Standard versus Selective Opening Security: Separation and Equivalence Results

Dennis Hofheinz and Andy Rupp

Karlsruhe Institute of Technology, Germany
{dennis.hofheinz,andy.rupp}@kit.edu

Supported by DFG grant GZ HO 4534/2-1

Abstract

Suppose many messages are encrypted using a public-key encryption scheme. Imagine an adversary that may adaptively ask for openings of some of the ciphertexts. Selective opening (SO) security requires that the *unopened* ciphertexts remain secure, in the sense that this adversary cannot derive any nontrivial information about the messages in the unopened ciphertexts.

Surprisingly, the question whether SO security is already implied by standard security notions has proved highly nontrivial. Only recently, Bellare, Dowsley, Waters, and Yilek (Eurocrypt 2012) could show that a strong form of SO security, *simulation-based* SO security, is not implied by standard security notions. It remains wide open, though, whether the potentially weaker (and in fact comparatively easily achievable) form of *indistinguishability-based* SO (i.e., IND-SO) security is implied by standard security. Here, we give (full and partial) answers to this question, depending on whether active or passive attacks are considered.

Concretely, we show that:

(a) For active (i.e., chosen-ciphertext) security, standard security does *not* imply IND-SO security. Concretely, we give a scheme that is IND-CCA, but not IND-SO-CCA secure.

(b) In the case of passive (i.e., chosen-plaintext) security, standard security *does* imply IND-SO security, at least in a generic model of computation and for a large class of encryption schemes. (Our separating scheme from (a) falls into this class of schemes.)

Our results show that the answer to the question whether standard security implies SO security highly depends on the concrete setting.

Keywords: security definitions, public-key encryption, selective opening security.

1 Introduction

Motivation. It is a challenging task to find a useful and achievable definition of security for encryption schemes. There seems to be no "one size fits all" security notion; for instance, certain settings involve key-dependent messages (e.g., [7, 9, 2]) or leakage of key material (e.g., [18, 13, 1]). In most of these specific settings, it is easily seen that standard encryption security notions (such as IND-CPA or IND-CCA security) do not provide any reasonable security guarantees. However, one particularly challenging setting is the setting of *selective opening attacks*, which models a specific (and realistic) form of adaptive corruptions. The topic of this paper is the connection of standard and selective opening security.

Selective opening attacks. The premise of a selective opening (SO) attack is as follows: suppose an adversary observes many ciphertexts c_i, and then gets to request openings of some of them. (Here, an opening corresponds to an adaptive corruption of the sender, and yields not only the plaintext m_i but also the random coins used during encryption.) The question is: can the adversary learn anything about the *unopened* m_i? Of course, if the encrypted messages are related, then the opened messages may already reveal information about the unopened messages. (In fact, this is the main source of trouble when trying

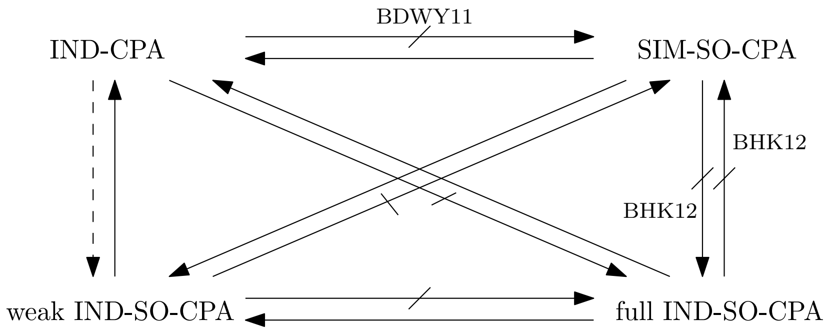

Figure 1: Relations among notions of selective opening security and IND-CPA security. Solid arrows denote implications, crossed arrows denote concrete counterexamples, and the dashed arrow stands for the remaining open question investigated in this paper.

to define selective opening security.) However, we would like to express that the unopened messages remain "as secure as possible", given the opened messages.

Selective opening security notions... Dwork et al. [12] were the first to propose a formal SO security notion; their notion is simulation-based and was formulated for commitments. Bellare et al. [5] gave a public-key encryption (PKE) version of the definition of [12] (SIM-SO-CPA[1]), along with a weaker, indistinguishability-based notion (weak IND-SO-CPA).[2] Most relations among SO security notions (and between SO and standard security notions) have already been investigated (see also Figure 1). Specifically, [8] provided separations[3] between SO notions, and Bellare et al. [4] have separated SIM-SO-CPA security from IND-CPA security. The *only* remaining open question (that we approach in this paper) is thus

Does standard security already imply *indistinguishability-based* selective opening security?

...and constructions. Bellare et al. [5] proved lossy encryption [22, 21, 20] weakly IND-SO-CPA secure, and the scheme of Goldwasser and Micali [15] SIM-SO-CPA secure. Subsequently, several works have developed chosen-ciphertext secure (i.e., weakly IND-SO-CCA and SIM-SO-CCA secure) PKE schemes [14, 16, 17]. However, it seems safe to say that (weak) indistinguishability-based SO security is significantly easier to achieve than simulation-based SO security. In particular, the most efficient SO secure PKE schemes are not known to be SIM-SO secure. This makes the question whether standard security implies weak IND-SO security even more interesting.

Our contribution. We tackle this last remaining question both in the chosen-plaintext (CPA) and in the chosen-ciphertext (CCA) case. We give a definite answer in the CCA case and a partial answer in the CPA case. First, we separate IND-CCA and IND-SO-CCA security: we give an IND-CCA secure but IND-SO-CCA insecure PKE scheme. Our result utilizes the standard model of computation and works under the minimal assumption that IND-CCA secure PKE schemes exist. Nonetheless, the IND-SO-CCA attack on our scheme is completely generic and does not make use of, e.g., non-black-box techniques (such as using the internal structure of the IND-CCA secure scheme). Our second result shows that IND-CPA and IND-SO-CPA security are equivalent in a generic model of computation and with respect to a restricted class of PKE schemes. We stress that the generic model considered for the CPA equivalence is realistic: it covers, e.g., ElGamal, Cramer-Shoup and similar encryption schemes, and in fact also concrete instantiations (e.g., based on Cramer-Shoup) of our separating example for the CCA case (including our attack on its weak IND-SO-CCA security). Interestingly, [4] shows that there is no such equivalence in the case of SIM-SO-CPA for the class of committing encryption schemes which also includes ElGamal and Cramer-Shoup. The adversary for which they can show that no simulator exists is a simple generic algorithm.

[1] The naming of SO notions is not quite consistent in the literature. We follow the naming of Böhl et al. [8].

[2] There is also a stronger indistinguishability-based SO notion called *full* IND-SO-CPA. Weak and full IND-SO-CPA security differ in the sense that the considered (joint) message distributions are arbitrary in full IND-SO-CPA, but restricted in weak IND-SO-CPA security. No fully IND-SO-CPA secure schemes are known.

[3] Here, with a "separation" between two security notions X and Y, we mean that there is a scheme that achieves X but not Y (or vice versa). We do *not* mean that a Y-secure scheme cannot be constructed from an X-secure one (or vice versa).

Another interesting point of view on our results is the following: For a broad class of encryption schemes (including instances of our separating scheme), it holds that any generic IND-SO-CPA adversary can be turned into a generic IND-CPA adversary, while this does not hold in the CCA case. For instance, there exists an efficient generic IND-SO-CCA adversary against our separating scheme, while there are no generic (or even non-generic) IND-CCA adversaries.

Details on our IND-CCA/IND-SO-CCA separation. To construct our separating scheme, we take an arbitrary IND-CCA secure PKE scheme and modify it such that a weak IND-SO-CCA attack becomes possible. To understand the basic idea behind our modification, recall that in the weak IND-SO-CCA experiment, an adversary A first receives a ciphertext vector $\mathbf{c} = (c_i)_i$ with $c_i \leftarrow \mathsf{Enc}(pk, m_i)$ for messages m_i sampled from a (joint) adversarially selected message distribution $\mathcal{D}$. A can then select a subset $\mathcal{I}$ of all c_i to be opened. In addition to the openings of all m_i (for $i \in \mathcal{I}$), A also receives a full message vector $\mathbf{m}$ which *either* consists of all actually encrypted messages m_i, *or* of messages m'_i freshly sampled from $\mathcal{D}$, conditioned on $m'_i = m_i$ for all $i \in \mathcal{I}$. As usual, A has to decide which case it is. Thus, A has to distinguish between the encrypted messages and messages that are "just as plausible" given only the opened messages.

To obtain our separating scheme, we take an IND-CCA secure scheme and modify its decryption algorithm. Namely, we now allow a special type of decryption queries $(\mathtt{soa}, Z)$ in which Z contains a whole ciphertext vector $\mathbf{c}$, along with openings of a subset of these ciphertexts. (For now, it is easiest to imagine that this subset is selected externally and randomly.) If the openings are valid, then decryption will return an error-corrected version of the message vector from $\mathbf{c}$. (Hence, the scheme itself actually helps an adversary that can prove that it is taking part in an SO attack.)

This immediately gives rise to a weak IND-SO-CCA attack: a suitable adversary A essentially only has to relay between its decryption oracle and the SO experiment to obtain the decryption of all challenge ciphertexts. The message distribution considered in the attack will only select codewords, so that the mentioned error correction will not disturb the decryption. Moreover, the underlying code has the property that a codeword is not fixed by the openings that occur during the attack. (Hence, a re-sampling will lead to a different message vector and can thus be detected.)

It is more challenging to prove that our modification does not harm the scheme's IND-CCA security. Intuitively, an IND-CCA adversary B could try to embed its own (IND-CCA) challenge c^* into a ciphertext vector $\mathbf{c}$ and obtain the decryption of c^* through a suitable $(\mathtt{soa}, Z)$ query. (With a little luck, B will not have to open c^*, so decryption will return the full message vector, including the decryption of c^*.)

To cope with such an IND-CCA adversary B, we will answer $(\mathtt{soa}, Z)$ only with the error-corrected message vector. Decryption will ensure (by the random choice of $\mathcal{I}$ and by ensuring suitably valid openings) that most of the encrypted messages m_i *and all opened messages* are consistent with a unique single codeword. (If this is not the case, then the query is rejected. Of course, we will have to make sure that B also learns nothing from the fact that the query was rejected.) Decryption then returns this unique codeword, and not simply the decryption of all individual ciphertexts. This procedure makes sure that a single ciphertext c^* embedded into $\mathbf{c}$ alone has no significant influence on the returned value.

Our strategy is somewhat reminiscent of the strategy of Bellare et al. [5], who show a black-box impossibility for IND-SO secure commitments. Our approach can be seen as a refinement and adaptation of their ideas to the PKE setting and to the standard model.

Note that our attack only uses two decryption queries; furthermore, one of these queries can be substituted by a random oracle query when adapting the scheme to the random oracle model. Thus, our scheme also gives rise to a separation between IND and weak IND-SO security in a bounded CCA setting [10]. Moreover, since CCA settings with only 1 decryption query and non-malleability are tightly related [6], our counterexample has also implications for non-malleability notions of security. (See Appendix D for details.)

