $\mathbb{A}_2$ by some fixed $T > \mathsf{time}(\mathbb{A})$. Then, if $\mathbb{A}_2$ does not halt within time T, the SSO-CCA game must have been ideal.

If $\mathbb{A}_2$ terminates in time, $\mathbb{D}$ checks whether the output $\hat{d}$ of $\mathbb{A}_2$ matches the drawn challenge bit d. If yes, it outputs the guess real, and ideal otherwise.

Let b denote the challenge bit of the SSO-CCA-S game, let $\hat{b}$ denote the guess of the distinguisher, and let $\mathsf{bad}_{\mathtt{M}}$ denote the event that out does not contain the original message vector $\mathtt{M}_{\mathsf{real}}$. Then $\mathbb{D}$ wins with the following probability.

$$\begin{aligned}
\Pr\left[\mathsf{Exp}^{\text{sso-cca-s}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}}(\mathbb{B},\mathbb{D})\right] &= \Pr\left[b = \hat{b} \wedge \neg\mathsf{bad}_{\mathtt{M}}\right] + \Pr\left[b = \hat{b} \wedge \mathsf{bad}_{\mathtt{M}}\right] \\
&= \Pr[\neg\mathsf{bad}_{\mathtt{M}}] \Pr\left[b = \hat{b} \;\middle|\; \neg\mathsf{bad}_{\mathtt{M}}\right] \\
&\quad + (1 - \Pr[\neg\mathsf{bad}_{\mathtt{M}}]) \Pr\left[b = \hat{b} \;\middle|\; \mathsf{bad}_{\mathtt{M}}\right] \\
&= \Pr[\neg\mathsf{bad}_{\mathtt{M}}] \Pr\left[b = \hat{b} \;\middle|\; \neg\mathsf{bad}_{\mathtt{M}}\right] + \frac{1}{2}\left(1 - \Pr[\neg\mathsf{bad}_{\mathtt{M}}]\right) .
\end{aligned}$$

Let $\mathsf{min}_{\mathsf{M}}(q_e)$ be the minimum resampling probability of M, as defined in Definition 6, after at most q_e samples. Then, since each resampling is independent, the probability that none of the ℓ resamples yielded $\mathtt{M}_{\mathsf{real}}$ can be bounded as

$$\Pr[\mathsf{bad}_{\mathtt{M}} \mid b = \mathsf{real}] \leqslant (1 - \mathsf{min}_{\mathsf{M}}(q_e))^{\ell} \coloneqq p_{\ell} .$$

Bayes' theorem then gives us that

$$\Pr[b = \mathsf{real} \mid \neg\mathsf{bad}_{\mathtt{M}}] = \frac{\Pr[b = \mathsf{real}] \Pr[\neg\mathsf{bad}_{\mathtt{M}} \mid b = \mathsf{real}]}{\Pr[\neg\mathsf{bad}_{\mathtt{M}}]} \geqslant \frac{1 - p_{\ell}}{2 \Pr[\neg\mathsf{bad}_{\mathtt{M}}]} .$$

Next, let us restrict the analysis to simulators that produce an output $\mathsf{out} = (\mathtt{M}, \mathtt{D})$ satisfying $\Pr[\mathsf{bad}_{\mathtt{M}} \mid b = \mathsf{ideal}] \leqslant p_{\ell}$ (which can similarly be achieved via ℓ calls to $\mathring{\mathsf{S}}$). As above, we get that

$$\Pr[b = \mathsf{ideal} \mid \neg\mathsf{bad}_{\mathtt{M}}] \geqslant \frac{1 - p_{\ell}}{2 \Pr[\neg\mathsf{bad}_{\mathtt{M}}]} .$$

[8] It is possible that the real message vector was already found to be in the second register, leading to $\mathtt{M}_{\mathsf{real}} = \mathtt{M}_{\mathsf{ideal}}$ and no advantage. This is not an issue, as fixing the resampled message vector to always be taken from the second register ensures that $\mathtt{M}_{\mathsf{ideal}}$ is chosen according to the correct distribution, and in particular that the event $\mathtt{M}_{\mathsf{real}} = \mathtt{M}_{\mathsf{ideal}}$ occurs with the same probability as in the original experiment.

Next, separate the term $\Pr\Big[b = \hat{b} \;\Big|\; \neg\mathsf{bad}_{\mathtt{M}}\Big]$ into the "real" and "ideal" cases:

$$\begin{aligned}\Pr\Big[b = \hat{b} \;\Big|\; \neg\mathsf{bad}_{\mathtt{M}}\Big] &= \Pr[b = \mathsf{real} \mid \neg\mathsf{bad}_{\mathtt{M}}] \Pr\Big[b = \hat{b} \;\Big|\; b = \mathsf{real} \wedge \neg\mathsf{bad}_{\mathtt{M}}\Big] \\ &\quad + \Pr[b = \mathsf{ideal} \mid \neg\mathsf{bad}_{\mathtt{M}}] \Pr\Big[b = \hat{b} \;\Big|\; b = \mathsf{ideal} \wedge \neg\mathsf{bad}_{\mathtt{M}}\Big] \\ &\geqslant \frac{1 - p_\ell}{2 \Pr[\neg\mathsf{bad}_{\mathtt{M}}]} \Big(\Pr\Big[b = \hat{b} \;\Big|\; b = \mathsf{real} \wedge \neg\mathsf{bad}_{\mathtt{M}}\Big] \\ &\quad + \Pr\Big[b = \hat{b} \;\Big|\; b = \mathsf{ideal} \wedge \neg\mathsf{bad}_{\mathtt{M}}\Big] \Big) .\end{aligned}$$

Let us look at each term separately. For $\Pr\Big[b = \hat{b} \;\Big|\; b = \mathsf{real} \wedge \neg\mathsf{bad}_{\mathtt{M}}\Big]$, $\mathbb{A}$ is run start to finish in an honest simulation of the ISO experimment; hence, the running time of $\mathbb{A}_2$ is bounded by that of $\mathbb{A}$ itself, and the strategy of $\mathbb{D}$ yields

$$\Pr\Big[b = \hat{b} \;\Big|\; b = \mathsf{real} \wedge \neg\mathsf{bad}_{\mathtt{M}}\Big] = \Pr\Big[d = \hat{d} \;\Big|\; b = \mathsf{real} \wedge \neg\mathsf{bad}_{\mathtt{M}}\Big] .$$

Moreover, $\mathbb{A}$ is given a faithful simulation of $\mathsf{Exp}^{\text{iso-cca-s}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{S}}(\cdot)$, so that we have

$$\Pr\Big[d = \hat{d} \;\Big|\; b = \mathsf{real} \wedge \neg\mathsf{bad}_{\mathtt{M}}\Big] = \Pr\Big[\mathsf{Exp}^{\text{iso-cca-s}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A})\Big] .$$

For the second term $\Pr\Big[b = \hat{b} \;\Big|\; b = \mathsf{ideal} \wedge \neg\mathsf{bad}_{\mathtt{M}}\Big]$, let bad_T denote the event that the runtime of $\mathbb{A}_2$ exceeds T when run by the distinguisher, which (since $T > \mathsf{time}(\mathbb{A})$) then wins by returning ideal. Then we can rewrite

$$\begin{aligned}&\Pr\Big[b = \hat{b} \;\Big|\; b = \mathsf{ideal} \wedge \neg\mathsf{bad}_{\mathtt{M}}\Big] \\ &\quad = \Pr[\mathsf{bad}_T] + (1 - \Pr[\mathsf{bad}_T]) \Pr\Big[d \neq \hat{d} \;\Big|\; b = \mathsf{ideal} \wedge \neg\mathsf{bad}_{\mathtt{M}} \wedge \neg\mathsf{bad}_T\Big] ,\end{aligned}$$

where we also used that $\Pr[\mathsf{bad}_T \mid b = \mathsf{real} \wedge \neg\mathsf{bad}_{\mathtt{M}}] = 0$. For the remaining expression, note that out and hence σ was produced by Sim without access to any information on the unopened messages, implying that the challenge bit d is information-theoretically hidden from $\mathbb{A}_2$, making it evaluate to $1/2$. Then,

$$\begin{aligned}\Pr\Big[b = \hat{b} \;\Big|\; b = \mathsf{ideal} \wedge \neg\mathsf{bad}_{\mathtt{M}}\Big] &= \Pr[\mathsf{bad}_T] + \frac{1}{2}(1 - \Pr[\mathsf{bad}_T]) \\ &= \frac{1}{2}\Pr[\mathsf{bad}_T] + \frac{1}{2} .\end{aligned}$$