Details on our generic group IND-CPA/IND-SO-CPA equivalence. Our equivalence result applies to a broad class of encryption schemes over prime order groups for which public keys as well as ciphertexts can be described by (low-degree) polynomials "in the exponent". We model the underlying group as generic (following Shoup's formalization) with respect to the IND-SO-CPA adversary and the adversarial message sampling algorithm. That means that the only basic group operations such algorithms may perform are equality testing, application of the group law, and computation of inverse elements. However, note that this is already sufficient, e.g., for realizing our efficient IND-SO-CCA adversary (see also Appendix C). A potential hash function utilized by the encryption scheme is modeled as a Random Oracle. Although the model we consider for our equivalence result may appear highly idealized, proving the equivalence is anything but trivial. There are several novel and challenging aspects about this proof; we only highlight a few here.

The common strategy of a proof in the generic group model is to show that, with overwhelming probability, an adversary does not obtain any information about the underlying secrets (e.g., secret keys, the challenge bit in indistinguishability games, etc.) of the considered game (IND-SO-CPA in our case). Thus, it can only win by mere guessing. To this end, one shows by means of a simulation game (where all secrets are replaced by formal variables) that a generic algorithm may only obtain information about secrets from nontrivial equations that hold between low-degree combinations of these secrets. (An equation is called trivial if it holds for all possible choices of the secrets.) If the secret values are chosen uniformly at random then by applying standard techniques (e.g., the Schwartz-Zippel Lemma in the case of prime power order groups) one can see that such equations may occur only very rarely. However, in our setting also the adversarial messages, which are chosen according to an arbitrary (efficiently re-samplable) distribution, belong to the secrets for which we want to argue that they are hidden information-theoretically. Moreover, in the opening phase, parts of the secrets are even disclosed to the adversary. We cope with these issues by modifying the way we usually simulate in the generic model and, hence, how non-trivial equations are defined. In particular, we need to adapt the simulation when the opening phase starts and show that a non-trivial equation and "bad" message distribution can be leveraged to win the IND-CPA game.

In a nutshell, our proof is split into two parts: First, we show that in order to win the IND-CPA game, it suffices that for all possible public keys and encryptions of a message vector, we can efficiently compute a non-trivial representation of the neutral group element in terms of the public key and (at least one of) the corresponding ciphertexts. The idea is to replace one of the messages with a different one for which this equation does not hold anymore and use the two messages in the IND-CPA game. Second, we show that from any generic IND-SO-CPA adversary, such a representation and message vector can be extracted.

Outline. After recalling some definitions in Section 2, we describe our separation in Section 3. The generic equivalence in the passive case can be found in Section 4. Sections A, B, and C discuss the restrictions we make for the CPA case, and in particular show that our separating scheme from the CCA case (and its analysis) is generic in our sense. Finally, Appendix D briefly describes extensions to our CCA separation.

2 Preliminaries

Notation. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. Throughout the paper, $k \in \mathbb{N}$ denotes the security parameter. For a finite set $\mathcal{S}$, we denote by $s \leftarrow \mathcal{S}$ the process of sampling s uniformly from $\mathcal{S}$. For a probabilistic algorithm A, we denote with $\mathcal{R}_A$ the space of A's random coins. $y \leftarrow A(x; R)$ denotes the process of running A on input x and with randomness $R \leftarrow \mathcal{R}_A$, and assigning y the result. We write $y \leftarrow A(x)$ for $y \leftarrow A(x; R)$ with uniform R. If A's running time is polynomial in k, then A is called probabilistic polynomial-time (PPT).

PRFs. A pseudorandom function (PRF) is a function $\mathsf{PRF} : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ for finite $\mathcal{K}, \mathcal{R}$, such that oracle access to $\mathsf{PRF}_K(\cdot)$ (for $K \leftarrow \mathcal{K}$) is indistinguishable from oracle access to a truly random function $RF : \mathcal{D} \to \mathcal{R}$. Concretely, for a distinguisher D, let $\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF},D}(k) := \Pr\left[D^{\mathsf{PRF}_K(\cdot)} = 1\right] - \Pr\left[D^{RF(\cdot)} = 1\right]$. We require that $\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF},D}$ is negligible for all PPT D.

PKE schemes. A public-key encryption (PKE) scheme PKE with message space $\mathcal{M}$ consists of three PPT algorithms $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}$. Key generation $\mathsf{Gen}(1^k)$ outputs a public key pk and a secret key sk. Encryption $\mathsf{Enc}(pk, m)$ takes pk and a message $m \in \mathcal{M}$, and outputs a ciphertext c. Decryption $\mathsf{Dec}(sk, c)$ takes sk and a ciphertext c, and outputs a message m. For correctness, we want $\mathsf{Dec}(sk, c) = m$ for all $m \in \mathcal{M}$, all $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$, and all $c \leftarrow \mathsf{Enc}(pk, m)$.

Standard security notions. Let PKE be a PKE scheme as above. For an adversary A, consider the following experiment: first, the experiment samples $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$ and runs A on input pk. Once A

outputs two messages m_0, m_1, the experiment flips a coin $b \leftarrow \{0,1\}$ and runs A on input $c^* \leftarrow \mathsf{Enc}(pk, m_b)$. We say that A wins the experiment iff $b' = b$ for A's final output b'. We denote A's advantage with $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{PKE},A}(k) := \Pr[A \text{ wins}] - 1/2$ and say that PKE is IND-CPA secure iff $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{PKE},A}$ is negligible for all PPT A. Similarly, write $\mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE},A}(k) := \Pr[A \text{ wins}] - 1/2$ for A's winning probability when A additionally gets access to a decryption oracle $\mathsf{Dec}(sk, \cdot)$ at all times. (To avoid trivialities, A may not query Dec on c^*, though.) PKE is IND-CCA secure iff $\mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE},A}$ is negligible for all PPT A.

Security under selective openings. Following [5, 16, 8], we present an indistinguishability-based definition for security under selective openings that captures security of an encryption scheme under adaptive attacks.

Intuitively, an adversary A that receives a vector of ciphertexts, along with openings of a subset of these ciphertexts, should not be able to distinguish the messages in the unopened ciphertexts from independently selected messages. The encrypted message vector is selected according to a (joint) message distribution selected by A. A also selects the set of ciphertexts to be opened, in a way possibly depending on the ciphertexts themselves. Since we currently do not know how to achieve this security notion for arbitrary (efficiently samplable) message distributions, we further restrict to efficiently *re*-samplable message distributions:

```
Experiment $\mathsf{Exp}^{\mathsf{weak\text{-}ind\text{-}so\text{-}cpa}}_{\mathsf{PKE},A}$
  $b \leftarrow \{0,1\}$
  $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
  $\mathsf{samp}(\cdot) \leftarrow A(pk)$
  $\mathbf{m}_0 := (m_i)_{i\in[n]} \leftarrow \mathsf{samp}()$
  $\mathbf{R} := (R_i)_{i\in[n]} \leftarrow (\mathcal{R}_{\mathsf{Enc}})^n$
  $\mathbf{c} := (c_i)_{i\in[n]} := (\mathsf{Enc}(pk, m_i; R_i))_{i\in[n]}$
  $\mathcal{I} \leftarrow A(\mathtt{sel}, \mathbf{c})$
  $\mathbf{m}_1 \leftarrow \mathsf{samp}(\mathbf{m}_{\mathcal{I}})$
  $out_A \leftarrow A(\mathtt{out}, (R_i)_{i\in\mathcal{I}}, \mathbf{m}_b)$
  return 1 if $out_A = b$, and 0 otherwise
```

Figure 2: Weak IND-SO-CPA experiment.

Definition 2.1 (Efficiently re-samplable)**.** *Let $n = n(k) > 0$, and let $\mathcal{D}$ be a joint distribution over $\mathcal{M}^n$. We say that $\mathcal{D}$ is* efficiently re-samplable *if there is a PPT algorithm* samp *such that for any $\mathcal{I} \subseteq [n]$ and any partial vector $\mathbf{m}'_{\mathcal{I}} := (m'_i)_{i\in\mathcal{I}} \in \mathcal{M}^{|\mathcal{I}|}$, $\mathsf{samp}(\mathbf{m}'_{\mathcal{I}})$ samples from $\mathcal{D} \mid \mathbf{m}_{\mathcal{I}}$, i.e., from the distribution $\mathcal{D}$, conditioned on $m_i = m'_i$ for all $i \in \mathcal{I}$. Note that in particular, $\mathsf{samp}()$ samples from $\mathcal{D}$.*

Definition 2.2 (Weak indistinguishability-based selective opening security)**.** *For a PKE scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, a polynomially bounded function $n = n(k) > 0$, and a stateful PPT adversary A, consider the experiment in Figure 2. We only allow A that always output re-sampling algorithms as in Definition 2.1. We call* PKE weakly IND-SO-CPA *secure if*

$$\mathsf{Adv}^{\mathsf{ind\text{-}so\text{-}cpa}}_{\mathsf{PKE},A}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{weak\text{-}ind\text{-}so\text{-}cpa}}_{\mathsf{PKE},A}(k) = 1\right] - \frac{1}{2}$$

is negligible for all PPT A. Similarly, we define an experiment $\mathsf{Exp}^{\mathsf{weak\text{-}ind\text{-}so\text{-}cca}}_{\mathsf{PKE},A}$ (with advantage $\mathsf{Adv}^{\mathsf{ind\text{-}so\text{-}cca}}_{\mathsf{PKE},A}$) that is identical to $\mathsf{Exp}^{\mathsf{weak\text{-}ind\text{-}so\text{-}cpa}}_{\mathsf{PKE},A}$, except that A gets access to a decryption oracle $\mathsf{Dec}(sk, \cdot)$ at all times. To avoid trivialities, we only allow A that never query their decryption oracle with any ciphertext from $\mathbf{c}$. We say that PKE *is* weakly IND-SO-CCA *secure if $\mathsf{Adv}^{\mathsf{ind\text{-}so\text{-}cca}}_{\mathsf{PKE},A}(k)$ is negligible.*

There are some minor technical differences between Definition 2.2 and the IND-SO-ENC definition from [5]: IND-SO-ENC security universally quantifies over all (efficiently re-samplable) message distributions. We let A choose samp instead, e.g., to allow a message distribution that depends on the public key pk. (In fact, otherwise it is not even clear that the resulting definition implies IND-CPA security.) Besides, unlike Böhl et al. [8], we model only one round of openings for simplicity. (However, our results hold also for multiple rounds of openings.)

3 Our Separating Encryption Scheme

In this section, we describe a PKE scheme that is IND-CCA secure, but not weakly IND-SO-CCA secure. So our scheme separates standard security from even the weakest considered form of selective opening security.

3.1 The Scheme

Specific notation and assumptions. In the following, let $\mathbb{F} = \mathbb{Z}_p$ be the finite field of size p for a prime p. (We will later choose a $(k+1)$-bit p as part of the public key of our scheme.) By $\mathsf{ipol}((X_i, Y_i)_{i=0}^{d})$ (for pairwise different X_i), we denote the unique degree-$\leq d$ polynomial $F \in \mathbb{F}[X]$ with $F(X_i) = Y_i$ for all i. Let $\mathcal{S}_\ell^S$ denote the set of all ℓ-sized subsets of S. We will assume a PRF $\mathsf{PRF} : \{0,1\}^k \times \{0,1\}^* \to \mathcal{S}_k^{[3k]}$ (such that oracle access to $\mathsf{PRF}_K(\cdot)$ for uniform $K \in \{0,1\}^k$ cannot be distinguished from access to a truly random function that maps arbitrary bitstrings to uniform k-sized subsets of $[3k]$). We will also assume an IND-CCA secure PKE scheme $\mathsf{PKE}' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ with message space $\mathbb{F}$. (The requirement about the message space is without loss of generality [19]; see also Appendix B for a scheme with a group as message space.)