Pulling the above pieces together then gives us

$$\Pr\Big[b = \hat{b} \;\Big|\; \neg\mathsf{bad}_{\mathtt{M}}\Big] \geqslant \frac{1 - p_\ell}{2 \Pr[\neg\mathsf{bad}_{\mathtt{M}}]} \left(\Pr\Big[\mathsf{Exp}^{\text{iso-cca-s}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A})\Big] + \frac{1}{2}\Pr[\mathsf{bad}_T] + \frac{1}{2} \right) ,$$

and with a little algebraic manipulation,

$$\begin{aligned}
\mathsf{Adv}^{\text{sso-cca-s}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}}(\mathbb{B},\mathbb{D}) &= 2\cdot\Pr\left[\mathsf{Exp}^{\text{sso-cca-s}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}}(\mathbb{B},\mathbb{D})\right]-1\\
&= 2\cdot\Big(\Pr[\neg\mathsf{bad}_{\mathtt{M}}]\cdot\Pr\left[b=\hat{b}\;\middle|\;\neg\mathsf{bad}_{\mathtt{M}}\right]\\
&\quad+\frac{1}{2}\left(1-\Pr[\neg\mathsf{bad}_{\mathtt{M}}]\right)\Big)-1\\
&= 2\cdot\Pr[\neg\mathsf{bad}_{\mathtt{M}}]\cdot\Pr\left[b=\hat{b}\;\middle|\;\neg\mathsf{bad}_{\mathtt{M}}\right]-\Pr[\neg\mathsf{bad}_{\mathtt{M}}]\\
&\geqslant(1-p_\ell)\left(\Pr\left[\mathsf{Exp}^{\text{iso-cca-s}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A})\right]+\frac{1}{2}\Pr[\mathsf{bad}_T]+\frac{1}{2}\right)\\
&\quad-\Pr[\neg\mathsf{bad}_{\mathtt{M}}]\,,
\end{aligned}$$

throw away the irrelevant bad_T term and apply the trivial bound to $\Pr[\neg\mathsf{bad}_{\mathtt{M}}]$,

$$\begin{aligned}
&\geqslant(1-p_\ell)\left(\Pr\left[\mathsf{Exp}^{\text{iso-cca-s}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A})\right]+\frac{1}{2}\right)-1\\
&=\frac{1-p_\ell}{2}\left(2\cdot\Pr\left[\mathsf{Exp}^{\text{iso-cca-s}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A})\right]+1\right)-1\\
&=\frac{1-p_\ell}{2}\left(2\cdot\Pr\left[\mathsf{Exp}^{\text{iso-cca-s}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{S}}(\mathbb{A})\right]-1-\frac{2\cdot p_\ell}{1-p_\ell}\right)\\
&=\frac{1-p_\ell}{2}\mathsf{Adv}^{\text{iso-cca-s}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathring{\mathsf{S}}}(\mathbb{A})-p_\ell\,,
\end{aligned}$$

which yields the statement. □

5 A New CCA Landscape for SOA

Our separation result Theorem 1 holds also absent openings. At a glance, this may seem to contradict the well-known equivalence between indistinguishability and semantic security, which is known to also hold in the CCA setting [WSI03]. However, this apparent contradiction stems from a fallacy, as we will see next.

5.1 IND-CCA in the High-Entropy Setting

As pointed out by BHS, the need to quantify the message samplers means we need to be extra careful when stringing together implications between notions of SOA. Since SSO security is defined relative to all message samplers, and standard indistinguishability does not involve samplers, it suffices to show that there exists a message sampler relative to which IND-CCA security follows; this is the left-or-right sampler of Figure 3.

If, on the other hand, the SOA notion is restricted to high-entropy samplers, a separation appears.

Theorem 3 (High-entropy SOA does not imply IND-CCA)**.** *Let* M *be any high-entropy message sampler. Then there exists a PKE scheme that simultaneously achieves (restricted)* SSO-CCA-B *and* ISO-CCA-B *security relative to* M*, yet is trivially* IND-CCA *insecure.*

Sketch. Let PKE be a PKE scheme simultaneously satisfying SSO-CCA-B and ISO-CCA-B,[9] and let PKE' be identical to PKE except punctured on a single, public value, e.g. $m=\mathsf{pm}$ (for all pk, $\mathsf{PKE}'.\mathsf{Enc}_{\mathsf{pk}}(\mathsf{pm})=\mathsf{pm}$). Then, IND-CCA is trivially broken, since the punctured value can be freely selected as one of the two challenge messages; and yet any high-entropy messages sampler will have negligible probability of sampling the punctured value, giving the SSO resp. ISO adversary only negligible additional advantage relative to PKE. □

[9] Any scheme satisfying non-committing encryption (NCE-CCA-B) will suffice [BHS23].

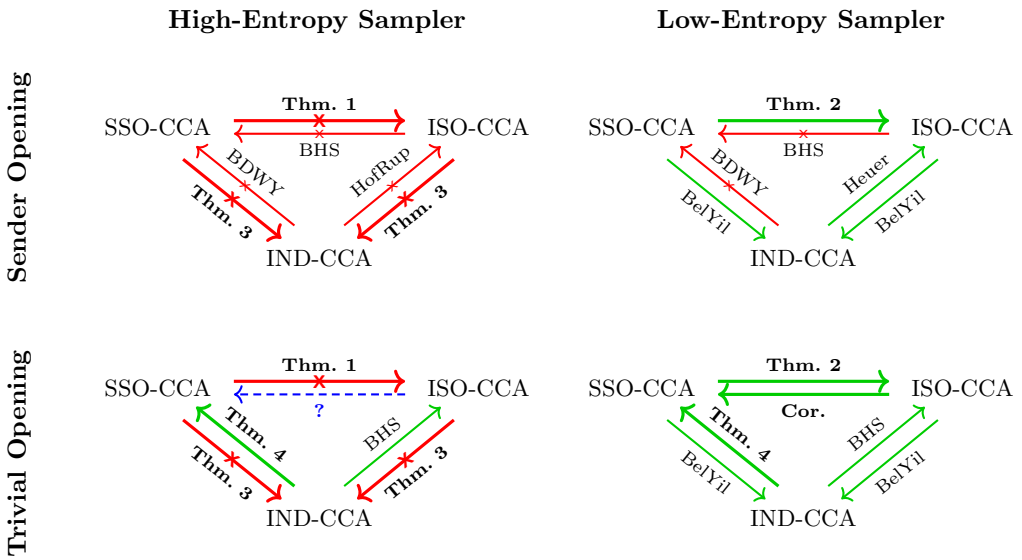


Figure 12: Relations between notions in the high- and low-entropy regimes, with and without non-trivial (sender) opening. Implications are shown in green, separations are shown in red with a cross, and the blue dashed arrow is open. Our results are highlighted in bold. Implications are for all message samplers of the relevant kind, while separations only requires that there exists a high-entropy resp. low-entropy sampler such that the notions are (semi-)separated relative to that sampler. Here, BDWY is Bellare et al. [BDWY12], HofRup is Hofheinz and Rupp [HR14], BelYil is Bellare and Yilek [BY12], BHS is Brunetta et al. [BHS24], and Heuer is Heuer [Heu17].

Taken together with Theorem 1 and Theorem 2, a picture starts to emerge: Restricting to high-entropy message samplers yields semi-separations, while restricting to low-entropy message samplers opens for partial implications. Indeed, Heuer [Heu17] already showed that IND-CCA implies ISO-CCA-S for any graph-induced message sampler traversable in polynomial time, which includes any low-entropy sampler. In fact, Heuer's implication also holds relative to uniform samplers, meaning that, unlike from SSO to ISO, there are high-entropy samplers relative to which IND-CCA implies ISO-CCA-T.

And yet, the notions are separated: Hofheinz et al. [HR14] showed that there are message samplers with strongly correlated outputs relative to which ISO-CCA-S is strictly stronger. Meanwhile, Bellare et al. [BDWY12] showed that no committing encryption scheme achieves SSO-CPA-S relative to *any* sampler. Since many committing encryption schemes achieve IND-CCA, this separates SSO-CPA-S (and therefore SSO-CCA-S) from IND-CCA. As pointed out by BHS, this can further be combined with Heuer's partial implication from IND-CCA to ISO-CCA-S to separate SSO-CPA-S from ISO-CCA-S.