Construction. $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is constructed from PKE':

Key generation adds a PRF key to sk: $\mathsf{Gen}(1^k)$ outputs $(pk, sk) = ((pk', p), (sk', K))$ for $(pk', sk') \leftarrow \mathsf{Gen}'(1^k)$, a uniformly chosen $(k+1)$-bit prime p, and $K \leftarrow \{0,1\}^k$.

Encryption marks ciphertexts as "regular": $\mathsf{Enc}(pk, m)$ runs $c' \leftarrow \mathsf{Enc}'(pk', m)$ and outputs $c = (\mathtt{reg}, c')$.

Decryption decrypts "regular" ciphertexts as PKE', but also offers possibilities to evaluate PRF_K and perform a special type of attack by decrypting "non-regular" ciphertexts:

$$\mathsf{Dec}(sk, c) = \begin{cases} \mathsf{Dec}'(sk', c') & \text{if } c = (\mathtt{reg}, c') \text{ for some } c', \\ \mathsf{PRF}_K(Z) & \text{if } c = (\mathtt{sel}, Z) \text{ for some } Z, \\ \mathsf{SOA}(sk, Z) & \text{if } c = (\mathtt{soa}, Z) \text{ for some } Z, \\ \perp & \text{else.} \end{cases}$$

Here, the function $\mathsf{SOA}(sk, Z)$ operates as follows:

1. Parse Z as $Z = ((c'_i)_{i \in [3k]}, (m_i, R_i)_{i \in \mathcal{I}})$, where $\mathcal{I} = \mathsf{PRF}_K((c'_i)_{i \in [3k]})$.
2. If there are indices $i \neq j$ with $c'_i = c'_j$, then return $\perp$.
3. If there is an $i \in \mathcal{I}$ with $\mathsf{Enc}'(pk', m_i; R_i) \neq c'_i$, then return $\perp$.
4. Decrypt the unopened ciphertexts by $m_i = \mathsf{Dec}'(sk', c'_i)$ for $i \in [3k] \setminus \mathcal{I}$.
5. First, determine if there is a degree-$\leq k$ polynomial $F \in \mathbb{F}[X]$ with $F = \mathsf{ipol}((i, m_i)_{i \in \mathcal{I} \cup \{j\}})$ for more than k values $j \in [3k] \setminus \mathcal{I}$. Note that there are only $2k$ candidates $F_\ell = \mathsf{ipol}((i, m_i)_{i \in \mathcal{I} \cup \{\ell\}})$ (for $\ell \in [3k] \setminus \mathcal{I}$) for F; hence, if such an F exists, it can be found efficiently (and in fact is unique). Return F, or $\perp$ if no such F exists.

Intuitively, $\mathsf{SOA}(sk, Z)$ returns a polynomial F that is consistent with *all* opened values m_i (for $i \in \mathcal{I}$), and *most* unopened values m_i (for $i \in [3k] \setminus \mathcal{I}$). (This slight distinction will be crucial to ensure that access to SOA does not enable IND-CCA attacks.)

Rationale and intuition for security analysis. The rationale of our modifications to PKE' is to enable a specific attack that only a weak IND-SO-CCA adversary is able to perform. Concretely, once an adversary supplies $3k$ ciphertexts along with openings of k of them (in a suitable $\mathsf{Dec}(sk, (\mathtt{soa}, Z))$ query), the scheme itself helps to decrypt all ciphertexts. Indeed, PKE is weakly IND-SO-CCA insecure with respect to the message distribution $\mathcal{D} = (F(i))_{i \in [3k]}$ with a uniform degree-$\leq k$ polynomial F: by relaying between the experiment and its Dec oracle, an adversary can obtain *all* (i.e., even unopened) challenge messages.

The difficult part will be to prove that our modification preserves the IND-CCA security of PKE'. That is, we will have to prove that $(\mathtt{sel}, Z)$ and $(\mathtt{soa}, Z)$ decryption queries do not help an IND-CCA adversary

A on PKE. For $(\mathtt{sel}, Z)$ queries, this is intuitively clear, as they are answered independently of the "actual" secret key sk'. For $(\mathtt{soa}, Z)$ queries, we will argue that the answer can already be deduced by "regular" decryption queries $(\mathtt{reg}, c')$. Concretely, if the PKE' ciphertext c^* from A's own challenge $(\mathtt{reg}, c^*)$ does not appear as ciphertext in Z, A can itself use Dec queries to emulate $\mathsf{SOA}(sk, Z)$. And even if Z contains c^*, A can still use Dec to decrypt all ciphertexts in Z except for c^*. We will show that $\mathsf{SOA}(sk, Z)$ can be reasonably well approximated when knowing all plaintexts encrypted in Z except for at most one. Namely, in order not to be rejected by $\mathsf{SOA}(sk, Z)$, almost all of the ciphertexts in Z must already decrypt to a value $F(i)$ that is consistent with one F. Knowing all but one plaintext allows a simulation to compute this F, and thus $\mathsf{SOA}(sk, Z)$'s answer.

Variations. Appendix D gives variations for bounded CCA security and non-malleability.

3.2 Why PKE is not weakly IND-SO-CCA secure

We now formally show that PKE allows for a simple weak IND-SO-CCA attack.

Theorem 3.1. *The PKE scheme* PKE *from Section 3.1 is not weakly IND-SO-CCA secure.*

Proof. We construct a weak IND-SO-CCA adversary A on PKE. On input pk, A outputs the $3k$-message distribution

$$\mathcal{D} = \big\{(F(1), \ldots, F(3k)) \,\big|\, F \in \mathbb{F}[X] \text{ uniformly chosen degree-} \leq k \text{ polynomial}\big\}$$

along with a suitable (re-)sampling algorithm samp. (For instance, samp can randomly extend its input $(F(i))_{i \in \mathcal{I}}$ to $k+1$ evaluation points as necessary and then use polynomial interpolation to retrieve F and thus all $F(i)$.) Note that k messages $m_i = F(i)$ from a $\mathcal{D}$-sample do not fully determine F and thus the whole message vector.

Once A receives a ciphertext vector $\mathbf{c} := (\mathtt{reg}, c'_i)_{i \in [3k]}$, it queries its decryption oracle on $(\mathtt{sel}, (c'_i)_{i \in [3k]})$ to receive a k-sized subset $\mathcal{I} \subset [3k]$. This $\mathcal{I}$ is the subset that A submits to its weak IND-SO-CCA experiment. Let $\mathsf{bad}_{\mathsf{coll}}$ be the event that $c'_i = c'_j$ for some $i \neq j$. By the correctness of PKE', this can only happen if $m_i = m_j$ for these i, j. By definition of $\mathcal{D}$, we have $\Pr[m_i = m_j] = 1/|\mathbb{F}| < 1/2^k$ for any fixed i, j. Hence, a union bound over all i, j shows that $\Pr[\mathsf{bad}_{\mathsf{coll}}] < \frac{3k(3k-1)}{2} \cdot \frac{1}{2^k} < \frac{5k^2}{2^k}$. We will thus assume $\neg\mathsf{bad}_{\mathsf{coll}}$ hereafter.

Upon receiving openings $(m_i, R_i)_{i \in \mathcal{I}}$ and a message vector $\mathbf{m}^* = (m^*_i)_{i \in [3k]}$, A queries its decryption oracle on $(\mathtt{soa}, ((c'_i)_{i \in [3k]}, (m_i, R_i)_{i \in \mathcal{I}}))$. By definition of Dec (and the function SOA), A will thus receive a polynomial F with $m_i = F(i)$ for all $i \in [3k]$ and can thus obtain the actually encrypted messages m_i. Finally, A will output $out_A = 0$ iff $m^*_i = F(i)$ for all $i \in [3k]$.

Still assuming $\neg\mathsf{bad}_{\mathsf{coll}}$, it is clear that A will output $out_A = 0$ when $b = 0$, i.e., when $\mathbf{m}^* = \mathbf{m}_0$. On the other hand, if $b = 1$, then $\mathbf{m}^* = \mathbf{m}_1$ has been re-sampled subject to $m^*_i = m_i$ for all $i \in \mathcal{I}$. However, since a message vector $\mathbf{m}$ from $\mathcal{D}$ is not fixed by only $k = |\mathcal{I}|$ values m_i, we have that $\mathbf{m}^* \neq \mathbf{m}_0$ (so that A outputs $out_A = 1$) except with probability at most $1/|\mathbb{F}| < 1/2^k$. Summarizing, we get

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ind\text{-}so\text{-}cca}}_{\mathsf{PKE},A}(k) \;&=\; \Pr[out_A = b] - \frac{1}{2} \;\geq\; \Pr[out_A = b \mid \neg\mathsf{bad}_{\mathsf{coll}}] - \Pr[\mathsf{bad}_{\mathsf{coll}}] - \frac{1}{2} \\ &>\; \frac{1}{2}\left(1 + (1 - \frac{1}{2^k})\right) - \frac{5k^2}{2^k} - \frac{1}{2} \;=\; \frac{1}{2} - \frac{5k^2+2}{2^k},\end{aligned}$$

which is non-negligible (and in fact negligibly close to the maximal advantage). □

3.3 Why PKE is still IND-CCA secure

We show that PKE inherits PKE''s IND-CCA security.

Theorem 3.2. *The PKE scheme* PKE *from Section 3.1 is IND-CCA secure, assuming that* PKE' *is IND-CCA secure, and* PRF *is pseudorandom.*

Proof. Let A be a PPT adversary on PKE that makes exactly q decryption queries. We proceed in games, and let out_i denote the output of Game i.

Game 1 is the original IND-CCA game with A. Consequently,

$$\Pr[out_1 = 1] - 1/2 \;=\; \mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE},A}(k).$$

In **Game** 2, we answer decryption queries of the form $(\mathtt{sel}, Z)$ with $RF(Z)$ instead of $\mathsf{PRF}_K(Z)$ for a truly random function $RF : \{0,1\}^* \to \mathcal{S}_k^{[3k]}$. (We will assume that RF is efficiently implemented, e.g., using lazy sampling.) A straightforward reduction to PRF's pseudorandomness yields

$$\Pr[out_1 = 1] - \Pr[out_2 = 1] \;=\; \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF},D}(k)$$

for a suitable PRF distinguisher D.

In **Game** 3, we slightly change the way decryption queries of the form $(\mathtt{soa}, Z)$ are answered. Our goal is to avoid a decryption of c^*, where $(\mathtt{reg}, c^*)$ is A's own challenge ciphertext. Informally, we simply skip decrypting c'_i if $c'_i = c^*$ in Step 4 of the function $\mathsf{SOA}(sk, Z)$. In Step 5, we skip any comparison of m_i for $c'_i = c^*$. Formally, we change Steps 4 and 5 into

4. Let $\mathcal{I}^*$ be the set of all $i \in [3k] \backslash \mathcal{I}$ with $c'_i = c^*$. (Note that $|\mathcal{I}^*| \leq 1$.) Decrypt the unopened ciphertexts not equal to c^* by $m_i = \mathsf{Dec}'(sk', c'_i)$ for $i \in [3k] \setminus (\mathcal{I} \cup \mathcal{I}^*)$.
5. If there is a degree-$\leq k$ polynomial $F \in \mathbb{F}[X]$ with $F = \mathsf{ipol}((i, m_i)_{i \in \mathcal{I} \cup \{j\}})$ for more than k values $j \in [3k] \setminus (\mathcal{I} \cup \mathcal{I}^*)$, then return F. Else return $\perp$.