The resulting relations when restricting to high-entropy resp. low-entropy samplers are summarized in Figure 12 (top).

5.2 IND-CCA Implies SOA with Trivial Opening

In a direction orthogonal to sampler entropy, Bellare et al. [BY12] showed that in the CPA setting, restricting to only transmission openings makes all three notions equivalent. BHS extended this to the CCA setting by showing that IND-CCA implies ISO-CCA-T, but left open whether SSO-CCA-T is also implied [BHS24, Open Problem 11]. We close this gap next.

Theorem 4. *For all PKE schemes,* IND-CCA *security implies* SSO-CCA-T *security relative to any message sampler. Furthermore, the reduction is tight.*

Sketch. We need to show that if PKE satisfies that the IND-CCA advantage of any adversary is negligible, then there is a simulator for every SSO-CCA-T adversary such that any distinguisher can distinguish them with at most negligible advantage. As usual we show the contrapositive: if there exist SSO-CCA adversary $\mathbb{A}$ and distinguisher $\mathbb{D}$ with a winning strategy against any simulator, then there is a simulator Sim relative to which our IND-CCA reduction $\mathbb{B}$ inherits their advantage.

Instead of standard IND-CCA, let $\mathbb{B}$ be playing the real-or-random indistinguishability game ROR-CCA, like IND-CCA except tasked with distinguishing encryptions of chosen messages from encryptions of uniform messages of the same length. The reduction simulates the SSO-CCA-T game for an adversary $\mathbb{A}$: when requested a challenge on input α, $\mathbb{B}$ samples message $m \leftarrow\!\!\$\ \mathsf{M}_{\langle s \rangle}(\alpha)$, forwards m to its own challenge oracle and returns the result, storing the sampled message for future calls to the transmission opening oracle $\mathcal{T}$. If $\mathbb{B}$ is playing the real game ($b = 0$), then this simulation is consistent with the real SSO-CCA-T game, and $\mathbb{A}$ will halt with some output out to be judged by distinguisher $\mathbb{D}$. If $\mathbb{B}$ is playing the random game ($b = 1$), then the view of $\mathbb{D}$ is consistent with the simulated SSO-CCA-T game instantiated with a simulator that produces its output out by itself simulating the SSO-CCA-T game for $\mathbb{A}$, as follows: after sampling public/private keypair $(\mathsf{pk}, \mathsf{sk})$, on input α, Sim forwards α to its own challenge oracle to receive a message length $|m|$, then uniformly samples and encrypts a message of the same length. With access to sk, the simulator can simulate the decryption oracle directly, whereas calls to $\mathcal{T}$ are simply forwarded. Once $\mathbb{D}$ halts with a guess $\hat{b}$, $\mathbb{B}$ halts with the same output.

Since the real and ideal worlds of ROR-CCA and SSO-CCA-T match up exactly, this gives a tight reduction, and with ROR-CCA itself tightly implied by standard IND-CCA [BDPR98] the theorem statement follows. □

The detailed reduction is presented in the full version. As a corollary to Theorem 4, we get that with low-entropy samplers, ISO-CCA-T implies SSO-CCA-T. We leave open the final question of whether ISO-CCA-T implies SSO-CCA-T also relative to high-entropy samplers, or whether the notions turn incomparable (shown as a blue dashed arrow in Figure 12).

5.3 Surveying the New CCA Landscape

The two orthogonal directions, high-vs-low entropy and trivial-vs-nontrivial opening, give rise to four "worlds", shown in Figure 12, from high-entropy samplers with sender opening at the top-left to low-entropy samplers with trivial (transmission or no) opening at the bottom right. (The figure shows sender opening, as several of the quoted results have only been shown for this setting, though we expect the results to be extendable to the receiver and bi-opening settings.)

Going clockwise from the top-left: in the high-entropy, sender opening world, all three notions are incomparable. In the low-entropy, sender opening world (top-right), SSO-CCA is the stronger notion, with ISO-CCA and IND-CCA equivalent. In the low-entropy, trivial opening world (lower right), all notions turn equivalent. Finally, in the high-entropy, trivial opening world, the expected hierarchy flips on its head, with IND-CCA the either the strongest notion, or tied for strongest with ISO-CCA. We leave the final question, of whether ISO-CCA-T implies SSO-CCA-T in this world, or the notions are incomparable, open.