This modified version SOA' only yields different values from that of Game 2 if

(a) $Z = ((c'_i)_{i \in [3k]}, (m_i, R_i)_{i \in \mathcal{I}})$ with pairwise different c'_i and $\mathcal{I} = RF((c'_i)_{i \in [3k]})$,

(b) all openings are valid in the sense $\mathsf{Enc}'(m_i; R_i) = c'_i$ for $i \in \mathcal{I}$, and

(c) there are *exactly* $k + 1$ indices $i \in [3k] \setminus \mathcal{I}$ with $m_i = F(i)$ for a degree-$\leq k$ polynomial F.

In this case, $\mathsf{SOA}(sk, Z)$ will return F, while $\mathsf{SOA}'(sk, Z)$ might return $\perp$ (in case there is an unopened $c'_i = c^*$ with $m_i = F(i)$). Let us call a query $(\mathtt{soa}, Z)$ satisfying (a)-(c) *implausible.* Denote by $\mathsf{bad}_{\mathsf{impl}}$ the event that A ever submits an implausible decryption query. Unless $\mathsf{bad}_{\mathsf{impl}}$ occurs, Game 2 and Game 3 are identical, so that $\Pr[\mathsf{bad}_{\mathsf{impl}}]$ is the same in these games.

Intuitively, $\mathsf{bad}_{\mathsf{impl}}$ is unlikely, because it necessitates that the (randomly chosen) subset $\mathcal{I} = RF((c'_i)_{i \in [3k]})$ happens to contain only indices i with $m_i = F(i)$ for the uniquely determined polynomial F. However, requirement (c) states that $k - 1$ indices i are *not* compatible with F, meaning $m_i \neq F(i)$. The probability that any such i is contained in $\mathcal{I}$ is overwhelming. We prove the following lemma after the main proof.

Lemma 3.3. $\Pr[\mathsf{bad}_{\mathsf{impl}}] \leq q \cdot \left(\frac{5}{6}\right)^k$ *for* $k \geq 2$.

Using Lemma 3.3, we thus get for $k \geq 2$:

$$|\Pr[out_3 = 1] - \Pr[out_2 = 1]| \leq \Pr[\mathsf{bad}_{\mathsf{impl}}] \leq q \cdot \left(\frac{5}{6}\right)^k.$$

Finally, we have

$$\Pr[out_3 = 1] \;=\; \mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE},B} + 1/2 \tag{1}$$

for a suitable IND-CCA adversary B on PKE'. Concretely, observe that the whole Game 3 only uses sk' to decrypt PKE' ciphertexts different from c^*. Hence, B can simulate A, using its own challenge public key pk' and ciphertext c^* as A's public key and challenge. A's choice of challenge messages m_0, m_1 is also used by B. A's decryption queries $(\mathtt{reg}, c')$ are relayed (as c') to B's decryption oracle; $(\mathtt{sel}, Z)$ and $(\mathtt{soa}, Z)$ queries are answered by B for A, using B's own decryption oracle as necessary for $(\mathtt{soa}, Z)$ queries. (Note that B's challenge c^* will never have to be decrypted by our change from Game 3.) This adversary B thus perfectly simulates Game 3 for A, so we get (1).

Taking things together yields

$$\left|\mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE},A} - \mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE},B}\right| \;=\; \left|\Pr\left[out_1\right] - \Pr\left[out_3\right]\right| \;\leq\; \left|\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF},D}(k)\right| + q \cdot \left(\frac{5}{6}\right)^k$$

for $k \geq 2$, which shows the theorem. □

It remains to prove Lemma 3.3:

Proof of Lemma 3.3. Given a ciphertext vector $\mathbf{c} = (c'_i)_{i\in[3k]}$, define $m_i = \mathsf{Dec}(sk', c'_i)$ for all i. (Correctness implies that these m_i are the same that will be recovered by SOA, either using openings given by A, or by decryption.) Say that there is a (unique) degree-$\leq k$ polynomial F and a $(2k+1)$-sized subset $\overline{\mathcal{I}} \subset [3k]$ with $m_i = F(i) \Leftrightarrow i \in \overline{\mathcal{I}}$. (Note that this is a prerequisite for $\mathsf{bad}_{\mathsf{impl}}$.)

The crucial observation is that the set $\mathcal{I} = RF(\mathbf{c})$ that determines which ciphertexts A must open is chosen independently and uniformly from the set of all k-sized subsets of $[3n]$. Furthermore, $\mathcal{I}$ is only chosen once A makes a $(\mathtt{sel}, Z)$ or $(\mathtt{soa}, Z)$ query that involves $\mathbf{c}$. If $\mathcal{I} \not\subset \overline{\mathcal{I}}$, then there can be no implausible query with this $\mathbf{c}$. (Condition (b) would require that some $i^* \notin \overline{\mathcal{I}}$ is opened, so that Condition (c) cannot be met, as $m_{i^*} \neq F(i^*)$.) Hence $\mathcal{I} \subset \overline{\mathcal{I}}$ is a necessary requirement for an implausible query with this $\mathbf{c}$. But $\mathcal{I} \subset \overline{\mathcal{I}}$ means that a random k-sized subset $\mathcal{I}$ of $[3k]$ is a subset of a fixed $(2k+1)$-sized subset $\overline{\mathcal{I}} \subset [3k]$. Hence,

$$\Pr\left[\mathcal{I} \subset \overline{\mathcal{I}}\right] \;=\; \frac{\binom{2k+1}{k}}{\binom{3k}{k}} \;=\; \frac{(2k+1)!(2k)!}{(3k)!(k+1)!} \;=\; \frac{(2k+1)\cdots(k+2)}{(3k)\cdots(2k+1)} \;\overset{k\geq 2}{\leq}\; \left(\frac{5}{6}\right)^k .$$

Since A makes only q decryption queries, it can only submit at most q different $\mathbf{c}$. For each $\mathbf{c}$, the probability is at most $(5/6)^k$ that an implausible query with this $\mathbf{c}$ exists. Hence, a union bound shows that

$$\Pr\left[\mathsf{bad}_{\mathsf{impl}}\right] \;\leq\; q \cdot \left(\frac{5}{6}\right)^k .$$

□

4 Equivalence of IND-SO-CPA and IND-CPA in the GGM

We give evidence towards the equivalence of IND-SO-CPA and IND-CPA by showing that for a broad class of encryption schemes any efficient generic IND-SO-CPA adversary can be turned into an efficient IND-CPA adversary.

In the following, some additional notation is needed: For a vector of variables $\mathbf{X}$ or polynomials $\mathbf{P}$, let $|\mathbf{X}|$ and $|\mathbf{P}|$ denote the size of the corresponding vector. For a polynomial P, let $|P|$ denote the number of non-zero monomials.

4.1 The Class of $(\mathbf{P}, \mathbf{E}, \mathbf{H}, \mathcal{H})$-CS-type Encryption Schemes

The following definition covers a broad class of public-key encryption schemes over prime order groups where messages are group elements. This includes ElGamal, Cramer-Shoup (CS), and also a slight variation of the separating scheme from Section 3.1, e.g., instantiated with Cramer-Shoup (see Appendix B). Note that the restrictions on the polynomials in Definition 4.1 are reasonable for meaningful encryption (see Appendix A).

Definition 4.1. *Let G be a group of prime order p with generator g and $\mathbb{F} = \mathbb{Z}_p$. Furthermore, let u_1, u_2, u_3, v_1, $v_2 \in \mathbb{N}$,*

- $\mathbf{P} = (P_1 = 1, P_2, \ldots, P_{u_1})$ *be public key polynomials in* $\mathbb{F}[X_1, \ldots, X_{v_1}]$,

- $\mathbf{E} = (E_1, \ldots, E_{u_2})$ *be polynomials in* $\mathbb{F}[X_1, \ldots, X_{v_1}, Y_1, \ldots, Y_{v_2}, Z, M]$, *called encryption polynomials, where all monomials have the form*
$$\alpha P^{e_1} Z^{e_2} M^{e_3} \prod_{i=1}^{v_2} Y_i^{d_i}$$
with $P \in \mathbf{P}$, $e_1, e_3 \in \{0, 1\}$, $e_1 + e_3 \leq 1$, *and* $e_2, d_i \in \mathbb{N}_0$,
- $\mathbf{H} = (H_1, \ldots, H_{u_3})$ *be tuple of hash input polynomials, where* $H_i \in \mathbf{E}$, $\deg_Z(H_i) = 0$, *and for at least one* H_i *it holds that* $\deg_M(H_i) > 0$ *or* $\max_j(\deg_{Y_j}(H_i)) > 0$,
- $\mathcal{H} : G^{u_3} \mapsto \mathbb{F}$ *be a hash function.*

Then we call an encryption scheme over G *a* $(\mathbf{P}, \mathbf{E}, \mathbf{H}, \mathcal{H})$-***CS-type encryption scheme*** *if the following conditions are satisfied:*

- *The public key is of the form* $(g^{P(\mathbf{x})})_{P \in \mathbf{P}}$, *where* $\mathbf{x} \leftarrow \mathbb{F}^{v_1}$.
- *The ciphertext of a message* $m = g^{m'}$ *is of the form* $\mathbf{c} = (g^{E(\mathbf{x}, \mathbf{y}, z, m')})_{E \in \mathbf{E}}$, *where* $\mathbf{y} \leftarrow \mathbb{F}^{v_2}$ *and* z *is the output of* $\mathcal{H}$ *given* $g^{H_1(\mathbf{x}, \mathbf{y}, m')}, \ldots, g^{H_{u_3}(\mathbf{x}, \mathbf{y}, m')}$.

Example 1. Cramer-Shoup encryption scheme can be viewed as $(\mathbf{P}, \mathbf{E}, \mathbf{H}, \mathcal{H})$-CS-type encryption scheme, where we assume that generator g from Definition 4.1 has been chosen randomly and

- $P_1 = 1$, $P_2 = X_1$, $P_3 = X_2 + X_1 X_3$, $P_4 = X_4 + X_1 X_5$, $P_5 = X_6$
- $E_1 = P_1 Y_1$, $E_2 = P_2 Y_1$, $E_3 = P_3 Y_1 + P_4 Y_1 Z$, $E_4 = P_5 Y_1 + M$
- $\mathcal{H} : G^3 \mapsto \mathbb{F}$ is a collision resistant hash function computed over group elements with exponents of the form $H_1 = E_1$, $H_2 = E_2$, $H_3 = E_4$.

4.2 IND-SO-CPA in the Generic Group Model

We base the formalization of the IND-SO-CPA game for generic adversaries on the generic group model (GGM) introduced by Shoup [23]. In Shoup's GGM elements are encoded as unique random bit strings, ensuring that no special property of a group's representation can be exploited. More precisely, let $\mathbb{E} \subset \{0, 1\}^{\lceil \log_2(p) \rceil}$, where $|\mathbb{E}| = p$, denote the set of possible element encodings of a cyclic group G of order p. Since any such group G is isomorphic to $(\mathbb{F}, +)$, we will always use $\mathbb{F}$ for the internal representation of G. A *generic group oracle* defines the random map between group elements and encodings and allows A to perform operations from $\Omega = \{+, -\}$ on encoded group elements. Equality testing can be done without the help of $\mathcal{O}$ since encodings are unique.

Internal state of $\mathcal{O}$. The oracle maintains two lists $\mathcal{L}$ and $\mathcal{E}$ which are used to define the random mapping between $\mathbb{F}$ and $\mathbb{E}$ in a lazy manner: $\mathcal{L} \subset \mathbb{F}$ will be initially populated with the elements comprising the public key of the considered encryption scheme. While A interacts with $\mathcal{O}$, additional elements are added to $\mathcal{L}$. The list $\mathcal{E} \subset \mathbb{E}$ contains the random encodings corresponding to the elements in $\mathcal{L}$. More precisely, the i-th encoding $\mathcal{E}_i$ represents the i-th element $\mathcal{L}_i$. We will denote the encoding of an element $a \in \mathcal{L}$ by $[\![a]\!]$.

Encoding of elements. Each time an element a should be added to $\mathcal{L}$, $\mathcal{O}$ checks if a is already contained in $\mathcal{L}$. If this is the case, $[\![a]\!]$ is already defined and will be appended to $\mathcal{E}$ again. Otherwise, a fresh encoding $\sigma \leftarrow \mathbb{E} \setminus \mathcal{E}$ is sampled and appended to $\mathcal{E}$. The encoding $[\![a]\!]$ is sent to A. We may assume that a generic algorithm A always outputs encodings it has previously received by the oracle: A fresh encoding not contained in $\mathcal{E}$ is associated with a random $a \in \mathbb{F} \setminus \mathcal{L}$. Assuming $|\mathcal{L}|$ is polynomial in $\log(p)$, such an element can be efficiently generated by A itself with overwhelming probability $1 - \frac{|\mathcal{L}|}{p}$ by sampling a random $a \in \mathbb{F}$. The corresponding encoding $[\![a]\!]$ can be computed from $[\![1]\!]$ using double-and-add. Similarly, in our upcoming IND-SO-CPA setting, A will be able to output an encoding that has been computed by another generic algorithm, but has not explicitly given to A, only with negligible probability of at most $\frac{|\mathcal{E}|}{p}$.

Query operations. A may ask $\mathcal{O}$ to perform an operation $\circ \in \Omega$ on encoded elements $[\![a]\!], [\![b]\!] \in \mathcal{E}$. Then $a \circ b$ is added to $\mathcal{L}$ and $[\![a \circ b]\!]$ is sent to A.

The GGM IND-SO-CPA game. The IND-SO-CPA game for generic adversaries against an $(\mathbf{P}, \mathbf{E}, \mathbf{H}, \mathcal{H})$-CS-type encryption scheme is shown in Figure 3a. By abuse of notation we assume that for a new input

$[\![a]\!]$ given to a generic algorithm (in the sense that $a \notin \mathcal{L}$), a is first added to $\mathcal{L}$ and then an encoding is determined. The hash function $\mathcal{H} : G^{u_4} \mapsto \mathbb{F}$ is modeled as a Random Oracle, which on input of u_4 concatenated encodings, outputs a fresh hash value $z \leftarrow \mathbb{F}$ if it has not received this input before. Otherwise, the hash value which has been chosen previously is returned. Furthermore, as can be seen from Figure 3a, both the adversary A and samp are modeled as generic algorithms. The algorithm samp is stateless but its output may depend on the public key $[\![P(\mathbf{x})]\!]_{P \in \mathbf{P}}$ since this was given as input to A before samp was created.

4.3 Equivalence for $(\mathbf{P}, \mathbf{E}, \mathbf{H}, \mathcal{H})$-CS-type Encryption

As a warm-up, consider the IND-SO-CPA game for ElGamal, viewed as a CS-type encryption scheme where $\mathbf{P} = (1, X)$ and $\mathbf{E} = (Y, XY + M)$, in the GGM. In this model, it is not hard to show (by means of a simulation game) that the only source of information for the adversary about the challenge bit b are non-trivial equations between elements that are linear combinations of the secret key x, the unopened random coins for encryption $(y_j)_{j \notin \mathcal{I}}$, and the unopened encrypted messages $(m'_{0,j})_{j \notin \mathcal{I}}$. More precisely, an adversary may only obtain information about b if the difference $\Delta(x, y_1, \ldots, y_n, m'_{0,1}, \ldots, m'_{0,n})$ of two computed elements is zero, where Δ is a non-zero polynomial of the form

$$\Delta = \alpha_0 + \alpha_1 X + \sum_{j=1}^{n} \beta_j Y_j + \sum_{j=1}^{n} \gamma_j (XY_j + M_j)$$

and $\beta_j = \gamma_j = 0$ for $j \in \mathcal{I}$ ($\mathcal{I} = \emptyset$ if we are not yet in the opening phase). What is the probability that this happens? Note that in contrast to the secret key and the random coins, the messages $m'_{0,j}$ are not necessarily uniformly chosen. So the well-known Schwartz-Zippel Lemma cannot immediately be applied. However, since samp is also assumed to be generic, $m'_{0,j}$ will be of the form $m'_{0,j} = R_j(x)$ for some polynomial R_j of the form $R_j = \alpha_0 + \alpha_1 X$. Let us consider the polynomial $\Delta' = \Delta(R_1, \ldots, R_n)$ which results from replacing any occurrence of M_j by R_j. It is easy to see that $\Delta' \neq 0$ if $\Delta \neq 0$. Finally, we can apply Schwartz-Zippel to Δ' to upper bound the probability that $\Delta(x, y_1, \ldots, y_n, m'_{0,1}, \ldots, m'_{0,n}) = \Delta'(x_1, y_1, \ldots, y_n)$ is zero, yielding the bound $\frac{2}{p}$.

Note that for more general public key and encryption polynomials as considered in Definition 4.1, Δ' is not guaranteed to be non-zero anymore: For instance, consider the slightly modified encryption polynomials $\mathbf{E} = (Y, XY + YM)$ and the difference polynomial $\Delta = -Y_1 + (XY_1 + Y_1 M_1)$. Here Δ' becomes zero for $R_1 = 1 - X$. Fortunately, it turns out that in this case the corresponding encryption scheme is already IND-CPA insecure. In our example, this is obvious: An IND-CPA adversary could choose $m_0 = g(g^x)^{-1}$ and a random message m_1 and check whether for the challenge ciphertext $c = (c_1, c_2)$ holds that $c_1^{-1} c_2 = g^{\Delta(X=x, Y_1=y_1, M_1=m'_b)}$ is equal to 1, where $m'_0 = 1 - x$ and $m'_1 = \log_g(m_1)$. With overwhelming probability this will not hold for $b = 1$.

More generally, we can show that any Δ and R_j's can be used to build an IND-CPA adversary that works similarly. This is done in the first part (Theorem 4.2) of our proof which is actually independent of the generic model. It essentially says that if for all possible public keys and all possible encryptions of certain messages, we can efficiently compute a non-trivial representation of $1 \in G$ in terms of the public key and the ciphertexts, then we can win the IND-CPA game with overwhelming probability. The idea is to replace one of the messages with a different one for which the equation does not hold anymore and use these two messages in the scope of the IND-CPA game. In the second part (Theorem 4.3), we show that any efficient generic adversary who wins the IND-SO-CPA game with non-negligible probability gives rise to such a representation (in form of a polynomial) and corresponding messages.

Theorem 4.2. *Let a $(\mathbf{P}, \mathbf{E}, \mathbf{H}, \mathcal{H})$-CS-type encryption scheme* PKE *over a group G of prime order p be given. Furthermore, let a polynomial Δ of the form*

$$\Delta = \sum_{P \in \mathbf{P}} \alpha_P P(\mathbf{X}) + \sum_{j=1}^{n} \sum_{E \in \mathbf{E}} \beta_{j,E} E(\mathbf{X}, \mathbf{Y}_j, Z_j, M_j)$$

and polynomials $R_1, \ldots, R_n$ of the form $R_i = \sum_{P \in \mathbf{P}} \alpha_P P(\mathbf{X})$ over $\mathbb{F}$ be given (where the coefficients α, β in the above representation are known) such that $\Delta \neq 0$ but $\Delta(M_1 = R_1, \ldots, M_n = R_n) = 0$. Then we can build a generic adversary B who wins the IND-CPA game for PKE*, modeling $\mathcal{H}$ as a Random Oracle, with probability at least $1 - \frac{\deg(\Delta)}{2p}$ using $O((\max_{E \in \mathbf{E}}(|E|)|\mathbf{E}||\mathbf{Y}| + |\mathbf{P}|)\log(p)n)$ multiplications over G and $\mathbb{F}$.*

Proof. First, observe that there is some $1 \leq i \leq n$ such that $\Delta(R_1, \ldots, R_{i-1}) \neq 0$ but $\Delta(R_1, \ldots, R_i) = 0$. Note that in this case, we know that $\deg_{M_j}(\Delta) = \deg_{Z_j}(\Delta) = \deg_Y(\Delta) = 0$, for all $Y \in \mathbf{Y}_j$ and $j > i$. Clearly, also for uniform $\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_i, z_1, \ldots, z_i$, it holds that $\Delta(\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_i, z_1, \ldots, z_i, m'_1, \ldots, m'_i) = 0$, where $m'_1 = R_1(\mathbf{x}), \ldots, m'_i = R_i(\mathbf{x})$. Furthermore, if we additionally choose $m''_i \in \mathbb{F}$ at random, the probability that

$$\Delta(\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_i, z_1, \ldots, z_i, m'_1, \ldots, m'_{i-1}, m''_i) = 0$$

is upper bounded by $\frac{\deg(\Delta)}{p}$. This follows from the Schwartz-Zippel Lemma observing that $\Delta(R_1, \ldots, R_{i-1})$ is a non-zero and of degree at most $\deg(\Delta)$.

Now, we are prepared to describe the IND-CPA adversary. First, B receives the public key $(g^{P(\mathbf{x}^*)})_{P \in \mathbf{P}}$ of the $(\mathbf{P}, \mathbf{E}, \mathbf{H}, \mathcal{H})$-CS-type encryption scheme from the challenger. Using this key it creates the message $g^{m_0^*} = g^{R_i(\mathbf{x}^*)} = \prod_{P \in \mathbf{P}} (g^{P(\mathbf{x}^*)})^{\alpha_P}$, where $R_i = \sum_{P \in \mathbf{P}} \alpha_P P(\mathbf{X})$, and $g^{m_1^*}$, where $m_1^* \leftarrow \mathbb{F}$. Then it sends them to the challenger who responds with the ciphertext

$$(g^{E(\mathbf{x}^*, \mathbf{y}^*, z^*, m_b^*)})_{E \in \mathbf{E}}\,, \tag{2}$$

where $b \leftarrow \{0, 1\}$ and z^* is the hash value associated with the message $g^{m_b^*}$. Since we consider the IND-CPA game in the Random Oracle Model z^* has been chosen uniformly at random from $\mathbb{F}$. Next, B creates the remaining values in order to evaluate Δ as exponent: It computes the messages $g^{m'_j} = g^{R_j(\mathbf{x}^*)}$ and chooses $\mathbf{y}_j \leftarrow \mathbb{F}^{v_2}$, $z_j \leftarrow \mathbb{F}$, for $j < i$. Finally, it computes

$$g^{\Delta(\mathbf{x}^*, \mathbf{y}_1, \ldots, \mathbf{y}_{i-1}, \mathbf{y}^*, z_1, \ldots, z_{i-1}, z^*, m'_1, \ldots, m'_{i-1}, m^*)}\,, \tag{3}$$

where it is easy to see that B is in fact able to evaluate this polynomial in the exponent (cf. paragraph on runtime). If the resulting element equals 1, B outputs $out_B = 0$ and otherwise 1.