6 Concluding Remarks

The discussion in Section 5 highlights the special care that must be taken when considering relations between notions of security that involve message sampling. Moreover, as explained in Remark 1, our notions of high-entropy and low-entropy samplers only represent two extremes on a wide spectrum, and for "mid-entropy" samplers that satisfy neither definition, several relations remain open. New tools may be needed to argue about implications and separations between notions of SOA relative to such samplers, and lacking such general tools, relations may have to be considered on a sampler-by-sampler basis.

Arguably, this situation represents a weakness for the SOA approach to modelling adaptive corruptions, and the complexity that come with message samplers may help explain its lack of adoption in constructions of higher-level protocols. BHS showed that (contrary to folklore) randomness leakage may be formalized as an indistinguishability-style security game by letting the game have multiple challenge bits, a subset of which may be adaptively corrupted; this notion is then equivalent to standard IND-CCA up to a loss liner in the number of (users and) challenge bits. They further observed that, for protocol design, whenever the standard and easy-to-work-with notions of indistinguishability fall short (for example because protocol security relies on opening challenges, such as when arguing about threshold security) the preferred route appears to be the Universal Composability (UC) framework [Can00], which can be extended to include openings [Wik16], making the notion equivalent to non-committing encryption [BHS24]. In this framework, rather than imitating an adversary, the simulator is simulating the environment, with the adversary itself serving as the distinguisher. This not only leads to fewer players (two in the UC/NCE experiment, versus three for SSO), simplifying reductions, but crucially avoids the need to argue about message distributions at all.

Acknowledgement

The author is indebted to Dennis Hofheinz for many helpful comments, in conversation with whom the main ideas for the separation first appeared.

References

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Berlin, Heidelberg, May 2000. doi:10.1007/3-540-45539-6_18.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 26–45. Springer, Berlin, Heidelberg, August 1998. doi:10.1007/BFb0055718.

[BDWY12] Mihir Bellare, Rafael Dowsley, Brent Waters, and Scott Yilek. Standard security does not imply security against selective-opening. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 645–662. Springer, Berlin, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_38.

[BHS23] Carlo Brunetta, Hans Heum, and Martijn Stam. Multi-instance secure public-key encryption. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part II*, volume 13941 of *LNCS*, pages 336–367. Springer, Cham, May 2023. doi:10.1007/978-3-031-31371-4_12.

[BHS24] Carlo Brunetta, Hans Heum, and Martijn Stam. SoK: Public key encryption with openings. In Qiang Tang and Vanessa Teague, editors, *PKC 2024, Part II*, volume 14604 of *LNCS*, pages 35–68. Springer, Cham, April 2024. doi:10.1007/978-3-031-57728-4_2.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, Berlin, Heidelberg, April 2009. doi:10.1007/978-3-642-01001-9_1.

[BJLS16] Christoph Bader, Tibor Jager, Yong Li, and Sven Schäge. On the impossibility of tight cryptographic reductions. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 273–304. Springer, Berlin, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_10.

[BY12] Mihir Bellare and Scott Yilek. Encryption schemes secure under selective opening attack. Cryptology ePrint Archive, Report 2009/101 (updated full version), 2012. https://eprint.iacr.org/2009/101, version 20120923:212424.

[Can00] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067, 2000. URL: https://eprint.iacr.org/2000/067.

[DNRS99] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *40th FOCS*, pages 523–534. IEEE Computer Society Press, October 1999. doi:10.1109/SFFCS.1999.814626.

[FHKP16] Georg Fuchsbauer, Felix Heuer, Eike Kiltz, and Krzysztof Pietrzak. Standard security does imply security against selective opening for Markov distributions. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 282–305. Springer, Berlin, Heidelberg, January 2016. doi:10.1007/978-3-662-49096-9_12.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *14th ACM STOC*, pages 365–377. ACM Press, May 1982. doi:10.1145/800070.802212.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001. doi:10.1017/CBO9780511546891.

[Heu17] Felix Heuer. *On the selective opening security of public-key encryption*. Doctoral thesis, Ruhr-Universität Bochum, Universitätsbibliothek, 2017.