As we know from the previous analysis, the element in Equation 3 happens to be 1 for both messages with probability at most $\frac{\deg(\Delta)}{p}$. So in this case B's guess is correct with probability $\frac{1}{2}$. If this failure does not happen B's guess is correct with probability 1. In total, we have a probability of at least $\frac{1}{2}\frac{\deg(\Delta)}{p} + 1 - \frac{\deg(\Delta)}{p}$.

Let us briefly consider the runtime of B. Note that elements involving $\mathbf{y}_i = \mathbf{y}^*$, and $m'_i = m^*$ or $z_i = z^*$, are given and do not need to be computed (cf. Equation 2). First, constructing the messages $g^{m'_j}$ requires $O(\log(p)|\mathbf{P}|n)$ group operations. To compute the group element from Equation 3, B uses the known representation in $\mathbf{P}$ and $\mathbf{E}$. For the first part $\prod_{P \in \mathbf{P}} (g^{P(\mathbf{x}^*)})^{\alpha_P}$ about $O(\log(p)|\mathbf{P}|)$ group operations are required. To compute the second at most $n - 1$ encryptions are needed. More precisely, the second part $\prod_{j=1}^{n} \prod_{E \in \mathbf{E}} g^{\beta_{j,E} E(\mathbf{x}, \mathbf{y}_j, z_j, m'_j)}$ can be computed (for $j \neq i$) as a multi-exponentiation (the exponents are the monomials of each E) with elements of the form

$$a^{\prod_{k=1}^{v_2} y_{j,k}^{d_{j,k}} z_j^{e_2} \beta_{j,E}}\,,$$

where $a = g^{(P(x^*))^{e_1}}$ or $a = g^{{m'_j}^{e_3}}$ which requires $O(\log(p)|\mathbf{Y}| \max_{E \in \mathbf{E}}(|E|)|\mathbf{E}|n)$ multiplications. □

Theorem 4.3 says that from any generic IND-SO-CPA adversary A certain polynomials as required for Theorem 4.2 can be extracted using "white-box access" to A. Here the extraction algorithm B does not only play the role of the IND-SO-CPA challenger and restricts itself to considering the in- and output of A (in this case we would be in the standard model) but closely observes the operations A performs on its inputs, i.e., B substitutes (and modifies) the generic oracle. More precisely, B's strategy is as follows: It turns the real IND-SO-CPA game in the generic model into a simulation game which does not reveal any information about the secret bit b chosen by the challenger. So A has no better chance than mere guessing to win the simulation game. Since the simulation game and the real game are equivalent unless a certain failure event occurs, an adversary who has a non-negligible advantage in winning the real game must cause this simulation failure with non-negligible probability. The crucial point is that a failure event is defined in a way such that it gives rise to the polynomials from Theorem 4.2.

Theorem 4.3. *Let a* $(\mathbf{P}, \mathbf{E}, \mathbf{H}, \mathcal{H})$*-CS-type encryption scheme* PKE *over a group* G *of prime order* p *be given. Furthermore, let* $d = \max_{S \in \mathbf{P} \cup \mathbf{E}}(\deg(S))$*,* $d' = \max_{S \in \mathbf{P}}(\deg(S))$*, and* $r = \max_{S \in \mathbf{P} \cup \mathbf{E}}(|S|)$*, and* $s = \max(|\mathbf{X}|, |\mathbf{Y}|)$*. Suppose there is a generic group adversary* A *that wins the IND-SO-CPA game for* PKE*, where we model* $\mathcal{H}$ *as Random Oracle, with advantage* $\mathsf{Adv}^{\text{ind-so-cpa}}_{\mathsf{PKE},A}$*, and by using* n *challenge messages. Let* $O(t)$ *and* $O(t')$ *denote the runtime of* A *and* samp*, respectively. Then there is a generic algorithm* B *which, by white-box access to* A*, extracts a polynomial* Δ *of degree at most* d *as well as polynomials* $R_1, \ldots, R_n$ *satisfying the conditions of Theorem 4.2 with a probability of at least* $\mathsf{Adv}^{\text{ind-so-cpa}}_{\mathsf{PKE},A} - \frac{dd'}{p}$ *and by performing at most* $O(r(|\mathbf{P}| + |\mathbf{E}|n)((t + t' + |\mathbf{P}| + |\mathbf{E}|n)^2 + \log(p)s))$ $\mathbb{F}$*-operations.*

Proof. **Game** 1 is the real IND-SO-CPA game as shown in Figure 3a.

Game 2 is the transition game shown in Figure 3b, which is actually equivalent to the real IND-SO-CPA game. Here, we define a new oracle $\mathcal{O}_1$ as follows: $\mathcal{O}_1$ uses polynomials to internally represent elements from $\mathbb{F}$. More precisely, we have $\mathcal{L} \subset \mathbb{F}[\mathbf{X}, \mathbf{Y}_1, \ldots, \mathbf{Y}_n, \mathbf{Z}, \mathbf{M}]$. Initially, the list is populated with the polynomials $\mathbf{P}$ describing the public key. Later, for each message, polynomials $\mathbf{E}$ describing its ciphertext are added. Applying the group operation in this polynomial representation translates to polynomial addition over $\mathbb{F}$. Moreover, the oracle receives certain elements $\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_n, \mathbf{z}, \mathbf{m}'_0$ which are used to evaluate the polynomials in order to determine encodings: Two elements $R_1, R_2 \in \mathcal{L}$ are assigned the same encoding if

$$((R_1 - R_2)(\mathbf{M} = \mathbf{m}'_0))(\mathbf{X} = \mathbf{x}, \mathbf{Y}_1 = \mathbf{y}_1, \ldots, \mathbf{Y}_n = \mathbf{y}_n, \mathbf{Z} = \mathbf{z}) \equiv 0 \bmod p\,. \tag{4}$$

Note that a message $m'_{0,j}$, $1 \leq j \leq n$, might be a non-constant polynomial of the form $\sum_{P \in \mathbf{P}} \alpha_P P(\mathbf{X})$ in which case we assume that it is also evaluated with $\mathbf{x}$. Now, each time a polynomial R_1 is added to $\mathcal{L}$, the list is searched for a polynomial R_2 satisfying Equation 4. If such an element is found, the corresponding encoding is returned, otherwise a fresh, unused encoding is sampled.

There is only a minor technical difference between the two oracles: $\mathcal{O}_0$ immediately evaluates polynomials and calculates with $\mathbb{F}$-elements, whereas $\mathcal{O}_1$ does the calculation with polynomials and delays the evaluation to the point when encodings are determined. However, this is equivalent and so A has the same success probability in the real and the transition game.

Game 3 is the simulation game, as shown in Figure 3c, in which the computation is independent of the bit b. More precisely, we make the computation independent of all (unopened) secrets and messages. Thus, A has no better chance than guessing b. The simulation game is equivalent to the transition game unless a simulation failure occurs yielding polynomials which can be used to build an IND-CPA adversary.

For the simulation game, we slightly modify $\mathcal{O}_1$ resulting in a oracle $\mathcal{O}_2$:

- During Initialization & Challenge, $\mathcal{O}_2$ assigns two elements $R_1, R_2 \in \mathcal{L}$ the same encoding if they are equal as polynomials over $\mathbb{F}$, i.e., $(R_1 - R_2) \equiv 0$.
- In the Opening Phase the oracle receives the choices $\{\mathbf{y}_i, z_i, m'_{0,i}\}_{i \in \mathcal{I}}$ revealed to A and assigns the same encoding if $(R_1 - R_2)(\mathbf{y}_i, z_i, m'_{0,i})_{i \in \mathcal{I}} \equiv 0$.

The reason why we need to simulate differently in the Opening Phase is that the adversary obtains additional information about part of the secrets. For instance, he now can compute the encryption of $m'_{0,i}$, for $i \in \mathcal{I}$, on his own. So we need to make sure that he receives the same encodings for the ciphertext that the oracle has assigned in the previous phase.

Now, the crucial observation is that in the simulation game given $[\![m'_{b,1}]\!], \ldots, [\![m'_{b,n}]\!]$ the only source of information about b would be encodings given to A in previous steps that depend on $m'_{0,i}$ or $m'_{1,i}$ for $i \notin \mathcal{I}$ since $m'_{0,i} = m'_{1,i}$ for $i \in \mathcal{I}$. However, encodings representing (combinations of) encryptions are independent of $m'_{0,i}$ (and $m'_{1,i}$) for $i \notin \mathcal{I}$, since we never evaluate the variables M_i. Hence, the probability that out_A equals b in the simulation game is $\frac{1}{2}$.

```
Initialization & Challenge
$b \leftarrow \{0,1\}$
$\mathbf{x} \leftarrow \mathbb{F}^{v_1}$
$\mathsf{samp}(\cdot) \leftarrow A^{\mathcal{O}_0,\mathcal{H}}(\llbracket P(\mathbf{x}) \rrbracket_{P \in \mathbf{P}})$
$\llbracket m'_{0,i} \rrbracket_{i \in [n]} \leftarrow \mathsf{samp}^{\mathcal{O}_0,\mathcal{H}}()$
$(\mathbf{y}_1, \ldots, \mathbf{y}_n) \leftarrow (\mathbb{F}^{v_2})^n$
$z_i \leftarrow \mathcal{H}(\llbracket H(\mathbf{x}, \mathbf{y}_i, m'_{0,i}) \rrbracket_{H \in \mathbf{H}}), 1 \leq i \leq n$
$\mathcal{I} \leftarrow A^{\mathcal{O}_0,\mathcal{H}}(\mathtt{sel}, \llbracket E(\mathbf{x}, \mathbf{y}_i, z_i, m'_{0,i}) \rrbracket_{E \in \mathbf{E}, i \in [n]})$
Opening
$\llbracket m'_{1,i} \rrbracket_{i \in [n]} \leftarrow \mathsf{samp}^{\mathcal{O}_0,\mathcal{H}}(\llbracket m'_{0,i} \rrbracket_{i \in \mathcal{I}})$
$out_A \leftarrow A^{\mathcal{O}_0,\mathcal{H}}(\mathtt{out}, (\mathbf{y}_i)_{i \in \mathcal{I}}, \llbracket m'_{b,i} \rrbracket_{i \in [n]})$
```

(a) Real Game

```
Initialization & Challenge
$b \leftarrow \{0,1\}$
$\mathbf{x} \leftarrow \mathbb{F}^{v_1}$
$\mathsf{samp}(\cdot) \leftarrow A^{\mathcal{O}_1(\mathbf{x}),\mathcal{H}}(\llbracket P(\mathbf{X}) \rrbracket_{P \in \mathbf{P}})$
$\llbracket m'_{0,i} \rrbracket_{i \in [n]} \leftarrow \mathsf{samp}^{\mathcal{O}_1(\mathbf{x}),\mathcal{H}}()$
$(\mathbf{y}_1, \ldots, \mathbf{y}_n) \leftarrow (\mathbb{F}^{v_2})^n$
$z_i \leftarrow \mathcal{H}(\llbracket H(\mathbf{X}, \mathbf{Y}_i, M_i) \rrbracket_{H \in \mathbf{H}}), 1 \leq i \leq n$
$\mathcal{I} \leftarrow A^{\mathcal{O}_1(\mathbf{x},\mathbf{y}_1,\ldots,\mathbf{y}_n,\mathbf{z},\mathbf{m}'_0),\mathcal{H}}(\mathtt{sel}, \llbracket E(\mathbf{X}, \mathbf{Y}_i, Z_i, M_i) \rrbracket_{E \in \mathbf{E}, i \in [n]})$
Opening
$\llbracket m'_{1,i} \rrbracket_{i \in [n]} \leftarrow \mathsf{samp}^{\mathcal{O}_2(\mathbf{x},\mathbf{y}_1,\ldots,\mathbf{y}_n,\mathbf{z},\mathbf{m}'_0),\mathcal{H}}(\llbracket m'_{0,i} \rrbracket_{i \in \mathcal{I}})$
$out_A \leftarrow A^{\mathcal{O}_2(\mathbf{x},\mathbf{y}_1,\ldots,\mathbf{y}_n,\mathbf{z},\mathbf{m}'_0),\mathcal{H}}(\mathtt{out}, (\mathbf{y}_i)_{i \in \mathcal{I}}, \llbracket m'_{b,i} \rrbracket_{i \in [n]})$
```

(b) Transition Game

```
Initialization & Challenge
$b \leftarrow \{0,1\}$
$\mathbf{x} \leftarrow \mathbb{F}^{v_1}$
$\mathsf{samp}(\cdot) \leftarrow A^{\mathcal{O}_2(),\mathcal{H}}(\llbracket P(\mathbf{X}) \rrbracket_{P \in \mathbf{P}})$
$\llbracket m'_{0,i} \rrbracket_{i \in [n]} \leftarrow \mathsf{samp}^{\mathcal{O}_2(),\mathcal{H}}()$
$(\mathbf{y}_1, \ldots, \mathbf{y}_n) \leftarrow (\mathbb{F}^{v_2})^n$
$z_i \leftarrow \mathcal{H}(\llbracket H(\mathbf{X}, \mathbf{Y}_i, M_i) \rrbracket_{H \in \mathbf{H}}), 1 \leq i \leq n$
$\mathcal{I} \leftarrow A^{\mathcal{O}_2(),\mathcal{H}}(\mathtt{sel}, \llbracket E(\mathbf{X}, \mathbf{Y}_i, Z_i, M_i) \rrbracket_{E \in \mathbf{E}, i \in [n]})$
Opening
$\llbracket m'_{1,i} \rrbracket_{i \in [n]} \leftarrow \mathsf{samp}^{\mathcal{O}_2((\mathbf{y}_i,z_i,m'_{0,i})_{i \in \mathcal{I}}),\mathcal{H}}(\llbracket m'_{0,i} \rrbracket_{i \in \mathcal{I}})$
$out_A \leftarrow A^{\mathcal{O}_2((\mathbf{y}_i,z_i,m'_{0,i})_{i \in \mathcal{I}}),\mathcal{H}}(\mathtt{out}, (\mathbf{y}_i)_{i \in \mathcal{I}}, \llbracket m'_{b,i} \rrbracket_{i \in [n]})$
```

(c) Simulation Game

Figure 3: IND-SO-CPA Games: From the Real Game to the Simulation Game

Clearly, due to the modification of $\mathcal{O}_1$ we changed the mapping between encodings and group elements. This might lead to a different behavior of generic algorithms when interacting with $\mathcal{O}_2$ in comparison to $\mathcal{O}_1$. More precisely, a simulation failure occurs during the

- Initialization & Challenge Phase (bad_1) if there exists $R_1, R_2 \in \mathcal{L}$ such that

$$(R_1 - R_2) \not\equiv 0 \text{ but } ((R_1 - R_2)(\mathbf{m}'_0))(\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_n, \mathbf{z}) \equiv 0 \tag{5}$$

- Opening Phase (bad_2) if there exists $R_1, R_2 \in \mathcal{L}$ such that

$$\begin{gathered} ((R_1 - R_2)(m'_{0,i})_{i \in \mathcal{I}})(z_i, \mathbf{y}_i)_{i \in \mathcal{I}} \not\equiv 0 \\ \text{but} \\ ((R_1 - R_2)(\mathbf{m}'_0))(\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_n, \mathbf{z}) \equiv 0 \end{gathered} \tag{6}$$

Note that if failure event bad_1 did not happen (during Initialization & Challenge) then bad_2 may only be caused by a new polynomial computed during the Opening Phase. So the Initialization & Challenge Phases of the simulation and transition game are equivalent unless bad_1 happens and the Opening Phases are equivalent unless bad_2 occurs. Hence, A's probability in winning the IND-SO-CPA game is upper bounded by $\frac{1}{2} + \Pr[\mathsf{bad}_1 \vee \mathsf{bad}_2]$. In other words, A causes a simulation failure with probability at least $\mathsf{Adv}^{\mathsf{ind\text{-}so\text{-}cpa}}_{\mathsf{PKE},A}$.

It remains to show that we can extract polynomials as required for Theorem 4.2 in case bad_1 or bad_2 occurs. The extraction algorithm B plays the IND-SO-CPA simulation game with A and takes over the role of the simulation oracle $\mathcal{O}_2$. B checks if bad_1 happens during the Initialization & Challenge Phase. If this is the case, it considers the corresponding polynomials which have caused the failure. Otherwise, it executes the Opening Phase and checks whether bad_2 occurs.

Let us now assume that bad_1 happens for some $\Delta := R_1 - R_2$ and $m'_{0,1}, \ldots, m'_{0,n}$ as well as $\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_n, \mathbf{z}$ chosen uniformly at random by B.[4] Note that since generic algorithms are only able to add polynomials whose encodings they receive as input, Δ is of the form

$$\Delta = \sum_{P \in \mathbf{P}} \alpha_P P(\mathbf{X}) + \sum_{j=1}^{n} \sum_{E \in \mathbf{E}} \beta_{j,E} E(\mathbf{X}, \mathbf{Y}_j, Z_j, M_j) \tag{7}$$

and $m'_{0,1}, \ldots, m'_{0,n}$ are of the form $m'_{0,i} = \sum_{P \in \mathbf{P}} \alpha_P P(\mathbf{X})$. The degree of Δ is upper bounded by $d = \max_{S \in \mathbf{P} \cup \mathbf{E}}(\deg(S))$ and the degree of $m'_{0,i}$ is upper bounded by $d' = \max_{S \in \mathbf{P}}(\deg(S))$.

In case bad_2 occurs, we consider the partially evaluated polynomial $\Delta := ((R_1 - R_2)((m'_{0,i})_{i \in \mathcal{I}}))((z_i, \mathbf{y}_i)_{i \in \mathcal{I}})$ and the polynomials $m'_{0,1}, \ldots, m'_{0,n}$ as before. Due to the form of the monomials of E, evaluation of E with $m'_{0,i}$, z_i, and $\mathbf{y}_i$ results in polynomials of the form $\sum_{P \in \mathbf{P}} \alpha_P P(\mathbf{X})$. Hence, also Δ can be viewed as a polynomial of the form in Equation 7, where the $\beta_{i,E}$ coefficients are zero for $i \in \mathcal{I}$. The upper bounds d and d' specified above also hold in this case.

To summarize, with probability at least $\Pr[\mathsf{bad}_1 \vee \mathsf{bad}_2]$, B can extract a non-zero polynomial Δ as in Equation 7 and polynomials $m'_{0,1}, \ldots, m'_{0,n}$. Δ becomes zero when evaluated with $m'_{0,1}, \ldots, m'_{0,n}$ and uniformly and independently chosen values $\mathbf{x}, \mathbf{y}_j, z_j$, where $j \in \{1, \ldots, n\}$ for the case bad_1 and $j \notin \mathcal{I}$ for the case bad_2. Applying Lemma 4.4 stated below yields that Δ already becomes zero when evaluated with the messages with probability at least $\Pr[\mathsf{bad}_1 \vee \mathsf{bad}_2] - \frac{dd'}{p}$. In this case B has found polynomials as required in Theorem 4.2.

Lemma 4.4. *Let $d, d' \in \mathbb{N}_0$, $k, i \in \mathbb{N}$ with $1 \leq i \leq k$. Let dist be a distribution over $(i+1)$-tuples $(P, x_1, \ldots, x_i)$ of polynomials from $\mathbb{F}[X_1, \ldots, X_k]$ where $P \neq 0$, $\deg(P) \leq d$, and $\deg(x_j) \leq d'$ for $1 \leq j \leq i$. Then it holds that*

$$\begin{aligned}
&\Pr_{(P,x_1,\ldots,x_i) \leftarrow \mathsf{dist}}[P(X_1 = x_1, \ldots, X_i = x_i) = 0] \geq \\
&\Pr_{\substack{(P,x_1,\ldots,x_i) \leftarrow \mathsf{dist} \\ x_{i+1},\ldots,x_k \leftarrow \mathbb{F}}}[(P(X_1 = x_1, \ldots, X_i = x_i))(X_{i+1} = x_{i+1}, \ldots, X_k = x_k) = 0] - \tfrac{dd'}{p}
\end{aligned}$$

Proof.

$$\begin{aligned}
& \Pr[(P(x_1, \ldots, x_i))(x_{i+1}, \ldots, x_k) = 0] \\
= \ & \Pr[(P(x_1, \ldots, x_i))(x_{i+1}, \ldots, x_k) = 0 \wedge P(x_1, \ldots, x_i) = 0] \\
+ \ & \Pr[(P(x_1, \ldots, x_i))(x_{i+1}, \ldots, x_k) = 0 \wedge P(x_1, \ldots, x_i) \neq 0] \\
\leq \ & \Pr[P(x_1, \ldots, x_i) = 0] \\
+ \ & \Pr[(P(x_1, \ldots, x_i))(x_{i+1}, \ldots, x_k) = 0 \mid P(x_1, \ldots, x_i) \neq 0] \\
\leq \ & \Pr[P(x_1, \ldots, x_i) = 0] + \tfrac{dd'}{p}
\end{aligned}$$

The last inequality follows from the Schwartz-Zippel Lemma. □

Let us briefly estimate the runtime of B. The algorithm runs A once, samp twice, plays the role of the IND-SO-CPA challenger, the generic oracle $\mathcal{O}_2$, and checks for a simulation failure. We will count the

[4] Note that the hash values z_j are indeed uniformly chosen since the input to the Random Oracle is guaranteed to be different for the n encryptions made: For $1 \leq j \leq n$, the variable M_j or $Y_{j,i} \in \mathbf{Y}_j$ appear in at least one of the encryption polynomials ensuring that the corresponding encoding, input to the hash function, is fresh.

number of operations on polynomials and group elements: B maintains the list $\mathcal{L}$ of polynomials on behalf of $\mathcal{O}_2$. This requires at most $O(t+t')$ additions of polynomials. Additionally, to determine encodings, B needs to compute at most $O(|\mathcal{L}|^2) = O((t+t'+|\mathbf{P}|+|\mathbf{E}|n)^2)$ difference polynomials Δ. Note that the monomials of all these polynomials come from a set of at most at most $r(|\mathbf{P}|+|\mathbf{E}|n)$ different monomials. Thus, one polynomial addition results in at most $r(|\mathbf{P}|+|\mathbf{E}|n)$ operations over $\mathbb{F}$.