[HPW15] Carmit Hazay, Arpita Patra, and Bogdan Warinschi. Selective opening security for receivers. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 443–469. Springer, Berlin, Heidelberg, November / December 2015. doi:10.1007/978-3-662-48797-6_19.

[HR14] Dennis Hofheinz and Andy Rupp. Standard versus selective opening security: Separation and equivalence results. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 591–615. Springer, Berlin, Heidelberg, February 2014. doi:10.1007/978-3-642-54242-8_25.

[HRW16] Dennis Hofheinz, Vanishree Rao, and Daniel Wichs. Standard security does not imply indistinguishability under selective opening. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 121–145. Springer, Berlin, Heidelberg, October / November 2016. doi:10.1007/978-3-662-53644-5_5.

[HS21] Hans Heum and Martijn Stam. Tightness subtleties for multi-user pke notions. In Maura B. Paterson, editor, *Cryptography and Coding*, pages 75–104, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-92641-0_5.

[LYHW21] Junzuo Lai, Rupeng Yang, Zhengan Huang, and Jian Weng. Simulation-based bi-selective opening security for public key encryption. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part II*, volume 13091 of *LNCS*, pages 456–482. Springer, Cham, December 2021. doi:10.1007/978-3-030-92075-3_16.

[Nie02] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 111–126. Springer, Berlin, Heidelberg, August 2002. doi:10.1007/3-540-45708-9_8.

[Wik16] Douglas Wikström. Simplified universal composability framework. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 566–595. Springer, Berlin, Heidelberg, January 2016. doi:10.1007/978-3-662-49096-9_24.

[WSI03] Yodai Watanabe, Junji Shikata, and Hideki Imai. Equivalence between semantic security and indistinguishability against chosen ciphertext attacks. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 71–84. Springer, Berlin, Heidelberg, January 2003. doi:10.1007/3-540-36288-6_6.

A ROR-CCA Tightly Implies SSO-CCA-T

Lemma 5. *Let* PKE *be a PKE scheme. Then there there exists a reduction* $\mathbb{B}_{\text{ror}}$ *and a simulator* Sim *such that for all adversaries* $\mathbb{A}_{\text{sso}}$*, distinguishers* $\mathbb{D}_{\text{sso}}$ *and message samplers* M*,*

$$\mathsf{Adv}^{\text{n-sso-cca-t}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}}(\mathbb{A}_{\text{sso}},\mathbb{D}_{\text{sso}}) \leqslant \mathsf{Adv}^{\text{n-ror-cca}}_{\mathsf{PKE}[\lambda]}(\mathbb{B}_{\text{ror}})\,,$$

where $\mathbb{B}_{\text{ror}}$ *and* Sim *each run in time roughly that of* $\mathbb{A}_{\text{sso}}$*.*

Proof. For the sake of simplicity, we show the result for the $n = 1$ case; nothing of essence changes in the reduction when upgrading both notions to multiple users.

The reduction $\mathbb{B}_{\text{ror}}$ (Figure 13, top) plays the real-or-random game, in which the challenge oracle takes a single message m (cf. Figure 4), and returns either an encryption of that message (if $b = 0$) or the encryption of a uniformly random message of the same length (if $b = 1$). The reduction simulates the SSO-CCA-T challenge oracle for $\mathbb{A}_{\text{sso}}$ by sampling messages from the message sampler M and forwarding them to its own challenge oracle, storing them in $\mathtt{M}_{\mathsf{real}}$ in order to later answer calls to the transmission opening oracle $\mathcal{T}$. If $b = 0$, this is a perfect simulation of the real-world SSO-CCA-T game.

If $b = 1$, then all challenge ciphertexts are encryptions of uniformly sampled messages. Let the simulator Sim be as given in Figure 13 (bottom), simulating the SSO-CCA-T game for $\mathbb{A}$ by providing challenge ciphertexts that encrypt uniformly sampled messages of the same length as the sampled message. From the view of $\mathbb{D}_{\text{sso}}$, the simulation of $\mathbb{B}_{\text{ror}}$ is then identical to the ideal SSO-CCA-T game instantiated with Sim.