To check for simulation failures, B needs to evaluate the difference polynomials Δ. To do so, B maintains a second list $\mathcal{L}' \subset \mathbb{F}$ just like the real $\mathcal{O}_0$ would do and computes the corresponding differences. Evaluating $\mathbf{P}$ and $\mathbf{E}$ when added to $\mathcal{L}'$ requires $O(\log(p)(|\mathbf{X}| \max_{P\in\mathbf{E}}(|P|)|\mathbf{P}|+|\mathbf{Y}| \max_{E\in\mathbf{E}}(|E|)|\mathbf{E}|n)$ $\mathbb{F}$-operations and computing the differences $O(|\mathcal{L}'|^2) = O((t+t'+|\mathbf{P}|+|\mathbf{E}|n)^2)$.

To check for a failure during the Opening Phase, B evaluates all polynomials in $\mathcal{L}$ with $\mathbf{y}_i, z_i, m'_{0,i}$, for $i \in \mathcal{I}$, when the Opening Phase starts. This requires $O(\log(p)|\mathbf{Y}|n \max_{E\in\mathbf{Q}}(|E|)|\mathbf{E}|)$ $\mathbb{F}$-operations. These evaluations do not increase the size of the set of monomials polynomials in $\mathcal{L}$ may consist of. □

Note that the success probability of the IND-CPA adversary B from Theorem 4.2 is non-negligible if the degrees of the public key and encryption polynomials of PKE are small, i.e., polynomial in $\log(p)$. Moreover, B is efficient if the representation of these polynomials is polynomial in $\log(p)$ (always the case for an efficient encryption scheme) as well as the number n of involved message polynomials R_i. The same statement holds for the polynomial extraction algorithm from Theorem 4.3, where we additionally need to assume that the runtime of the IND-SO-CPA adversary is polynomial and its advantage is non-negligible.

Acknowledgements

We would like to thank the anonymous reviewers for very helpful and constructive comments.

References

[1] Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 36–54. Springer (Aug 2009)

[2] Barak, B., Haitner, I., Hofheinz, D., Ishai, Y.: Bounded key-dependent message security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 423–444. Springer (May 2010)

[3] Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: Krawczyk, H. (ed.) CRYPTO'98. LNCS, vol. 1462, pp. 26–45. Springer (Aug 1998)

[4] Bellare, M., Dowsley, R., Waters, B., Yilek, S.: Standard security does not imply security against selective-opening. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 645–662. Springer (Apr 2012)

[5] Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer (Apr 2009)

[6] Bellare, M., Sahai, A.: Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In: Wiener, M.J. (ed.) CRYPTO'99. LNCS, vol. 1666, pp. 519–536. Springer (Aug 1999)

[7] Black, J., Rogaway, P., Shrimpton, T.: Encryption-scheme security in the presence of key-dependent messages. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 62–75. Springer (Aug 2002)

[8] Böhl, F., Hofheinz, D., Kraschewski, D.: On definitions of selective opening security. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 522–539. Springer (May 2012)

[9] Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-secure encryption from decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer (Aug 2008)

[10] Cramer, R., Hanaoka, G., Hofheinz, D., Imai, H., Kiltz, E., Pass, R., Shelat, A., Vaikuntanathan, V.: Bounded CCA2-secure encryption. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 502–518. Springer (Dec 2007)

[11] Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (1998), manuscript

[12] Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: 40th FOCS. pp. 523–534. IEEE Computer Society Press (Oct 1999)

[13] Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: 49th FOCS. pp. 293–302. IEEE Computer Society Press (Oct 2008)

[14] Fehr, S., Hofheinz, D., Kiltz, E., Wee, H.: Encryption schemes secure against chosen-ciphertext selective opening attacks. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 381–402. Springer (May 2010)

[15] Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)

[16] Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 70–88. Springer (Dec 2011)

[17] Hofheinz, D.: All-but-many lossy trapdoor functions. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 209–227. Springer (Apr 2012)

[18] Micali, S., Reyzin, L.: Physically observable cryptography (extended abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer (Feb 2004)

[19] Myers, S., Shelat, A.: Bit encryption is complete. In: 50th FOCS. pp. 607–616. IEEE Computer Society Press (Oct 2009)

[20] Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: Kosaraju, S.R. (ed.) 12th SODA. pp. 448–457. ACM-SIAM (Jan 2001)

[21] Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer (Aug 2008)

[22] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 187–196. ACM Press (May 2008)

[23] Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT'97. LNCS, vol. 1233, pp. 256–266. Springer (May 1997)

A Some Remarks on Definition 4.1

We would like to note that the restrictions on the form and degrees of polynomials made in Definition 4.1 are not of artificial nature and just derived from the proofs but need to be satisfied by a meaningful encryption scheme. On the other hand, we would like to stress that they are not sufficient for such a scheme as, e.g., no conditions on the nature of the decryption algorithm are made.

In particular, the encryption polynomials E may not be "arbitrary" polynomials in $\mathbf{X}$ since during encryption we are usually only given $P(\mathbf{x})$ and do not know how to evaluate encryption polynomials not being "combinations" of public key polynomials. Furthermore, any public key polynomial may only appear linearly in any encryption polynomial. Otherwise, in absence of a pairing we do not know how to compute, e.g., P^2 efficiently. In fact, in the case of a single group this translates to solving the Square-DH problem. For this reason, also any monomial of an encryption polynomial might only contain at most one public key polynomial. Moreover, assume an encryption polynomial E would include a monomial of the form $\alpha P^{e_1} Z^{e_2} M^{e_3} \prod_{i=1}^{v_2} Y_i^{d_i}$ with $e_1 + e_3 > 1$. This would mean that we have to solve a variant of the DH problem to encrypt a message unless we know the DL of the message.

Finally, the condition on the input of the hash function ensures that the input is not constant for different m'. The use of a hash function for constant input would be meaningless. Note that for an encryption scheme without hash function like ElGamal e_2 is simply set to zero in all encryption polynomials.

B A Separating CS-type Encryption Scheme

Interestingly, we again obtain a CS-type scheme if we instantiate (a slight variation of) the separating scheme from Section 3.1 with a CS-type scheme according to Definition 4.1. Some details can be found in the following.

- Compared to the original definition of the separating scheme, the message space of a CS-type scheme is a group G of order p and not $\mathbb{F}$. But as already mentioned, this is no real issue. What we can do in our particular case here is the following: We only need a means such as the degree-$\leq k$ polynomial F before that allows to generate and reconstruct the data points $(i, m_i)_{i\in[3k]}$, where $m_i = g^{m'_i} \in G$. This can easily be done "using F in the exponent". Especially, Lagrange Interpolation works in this case. To evaluate a degree-$\leq k$ interpolation polynomial F defined by $k+1$ data points (i, m_i) in the exponent with some $\ell \in [3k]$, one would compute $\prod_i m_i^{Q_i(\ell)}$, where $Q_i(x) = \prod_{t\neq i} \frac{x-t}{i-t}$ is a Lagrange basis polynomial. To determine if there exists a unique F in Step 5 of $\mathsf{SOA}(sk, Z)$, one could check if there exists some $j \in [3k] \setminus \mathcal{I}$ such that

$$\prod_{i\in\mathcal{I}\cup\{j\}} m_i^{Q_i(\ell)} = m_\ell$$

  for k values $\ell \in [3k] \setminus (\mathcal{I} \cup \{j\})$. If this is the case, it suffices to return m_j.
- The secret key of the CS-type scheme needs to be extended by a key K for the PRF. Note that this is not forbidden by Definition 4.1. Moreover, due to our slight modification above, we do not need an additional prime p in the public key of the CS-type scheme.
- In order to mark a ciphertext as regular, Enc' simply adds a fixed group element to each regular CS-type ciphertext. For instance, this could translate to adding polynomials $P_6 = 0$ and $E_5 = P_6$ to the specification of Cramer-Shoup as CS-type encryption scheme shown in Section 4.1. The other types of inputs we allow to Dec' can be marked similarly using other fixed elements.
- Note that apart from these markers, we do not need to care about how the decryption function looks like since Definition 4.1 only specifies the form of public keys and (regular) ciphertexts.

C Our CCA-Separation Works in the GGM

In this section, we briefly argue why our CCA-separation also holds in the GGM, i.e., there exists a IND-CCA secure generic group encryption scheme that can be efficiently broken by a generic group IND-SO-CCA adversary.

First, it is easy to see that our separating scheme works over any (prime order) group G when it is instantiated with a generic group encryption scheme like Cramer-Shoup: The original IND-CCA secure scheme (CS in our case) is treated as a black box and also the modifications applied work for any group. In particular, in Appendix B we show how the message space can be switched to G and how $\mathsf{SOA}(sk, Z)$ can be implemented in this case. It is clear, that the resulting separating scheme is IND-CCA secure despite this switch of the message space. To summarize, it perfectly makes sense and is meaningful to consider the IND-SO-CCA game for our separating scheme in the GGM.

Second, the IND-SO-CCA adversary A and its sampling algorithm only apply generic group operations: The message distribution A outputs will be

$$\mathcal{D} = \left\{ (g^{F(1)}, \ldots, g^{F(3k)}) \,\middle|\, F \in \mathbb{F}[X] \text{ uniformly chosen degree-} \leq k \text{ polynomial} \right\},$$

where $\langle g \rangle = G$, and can be implemented by a generic group algorithm. Moreover, the polynomial interpolation A's resampling algorithm uses can be realized over generic groups as shown in Appendix B. As also shown there, the interpolation polynomial returned by the decryption oracle on a $\mathtt{soa}$-query can be evaluated using only multiplications with given group elements.

D Variations of our CCA-Separation

An observation. We remark that the attack from Theorem 3.1 actually only uses two decryption queries. Moreover, one of these queries is a query $(\mathtt{sel}, Z)$ to a (pseudo)random function. Our proof would work also in the random oracle model, if we defined $\mathcal{I} = \mathcal{RO}((c_i')_{i\in[3k]})$ (instead of $\mathcal{I} = \mathsf{PRF}_K((c_i')_{i\in[3k]})$). With this change, we would get the same separation in the random oracle model, but with a weak IND-SO-CCA attack that requires only one decryption query.

Bounded CCA security. Cramer et al. [10] define a bounded notion (called IND-q-CCA security) of IND-CCA security, in which an adversary only gets an a-priori bounded number q of decryption queries. If we define weak IND-SO-q-CCA security in the obvious way, our observation above immediately yields a separation between IND-2-CCA and weak IND-SO-2-CCA security. Furthermore, we get a separation between IND-1-CCA and weak IND-SO-1-CCA security in the random oracle model.

Non-malleability. IND-1-CCA security is known to be tightly related to non-malleability [11, 3]. Concretely, Bellare and Sahai [6] show that non-malleability under chosen-plaintext attacks (NM-CPA) is equivalent to a mild form of IND-CCA security, which in turn implies IND-1-CCA security. Since our results yield a separation between IND-1-CCA and IND-SO-1-CCA security in the random oracle model, we can expect a similar separation between between NM-CPA and NM-SO-CPA security. Here, NM-SO-CPA stands for "non-malleability under chosen-plaintext selective opening attacks," a notion which has not yet been formally defined. (We leave such a definition for future work; however, if one opts to simply equip an NM-CPA adversary with an "opening oracle" for NM-SO-CPA, the random oracle variation of our result seems to directly apply.)