Reduction $\mathbb{B}_{\text{ror}}^{\mathcal{E},\mathcal{D}}(\mathsf{pk})$

$\mathsf{out} \leftarrow\!\$\ \mathbb{A}_{\text{sso}}^{\mathcal{E},\mathcal{D},\mathcal{T}}(\mathsf{pk})$

$\hat{b} \leftarrow\!\$\ \mathbb{D}_{\text{sso}}(\mathsf{pm}, \mathtt{A}, \mathtt{M}_{\mathsf{real}}, \mathcal{J}, \mathsf{out})$

return $\hat{b}$

If $\mathbb{A}_{\text{sso}}$ calls $\mathcal{E}(\alpha)$

$\mathtt{A} \twoheadleftarrow \alpha$

$m \leftarrow\!\$\ \mathsf{M}_{\langle s \rangle}(\alpha)$

$\mathtt{M}_{\mathsf{real}} \twoheadleftarrow m$

$c^* \leftarrow \mathcal{E}_{\mathbb{B}}(m)$

return $(|m|, c^*)$

If $\mathbb{A}_{\text{sso}}$ calls $\mathcal{D}(c)$

return $\mathcal{D}_{\mathbb{B}}(c)$

If $\mathbb{A}_{\text{sso}}$ calls $\mathcal{T}(j)$

$\mathcal{J} \xleftarrow{\cup} j$

return $\mathtt{M}_{\mathsf{real}}[j]$

Simulator $\mathsf{Sim}^{\mathcal{E},\mathcal{T}}(\mathsf{pm})$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{PKE.Kg}(\mathsf{pm})$

$\mathsf{out} \leftarrow\!\$\ \mathbb{A}_{\text{sso}}^{\mathcal{E},\mathcal{D},\mathcal{T}}(\mathsf{pk})$

return out

If $\mathbb{A}_{\text{sso}}$ calls $\mathcal{E}(\alpha)$

$|m| \leftarrow \mathcal{E}_{\mathsf{Sim}}(\alpha)$

$m' \leftarrow\!\$\ \{0,1\}^{|m|}$

$c^* \leftarrow\!\$\ \mathsf{PKE.Enc}_{\mathsf{pk}}(m')$

$\mathcal{C} \xleftarrow{\cup} c^*$

return $(|m|, c^*)$

If $\mathbb{A}_{\text{sso}}$ calls $\mathcal{D}(c)$

if $c \in \mathcal{C}$: **return** $\lightning$

return $\mathsf{PKE.Dec}_{\mathsf{sk}}(c)$

If $\mathbb{A}_{\text{sso}}$ calls $\mathcal{T}(j)$

return $\mathcal{T}_{\mathsf{Sim}}(j)$

Figure 13: The reduction $\mathbb{B}_{\text{ror}}$ (top) and simulator Sim (bottom) of Lemma 5.

Once $\mathbb{D}_{\text{sso}}$ halts with a guess, $\mathbb{B}_{\text{ror}}$ halts with the same guess. Let G_0 be the real SSO-CCA-T game (out generated by $\mathbb{A}_{\text{sso}}$) and G_1 be the ideal game (out generated by Sim), respectively. Then:

$$\begin{aligned}
\Pr\left[\mathsf{Exp}^{\text{ror-cca}}_{\mathsf{PKE}[\lambda]}(\mathbb{B}_{\text{ror}})\right] &= \Pr\left[\hat{b} \leftarrow\!\$\ \mathbb{B}_{\text{ror}} \ :\ \hat{b} = b\right] = \Pr\left[\hat{b} \leftarrow\!\$\ \mathbb{D}_{\text{sso}} \ :\ \hat{b} = b\right] \\
&= \Pr[\mathrm{G}_b(\mathbb{D}_{\text{sso}}) \ :\ b \leftarrow\!\$\ \mathbb{D}_{\text{sso}}] = \Pr\left[\mathsf{Exp}^{\text{sso-cca-t}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}}(\mathbb{A}_{\text{sso}}, \mathbb{D}_{\text{sso}})\right], \\
\iff \mathsf{Adv}^{\text{ror-cca}}_{\mathsf{PKE}[\lambda]}(\mathbb{B}_{\text{ror}}) &= \mathsf{Adv}^{\text{sso-cca-t}}_{\mathsf{PKE}[\lambda],\mathsf{M},\mathsf{Sim}}(\mathbb{A}_{\text{sso}}, \mathbb{D}_{\text{sso}})\,.
\end{aligned}$$

□